



A Genetic Algorithm Search for the Optimal Design of Water Distribution Systems

Laurence Joseph Murphy

**Thesis submitted for the degree of
Doctor of Philosophy**

in

The University of Adelaide

Faculty of Engineering

Department of Civil and Environmental Engineering

Statement of Originality

This work contains no material which has been accepted for the award of any other degree or diploma in any University or other tertiary institution and, to the best of my knowledge and belief, contains no material previously written or published by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

DATE: 7/10/97

Acknowledgements

This research was conducted in the Department of Civil and Environmental Engineering at the University of Adelaide between 1991 and 1997. The author was supervised for the period of his candidature by Dr. Angus R. Simpson and Associate Professor Graeme C. Dandy. The author is indebted to the his supervisors and the Department for the opportunity to pursue this research. The author is most grateful to his supervisors, Dr Simpson and Assoc. Prof Dandy, for their guidance and encouragement, their valued advice and also their endurance, in fostering this research from an undergraduate research project.

The author would like to give special mention to Mr Jeffery Frey and Mr John Gransbury for their unwavering enthusiasm. They have been, along with my supervisors, genuine, energetic supporters of this genetic algorithm application.

The author would also like to extend thanks to Mr. Rod Kitto, Mr. Terry Farrill, Mr. Paul Doherty and Mr. Jim Zissopoulos, Mr. Andris Rubenis and Mr. Mun Thim Wong. The willingness of these system engineers and designers to provide details of their water systems is greatly appreciated, as is their foresight and openness to this new idea. They have helped to convince me of the importance of efforts to apply this research to real-world water system applications.

The author acknowledges the support provided by the administrative and computing staff of the Department of Civil and Environmental Engineering, and for the valuable association with other teachers and fellow students. Funding for stages of this research was provided by the Department, by way of a CivilTest Scholarship and this support is gratefully acknowledged.

I would especially like to thank my wonderful family, my friends and my wife-to-be (soon), Jacqueline, for all believing in me, and thank you Bobby Z.

All these people, and many others are thanked for their contribution in many different ways to this research.

Table of Contents

<i>Statement of Originality</i>	ii
<i>Acknowledgements</i>	iii
<i>Table of Contents</i>	iv
<i>List of Figures</i>	x
<i>List of Tables</i>	xviii
<i>Principal Notations</i>	xxiii
Chapter 1. Introduction	1
1.1 Water Distribution Systems	1
1.2 System Expansions	1
1.3 Hydraulic Simulation of Water Distribution Systems	2
1.4 Optimisation of the Design of Water Distribution Systems	2
1.5 Pipe Network Optimisation Techniques	3
1.6 Genetic Algorithms (GAs)	4
1.7 A Traditional GA for Pipe Network Optimisation	5
1.8 The Two-Reservoir Gessler Network	7
1.9 Modifications to the Traditional GA	8
1.10 Multiple Gessler Problems	8
1.11 The New York City Water Supply Tunnels	9
1.12 The Fort Collins - Loveland System Expansion Plan	9
Chapter 2. Development of a Hydraulic Simulation Model	11
2.1 Introduction	11
2.2 Pipe Network Components	13
2.3 Tree Networks and Looped Networks	14
2.4 Natural Loops and Pseudo Loops	15
2.5 Hydraulic Grade Line (HGL)	15
2.6 Pipe Friction Head Losses	17
2.7 Parallel Pipes	18
2.8 Pumping Heads	19
2.9 The Set of Pipe Network Equations	21
2.9.1 Pipe flow equations (<i>Q</i> -equations)	23
2.9.2 Node equations (<i>H</i> -equations)	27
2.9.3 Loop corrective flow equations (ΔQ_L -equations)	30
2.10 The Adopted Method of Hydraulic Analysis	34
2.10.1 The <i>Anytown</i> water distribution system	34

2.10.2	Determination of natural loops and pseudo loops	38
2.10.3	Assumed initial flows	42
2.10.4	Near minimum bandwidth of the Jacobian matrix	45
2.10.5	The Newton-Raphson method applied to the loop equations	47
2.10.6	Convergence test	52
2.10.7	Sparse matrix routines	54
2.10.8	Junction node pressure heads	61
2.10.9	Analysis of pressure reducing valves and check valves	62
2.10.10	Extended period simulation (EPS)	66
2.11	Integration of the Hydraulic Simulation Model and the Genetic Algorithm Pipe Network Optimisation Model	68
Chapter 3. Optimisation of Water Distribution System Design		73
3.1	The Optimisation Problem	73
3.1.1	The decision variables	74
3.1.2	The objective function	77
3.1.3	System performance constraints	77
3.1.4	Hydraulic constraints	79
3.1.5	General design constraints	79
3.1.6	The solution space	79
3.2	The Benefits of Optimisation	80
3.3	Pipe Network Optimisation Techniques	81
3.3.1	Nonlinear programming (NLP)	81
3.3.2	Linear programming (LP)	85
3.3.3	Two-phase decomposition methods	90
3.3.4	Global search and local optimisation	91
3.3.5	Dynamic programming	93
3.3.6	Enumeration algorithms	95
3.3.7	Heuristic techniques	96
3.3.8	Equivalent pipe methods	98
3.3.9	Evolutionary strategy	99
3.4	Complexities of Pipe Network Optimisation	100
Chapter 4. Overview of Genetic Algorithms		102
4.1	Chromosomes	102
4.2	Fitness of Coded Structures	104
4.3	The Solution Space	105
4.4	Populations of Coded Structures	105
4.5	Genetic Algorithm Operations	106

4.6 Selection	106
4.7 Crossover	108
4.8 Mutation	109
4.9 String Similarities (Schemata)	109
4.10 The Power of the Genetic Algorithm	111
4.11 GAs applied to Pipe Optimisation Problems	111
Chapter 5. Application of the Traditional Genetic Algorithm to Pipe Network Optimisation	113
5.1 A Genetic Algorithm Approach	113
5.1.1 Coded strings	113
5.1.2 Fitness of coded strings	114
5.1.3 Implementation of a simple genetic algorithm	114
5.2 Case Study: The Two-Reservoir <i>Gessler</i> Pipe Network	118
5.2.1 Description of the <i>Gessler</i> problem	118
5.2.2 Exhaustive enumeration of the <i>Gessler</i> problem	121
5.2.3 Partial enumeration applied by <i>Gessler</i> (1985)	127
5.2.4 Nonlinear programming optimisation of the <i>Gessler</i> problem	128
5.3 A Small Scale Simulation of the Simple Genetic Algorithm applied to the Gessler Problem	132
5.3.1 Coded strings representing <i>Gessler</i> network designs	132
5.3.2 Coded string populations	134
5.3.3 Evaluation of the coded strings	134
5.3.4 The second generation of coded strings	139
5.3.5 Selection	139
5.3.6 One-point crossover	140
5.3.7 Mutation	141
5.3.8 Subsequent generations and the Schema Theorem	142
5.4 Sensitivity Analysis of Genetic Algorithm Parameters	143
5.4.1 Variations of the random number generator seed	145
5.4.2 Variations of the population size, N	149
5.4.3 Variations of the probability of crossover, p_c	155
5.4.4 Variations of the probability of mutation, p_m	159
5.4.5 Infeasible network designs	162
5.4.6 Findings of the sensitivity analysis	163
5.5 Conclusions	164

Chapter 6. Improvements to the Simple Genetic Algorithm for Pipe Network Optimisation	166
6.1 Introduction	166
6.2 Performance Measures	166
6.3 Penalty Functions	171
6.3.1 Penalty functions for pipe network optimisation	172
6.3.2 The penalty multiplier	173
6.3.3 Varying the penalty multiplier	174
6.3.4 GA model runs to compare penalty functions and penalty multipliers	175
6.3.5 Recommendations for the penalty function	187
6.4 Selection Methods and Fitness Functions	188
6.4.1 Proportionate selection	188
6.4.2 Tournament selection	189
6.4.3 Fitness functions	190
6.4.4 Fitness scaling mechanisms	191
6.4.5 Experimental analysis of selection methods and fitness functions	192
6.4.6 Recommendations for selection schemes	204
6.5 Coding Schemes	206
6.5.1 Binary codes and Gray codes	208
6.5.2 Integer codes	210
6.5.3 The optimum arrangement of decision-variable substring positions within the coded string	212
6.5.4 Genetic algorithm runs to compare coding schemes	213
6.5.5 Counting the numbers of decision-variable substrings	219
6.5.6 An ideal coded structure	229
6.6 Crossover Mechanisms	232
6.6.1 GA model runs to compare crossover mechanisms	235
6.6.2 Recommendations for crossover mechanisms	242
6.7 A Creeping Mutation Operator	245
6.7.1 A creeping mutation operator for binary-coded substrings	246
6.7.2 GA model runs to measure the effectiveness of creeping mutations	247
6.7.3 Recommendations for creeping mutations	250
6.8 Conclusions	257
 Chapter 7. Larger Problems with Known Optimal Solutions	 259
7.1 The Original Gessler Problem	259
7.2 Simultaneous Optimisation of Two Gessler Problems	259
7.2.1 The <i>improved</i> genetic algorithm approach	261
7.2.2 Elitism	263

7.2.3 Performance of the improved GA (with elitism) applied to two Gessler problems	265
7.3 Simultaneous Optimisation of Three Gessler Problems	268
7.4 Simultaneous Optimisation of Five Gessler Problems	273
7.4.1 The GA parameter sets F1-F5	274
7.4.2 Performance of the improved GA (with elitism) applied to five Gessler problems	276
7.4.3 Performance of the improved GA (without elitism) applied to five Gessler problems	279
7.4.4 Performance of the traditional GA applied to five Gessler problems	279
7.5 Conclusions	280

Chapter 8. An Improved Genetic Algorithm Formulation Applied to the New York Tunnels Problem

8.1 The New York Tunnels Problem	282
8.2 The Genetic Algorithm Optimisation Approach	286
8.3 The Improved Genetic Algorithm Formulation	287
8.3.1 Structure of the coded strings	287
8.3.2 Binary codes and Gray codes	287
8.3.3 Raw fitness of a coded string	289
8.3.4 The reproduction operator	289
8.3.5 Scaled fitness of a coded string	290
8.3.6 The penalty function	290
8.3.7 Creeping mutations	291
8.3.8 Elitism	292
8.3.9 Population size, crossover and random bit-wise mutations	292
8.4 Establishing a Penalty Multiplier (GA Runs NY1-NY10)	294
8.4.1 Results of GA Runs NY1-NY10	295
8.5 Performance of the Improved GA Formulation (GA Runs NY11-NY69)	302
8.5.1 Results of GA Runs NY11-NY69	303
8.6 The GA Solutions to the New York Tunnels Problem	320
8.7 Comparison of GA Results with Previous Studies	323
8.8 Summary and Conclusions	328

Chapter 9. GA Optimisation of the Water System Expansion Plan for the Fort Collins - Loveland Water District

9.1 Introduction	330
9.2 The 1993 Master Plan	330
9.3 The Genetic Algorithm Approach to the FCLWD System	331

9.4 Sources of Supply in 2015	333
9.5 Booster Pump Stations	335
9.6 Storage Tanks	337
9.7 Existing Pipelines	338
9.8 New Pipes and Duplicate Pipes	338
9.8.1 The Master Plan pipe network design	339
9.8.2 The Genetic Algorithm pipe network design	343
9.9 Pressure Reducing Valve Settings	348
9.10 System Performance of Master Plan and GA Design	351
9.10.1 Node pressures and pipe velocities	352
9.10.2 Supply flows, transmission flows, tank outflows and inflows	353
9.10.3 Taft Hill source pump station	355
9.11 Extended Period Simulation (EPS)	356
9.11.1 Maximum day demands for 2015	356
9.11.2 Variable head storage tanks	357
9.11.3 Pump station characteristics for EPS	361
9.11.4 The sources of supply	363
9.11.5 System performance for EPS	364
9.12 Summary and Conclusions	364
9.13 Scope for Further GA Optimisation Expansion Planning	365
 Chapter 10. Conclusions and Recommendations	 367
10.1 GAs for Pipe Network Optimisation	373
10.2 The Hydraulic Simulation Model	374
10.3 Other Possible Applications of GAs to Pipe Networks	375
10.4 Future GA Model Development	375
 Chapter 11. References	 378
 Appendix A Fort Collins - Loveland System Expansion Plan EPANET hydraulic simulation input data and output results for the Master Plan design subject to the 2015 peak hour demands	 388
 Appendix B Fort Collins - Loveland System Expansion Plan EPANET hydraulic simulation input data and output results for the Genetic Algorithm design subject to the 2015 peak hour demands	 402

List of Figures

1.1	Flow chart for the proposed genetic algorithm model	6
2.1	The HGL for a length of pipe	16
2.2	The HGL at a reservoir	16
2.3	The head added by a pump	19
2.4	The pump head and system head curves	20
2.5	The <i>Anytown</i> water distribution system	35
2.6	The natural loops (A-R) and pseudo loops (S-T) identified for the <i>Anytown</i> pipe network	40
2.7	A tree network generated by a Breadth First Search (BFS) originating from the principal source node 10	43
2.8	The loops ordered for near minimum bandwidth	46
2.9	The operative mode for a PRV	63
2.10	The inoperative mode for a PRV	63
2.11	The open check valve	65
2.12	The closed check valve	65
3.1	Coupled NLP model and simulation model	84
3.2	Example pipe network with assumed flow directions	93
3.3	Decomposition of the dynamic programming procedure	94
4.1	Hypothetical pipe network design problem requiring the selection of five new pipe sizes	103
4.2	A model of a generation of a simple evolution strategy	107
5.1	Layout of the <i>Gessler</i> Network	119
5.2	Number of solutions with a particular number of nodes at which pressure constraints are violated	124
5.3	Critical demand patterns and critical nodes	124
5.4	Feasible solutions when implementing pipe [1] rehabilitation decisions	125
5.5	Feasible solutions when implementing decisions for new pipe [6]	125
5.6	The feasibility of solutions with given pipe network costs	126
5.7	The formation of a 24-bit coded string	133
5.8	Best generation costs for random number seeds, <i>seed</i> =200 and <i>seed</i> =400	148
5.9	Average generation costs for random number seeds, <i>seed</i> =200 and <i>seed</i> =400	148
5.10	Best generation costs for a population size, <i>N</i> =20	151

5.11	Average generation costs for a population size, $N=20$	151
5.12	Best generation costs for a population size, $N=50$	152
5.13	Average generation costs for a population size, $N=50$	152
5.14	Best generation costs for a population size, $N=80$	153
5.15	Average generation costs for a population size, $N=80$	153
5.16	Best generation costs for population sizes, $N=150$ and $N=500$	154
5.17	Average generation costs for population sizes, $N=150$ and $N=500$	154
5.18	Best generation costs for probability of crossover, $p_c=0.2$ and $p_c=0.4$	157
5.19	Best generation costs for probability of crossover, $p_c=0.6$ and $p_c=0.8$	157
5.20	Best generation costs for a probability of crossover, $p_c=1.0$	158
5.21	Average generation costs for varying probability of crossover, p_c	158
5.22	Best generation costs for probability of mutation, $p_m=0.0$ and $p_m=0.005$	160
5.23	Best generation costs for probability of mutation, $p_m=0.03$ and $p_m=0.1$	160
5.24	Average generation costs for varying probability of mutation, p_m	161
6.1	Best generation costs (feasible solutions only) for GA runs PEN4 ($k=\$25,000/\text{psi}$), PEN14 ($k=\$50,000/\text{psi}$) and PEN24 ($k=\$75,000/\text{psi}$) with penalties PC_{max} based on <i>maximum</i> violations of pressure constraints for each demand pattern	181
6.2	Best generation costs (feasible solutions only) for GA runs PEN34 ($k=\$25,000/\text{psi}$), PEN44 ($k=\$50,000/\text{psi}$) and PEN54 ($k=\$75,000/\text{psi}$) with penalties PC_{sum} based on <i>sum</i> of violations of pressure constraints for each demand pattern	181
6.3	Best generation costs (feasible solutions only) for GA runs PEN14 ($k=\$50,000/\text{psi}$ with penalties PC_{max}), PEN44 ($k=\$50,000/\text{psi}$ with penalties PC_{sum}) and PEN64 (increasing k with penalties PC_{max})	182
6.4	Best generation costs (feasible and infeasible solutions) for GA runs PEN14 ($k=\$50,000/\text{psi}$ with penalties PC_{max}), PEN44 ($k=\$50,000/\text{psi}$ with penalties PC_{sum}) and PEN64 (increasing k with penalties PC_{max})	182
6.5	Average generation costs (feasible and infeasible) for GA runs PEN4 ($k=\$25,000/\text{psi}$), PEN14 ($k=\$50,000/\text{psi}$) and PEN24 ($k=\$75,000/\text{psi}$) with penalties based on <i>maximum</i> pressure constraint violations	184
6.6	Average generation costs (feasible and infeasible) for GA runs PEN34 ($k=\$25,000/\text{psi}$), PEN44 ($k=\$50,000/\text{psi}$) and PEN54 ($k=\$75,000/\text{psi}$) with penalties based on <i>sum</i> of all pressure constraint violations	184
6.7	Average generation costs (feasible and infeasible) for GA runs PEN14 ($k=\$50,000/\text{psi}$ with penalties PC_{max}), PEN44 ($k=\$50,000/\text{psi}$ with penalties PC_{sum}) and PEN64 (increasing k with penalties PC_{max})	184
6.8	Infeasible solutions in the populations (population size, $N=100$) for GA runs PEN4 ($k=\$25,000/\text{psi}$), PEN14 ($k=\$50,000/\text{psi}$) and PEN24 ($k=\$75,000/\text{psi}$) with penalties based on <i>maximum</i> constraint violations	186

6.9 Infeasible solutions in the populations ($N=100$) for GA runs PEN34 ($k=\$25,000/\text{psi}$), PEN44 ($k=\$50,000/\text{psi}$) and PEN54 ($k=\$75,000/\text{psi}$) with penalties based on <i>sum</i> of all constraint violations	186
6.10 Infeasible solutions in the populations for GA runs PEN14 ($k=\$50,000/\text{psi}$ with penalties PC_{max}), PEN44 ($k=\$50,000/\text{psi}$ with penalties PC_{sum}) and PEN64 (increasing k as GA run proceeds with penalties PC_{max})	186
6.11 Best generation costs for GA runs FIT4 ($n=1$ throughout), FIT14 ($n=2$ throughout) and FIT44 (n is increasing as GA run proceeds) using the <i>inverse</i> fitness function	200
6.12 Best generation costs for GA runs FIT24 ($n=1$ throughout), FIT34 ($n=2$ throughout) and FIT54 (n is increasing as GA run proceeds) using the <i>linear</i> fitness function	200
6.13 Best generation costs for GA runs FIT64 (binary tournament with $p_r=1.0$), FIT74 (binary tournament with $p_r=0.9$), FIT84 (binary tournament with $p_r=0.8$) and FIT94 (ternary tournament)	200
6.14 Offline performance (running average of best cost solutions) for GA runs FIT4 ($n=1$ throughout), FIT14 ($n=2$ throughout) and FIT44 (n is increasing as GA run proceeds) using the <i>inverse</i> fitness function	201
6.15 Offline performance (running average of best cost solutions) for GA runs FIT24 ($n=1$ throughout), FIT34 ($n=2$ throughout) and FIT54 (n is increasing as GA run proceeds) using the <i>linear</i> fitness function	201
6.16 Offline performance for GA runs FIT64 (binary tournament with $p_r=1.0$), FIT74 (binary tournament with $p_r=0.9$), FIT84 (binary tournament with $p_r=0.8$) and FIT94 (ternary tournament)	201
6.17 Average generation costs for GA runs FIT4 ($n=1$ throughout), FIT14 ($n=2$ throughout) and FIT44 (n is increasing as GA run proceeds) using the <i>inverse</i> fitness function	202
6.18 Average generation costs for GA runs FIT24 ($n=1$ throughout), FIT34 ($n=2$ throughout) and FIT54 (n is increasing as GA run proceeds) using the <i>linear</i> fitness function	202
6.19 Average generation costs for GA runs FIT64 (binary tournament with $p_r=1.0$), FIT74 (binary tournament with $p_r=0.9$), FIT84 (binary tournament with $p_r=0.8$) and FIT94 (ternary tournament)	202
6.20 Online performance (running average of all solutions) for GA runs FIT4 ($n=1$ throughout), FIT14 ($n=2$ throughout) and FIT44 (n is increasing as GA run proceeds) using the <i>inverse</i> fitness function	203
6.21 Online performance (running average of all solutions) for GA runs FIT24 ($n=1$ throughout), FIT34 ($n=2$ throughout) and FIT54 (n is increasing as GA run proceeds) using the <i>linear</i> fitness function	203

6.22	Online performance for GA runs FIT64 (binary tournament with $p_r=1.0$), FIT74 (binary tournament with $p_r=0.9$), FIT84 (binary tournament with $p_r=0.8$) and FIT94 (ternary tournament)	203
6.23	Average generation costs and average cost of strings selected from the generation for mating for GA run FIT4 ($n=1$ throughout using the <i>inverse</i> fitness function)	205
6.24	Average generation costs and average cost of strings selected from the generation for mating for GA run FIT44 (n is increasing as GA run proceeds using the <i>inverse</i> fitness function)	205
6.25	Average generation costs and average cost of strings selected from the generation for mating for GA run FIT64 (binary tournament with $p_r=1.0$)	205
6.26	The formation of the coded string for the Gessler problem	207
6.27	An alternative arrangement of the decision substrings in the string	213
6.28	Best generation costs for GA runs CODE4 (substrings of binary codes), CODE14 (Gray codes), CODE24 (strings of integers) and CODE34 (binary codes with an alternative arrangement of substring positions)	217
6.29	Offline performance for GA runs CODE4 (substrings of binary codes), CODE14 (Gray codes), CODE24 (strings of integers) and CODE34 (binary codes with an alternative arrangement of substring positions)	217
6.30	Average generation costs for GA runs CODE4 (substrings of binary codes), CODE14 (Gray codes), CODE24 (strings of integers) and CODE34 (binary codes with an alternative arrangement of substring positions)	218
6.31	Online performance for GA runs CODE4 (substrings of binary codes), CODE14 (Gray codes), CODE24 (strings of integers) and CODE34 (binary codes with an alternative arrangement of substring positions)	218
6.32	The variations with time of numbers of decision-variable substrings of binary codes at the first substring position (corresponding to existing pipe [1]) for GA run CODE4 (optimum substring is OOO)	220
6.33	The variations with time of numbers of decision-variable substrings of Gray codes at the first substring position (corresponding to existing pipe [1]) for GA run CODE14 (optimum substring is OOO)	220
6.34	The variations with time of numbers of decision-variable substrings of binary codes at the second substring position (corresponding to existing pipe [4]) for GA run CODE4 (optimum substring is llo)	221
6.35	The variations with time of numbers of decision-variable substrings of Gray codes at the second substring position (corresponding to existing pipe [4]) for GA run CODE14 (optimum substring is lOl)	221
6.36	The variations with time of numbers of decision-variable substrings of binary codes at the third substring position (corresponding to existing pipe [5]) for GA run CODE4 (optimum substring is OOO)	222

6.37	The variations with time of numbers of decision-variable substrings of Gray codes at the third substring position (corresponding to existing pipe [5]) for GA run CODE14 (optimum substring is OOO)	222
6.38	The variations with time of numbers of decision-variable substrings of binary codes at the fourth substring position (corresponding to new pipe [6]) for GA run CODE4 (optimum substring is Oll)	223
6.39	The variations with time of numbers of decision-variable substrings of Gray codes at the fourth substring position (corresponding to new pipe [6]) for GA run CODE14 (optimum substring is OIO)	223
6.40	The variations with time of numbers of decision-variable substrings of binary codes at the fifth substring position (corresponding to new pipe [8]) for GA run CODE4 (optimum substring is OOl)	224
6.41	The variations with time of numbers of decision-variable substrings of Gray codes at the fifth substring position (corresponding to new pipe [8]) for GA run CODE14 (optimum substring is OOl)	224
6.42	The variations with time of numbers of decision-variable substrings of binary codes at the sixth substring position (corresponding to new pipe [11]) for GA run CODE4 (optimum substring is OOl or OIO)	225
6.43	The variations with time of numbers of decision-variable substrings of Gray codes at the sixth substring position (corresponding to new pipe [11]) for GA run CODE14 (optimum substring is OOl or Oll)	225
6.44	The variations with time of numbers of decision-variable substrings of binary codes at the seventh substring position (corresponding to new pipe [13]) for GA run CODE4 (optimum substring is OOO)	226
6.45	The variations with time of numbers of decision-variable substrings of Gray codes at the seventh substring position (corresponding to new pipe [13]) for GA run CODE14 (optimum substring is OOO)	226
6.46	The variations with time of numbers of decision-variable substrings of binary codes at the last substring position (corresponding to new pipe [14]) for GA run CODE4 (optimum substring is OIO or OOl)	227
6.47	The variations with time of numbers of decision-variable substrings of Gray codes at the last substring position (corresponding to new pipe [14]) for GA run CODE14 (optimum substring is Oll or OOl)	227
6.48	An ideal coded structure representation of the Gessler problem	230
6.49	Array representing ideal coded structure	230
6.50	Possible crossover cuts for the ideal coded structure	231
6.51	Simple one-point crossover	232
6.52	Two-point crossover	232
6.53	Uniform crossover (multiple random crossover points)	233

6.54	Two-point crossover (crossover points at substring boundaries)	235
6.55	Best generation costs for GA runs CROSS14 (two-point crossover), CROSS24 (four-point crossover) and CROSS34 (uniform crossover)	240
6.56	Best generation costs for GA runs CROSS44 (one-point crossover at substring boundaries), CROSS54 (two-point crossover at substring boundaries) and CROSS64 (uniform crossover at substring boundaries)	240
6.57	Best generation costs for GA runs CROSS4 (one-point crossover) and CROSS44 (one-point crossover at decision-variable substring boundaries)	240
6.58	Offline performance (running average of best solution costs) for GA runs CROSS14 (two-point crossover), CROSS24 (four-point crossover) and CROSS34 (uniform crossover)	241
6.59	Offline performance for GA runs CROSS44 (one-point crossover at substring boundaries), CROSS54 (two-point crossover at substring boundaries) and CROSS64 (uniform crossover at substring boundaries)	241
6.60	Offline performance for GA runs CROSS4 (one-point crossover) and CROSS44 (one-point crossover at decision-variable substring boundaries)	241
6.61	Average generation costs for GA runs CROSS14 (two-point crossover), CROSS24 (four-point crossover) and CROSS34 (uniform crossover)	243
6.62	Average generation costs for GA runs CROSS44 (one-point crossover at substring boundaries), CROSS54 (two-point crossover at substring boundaries) and CROSS64 (uniform crossover at boundaries)	243
6.63	Average generation costs for GA runs CROSS4 (one-point crossover) and CROSS44 (one-point crossover at decision-variable substring boundaries)	243
6.64	Online performance (running average of all solution costs) for GA runs CROSS14 (two-point crossover), CROSS24 (four-point crossover) and CROSS34 (uniform crossover)	244
6.65	Online performance for GA runs CROSS44 (one-point crossover at substring boundaries), CROSS54 (two-point crossover at substring boundaries) and CROSS64 (uniform crossover at substring boundaries)	244
6.66	Online performance for GA runs CROSS4 (one-point crossover) and CROSS44 (one-point crossover at decision-variable substring boundaries)	244
6.67	The action of a creeping mutation	247
6.68	Best generation costs for GA runs CREEP4 (No creeping mutation, $p_c=0.0$) and CREEP24 (probability of creep, $p_c=0.125$, and probability of creeping down, $p_d=0.5$)	253
6.69	Best generation costs for GA runs CREEP14 ($p_c=0.0625$, $p_d=0.5$), CREEP24 ($p_c=0.125$, $p_d=0.5$) and CREEP34 ($p_c=0.25$, $p_d=0.5$)	253
6.70	Best generation costs for GA runs CREEP24 ($p_c=0.125$, $p_d=0.5$), CREEP44 ($p_c=0.125$, $p_d=0.25$) and CREEP54 ($p_c=0.125$, $p_d=0.75$)	253

6.71	Offline performance (running average of best cost solutions) for GA runs CREEP4 (No creeping mutation, $p_c=0.0$) and CREEP24 (probability of creep, $p_c=0.125$, and probability of creeping down, $p_d=0.5$)	254
6.72	Offline performance for GA runs CREEP14 ($p_c=0.0625$, $p_d=0.5$), CREEP24 ($p_c=0.125$, $p_d=0.5$) and CREEP34 ($p_c=0.25$, $p_d=0.5$)	254
6.73	Offline performance for GA runs CREEP24 ($p_c=0.125$, $p_d=0.5$), CREEP44 ($p_c=0.125$, $p_d=0.25$) and CREEP54 ($p_c=0.125$, $p_d=0.75$)	254
6.74	Average generation costs for GA runs CREEP4 (No creeping mutation, $p_c=0.0$) and CREEP24 (probability of creep, $p_c=0.125$, and probability of creeping down, $p_d=0.5$)	255
6.75	Average generation costs for GA runs CREEP14 ($p_c=0.0625$, $p_d=0.5$), CREEP24 ($p_c=0.125$, $p_d=0.5$) and CREEP34 ($p_c=0.25$, $p_d=0.5$)	255
6.76	Average generation costs for GA runs CREEP24 ($p_c=0.125$, $p_d=0.5$), CREEP44 ($p_c=0.125$, $p_d=0.25$) and CREEP54 ($p_c=0.125$, $p_d=0.75$)	255
6.77	Online performance (running average of all solution costs) for GA runs CREEP4 (No creeping mutation, $p_c=0.0$) and CREEP24 (probability of creep, $p_c=0.125$, and probability of creeping down, $p_d=0.5$)	256
6.78	Online performance for GA runs CREEP14 ($p_c=0.0625$, $p_d=0.5$), CREEP24 ($p_c=0.125$, $p_d=0.5$) and CREEP34 ($p_c=0.25$, $p_d=0.5$)	256
6.79	Online performance for GA runs CREEP24 ($p_c=0.125$, $p_d=0.5$), CREEP44 ($p_c=0.125$, $p_d=0.25$) and CREEP54 ($p_c=0.125$, $p_d=0.75$)	256
7.1	Formation of a 48-bit string (from two 24-bit strings) representing trial solutions in the GA search for two independent Gessler network designs	260
7.2	The elitist model	264
7.3	Best generation costs for the GA run D1 - two Gessler problems	266
7.4	Average generation costs for the GA run D1 - two Gessler problems	267
7.5	Best generation costs for the GA run T1 - three Gessler problems	271
7.6	Average generation costs for the GA run T1 - three Gessler problems	272
7.7	Best generation costs for GA runs F1 (improved GA with elitism), F1' (improved GA without elitism) and F1'' (traditional GA) - five Gessler problems	277
7.8	Average generation costs for GA runs F1 (improved GA with elitism), F1' (improved GA without elitism) and F1'' (traditional GA) - five Gessler problems	278
8.1	The New York City water supply tunnels network in 1969	283
8.2	Coded strings representing the best GA designs	288
8.3	Best of generation costs for GA runs NY1 ($k=\$5\text{mill./ft}$) and NY5 ($k=\40mill./ft)	298
8.4	Average generation costs for GA runs NY1, NY3 and NY5	299
8.5	Fluctuations of number of infeasible solutions per population for GA runs NY1-NY5	301

8.6 Best of generation costs for GA runs NY13 (Improved GA with $N=500$) and NY23 (No elitism)	306
8.7 Average generation costs for GA runs NY13 (Improved GA with $N=500$) and NY23 (No elitism)	307
8.8 Best of generation costs for GA runs NY13 (Improved GA) and NY33 (Binary codes)	309
8.9 Average generation costs for GA runs NY13 (Improved GA) and NY33 (Binary codes)	310
8.10 Best of generation costs for GA runs NY13 (Improved GA) and NY43 (Raw fitness values)	312
8.11 Average generation costs for GA runs NY13 (Improved GA) and NY43 (Raw fitness values)	313
8.12 Best of generation costs for GA runs NY13 (Improved GA) and NY53 (No creep)	315
8.13 Average generation costs for GA runs NY13 (Improved GA) and NY53 (No creep)	316
8.14 Best of generation costs for GA runs NY13 (Improved GA) and NY63 (Traditional GA)	318
8.15 Average generation costs for GA runs NY13 (Improved GA) and NY63 (Traditional GA)	319
9.1 The layout of the Fort Collins - Loveland water supply system including the existing system and the system expansions proposed in the 1993 Master Plan	332
9.2 Pump curves for the Overland Trail source pump station	335
9.3 Pump curves for the Airport booster pump station	336
9.4 The Master Plan pipe network design and proposed PRV pressure settings for the Fort Collins - Loveland system expansion plan for the year 2015	341
9.5 The coding scheme for the GA optimisation of the FCLWD water distribution system expansion plans	344
9.6 The Genetic Algorithm pipe network design and proposed PRV pressure settings for the Fort Collins - Loveland system expansion plan for the year 2015	346
9.7 Zone 1 1.0MG Tank water level variation for EPS	359
9.8 Zone 2 4.0MG Tank water level variation for EPS	359
9.9 New Zone 3 4.0MG Tank water level variation for EPS	359
9.10 New McCloughan Hill Tank water level variation for EPS	360
9.11 Elevated Airport Tank water level variation for EPS	360
9.12 Timnath Tank water level variation for EPS	360

List of Tables

2.1 The pipes of the <i>Anytown</i> network	36
2.2 The nodes of the <i>Anytown</i> network	37
2.3 <i>Anytown</i> pump characteristics	37
2.4 The loops, initially assumed loop flow corrections and final loop flow corrections (after 6 iterations)	41
2.5 The initially assumed flows and the balanced flows in the <i>Anytown</i> network	44
2.6 Convergence of the Newton-Raphson method applied to the loop equations for the <i>Anytown</i> network	52
2.7 The unbalanced head losses around the loops	53
2.8 Potential pivot elements for the scaled Jacobian matrix for the <i>Anytown</i> network in Eq. 2.56	59
2.9 Chosen pivot order for Markowitz threshold pivoting	60
2.10 Chosen pivot order for modified Markowitz threshold pivoting	61
2.11 The balanced pressure heads at the junction nodes for the <i>Anytown</i> system	62
4.1 The representation of design parameters by pieces of code	104
4.2 Example schemata and subset members	110
5.1 Pipe connectivity, lengths, diameters and roughness coefficients	120
5.2 Node elevations, demands and associated minimum pressures	121
5.3 Available pipe sizes and associated costs	121
5.4 The best 50 solutions to the Gessler problem	123
5.5 The balanced pipe flows for global optimal solution 1	126
5.6 Junction node pressure heads for global optimal solution 1	127
5.7 Pipe groups (as used by Gessler, 1985)	127
5.8 Fitted cost functions for the nonlinear optimisation model	128
5.9 Solution from GINO nonlinear optimisation for the Gessler network	130
5.10 Decision variable choices and corresponding binary substrings	132
5.11 The starting population	134
5.12 Decoding String 9	135
5.13 Equivalent diameters and Hazen-Williams coefficients for the network solution represented by String 9	135
5.14 Determining the pipe cost of String 9	136
5.15 The balanced pipe flows	137
5.16 Comparison of actual and allowable node pressure heads	137
5.17 Calculation of the penalty cost for String 9	138

5.18	Cost and fitness of the strings in the starting population	138
5.19	Starting population cost statistics	139
5.20	The probability of survival	140
5.21	The second generation	141
5.22	Cost of the strings in the second generation	142
5.23	Second generation cost statistics	142
5.24	Distribution of 1,000,000 random numbers	146
5.25	Minimum cost network solution with varying random number seed	147
5.26	Minimum cost network solution with varying population size, N	155
5.27	Minimum cost solution with varying probability of crossover, p_c	156
5.28	Minimum cost solution with varying probability of mutation, p_m	159
5.29	Commonly identified infeasible pipe network configurations	162
5.30	The costs of the infeasible network solutions	163
5.31	The critical pressure head deficiencies for the infeasible solutions	163
6.1	The chosen GA parameter sets	167
6.2	Summary of the GA model runs performed in this chapter	168
6.3	Extra pipe costs for pipe [14] for a 1 psi improvement in pressure at the critical node 12 (for the optimal solution 1 subject to demand pattern GE3)	174
6.4	Variation of the penalty multiplier for GA model runs PEN61-PEN65	176
6.5	Search results for genetic algorithm model runs PEN1-PEN5	176
6.6	Search results for genetic algorithm model runs PEN11-PEN15	177
6.7	Search results for genetic algorithm model runs PEN21-PEN25	177
6.8	Search results for genetic algorithm model runs PEN31-PEN35	178
6.9	Search results for genetic algorithm model runs PEN41-PEN45	179
6.10	Search results for genetic algorithm model runs PEN51-PEN55	179
6.11	Search results for genetic algorithm model runs PEN61-PEN65	180
6.12	Variation of fitness scaling exponent, n for GA runs FIT41-FIT45 (inverse fitness function) and FIT51-FIT55 (linear fitness function)	192
6.13	Search results for genetic algorithm model runs FIT1-FIT5	194
6.14	Search results for genetic algorithm model runs FIT11-FIT15	194
6.15	Search results for genetic algorithm model runs FIT21-FIT25	195
6.16	Search results for genetic algorithm model runs FIT31-FIT35	195
6.17	Search results for genetic algorithm model runs FIT41-FIT45	196
6.18	Search results for genetic algorithm model runs FIT51-FIT55	196
6.19	Search results for genetic algorithm model runs FIT61-FIT65	197
6.20	Search results for genetic algorithm model runs FIT71-FIT75	197
6.21	Search results for genetic algorithm model runs FIT81-FIT85	198
6.22	Search results for genetic algorithm model runs FIT91-FIT95	198

6.23	Representation mappings	208
6.24	Ranked design parameters for the upgrade of existing pipe [1]	209
6.25	Ranked design parameters for the upgrade of existing pipe [4]	209
6.26	Search results for genetic algorithm model runs CODE1-CODE5	214
6.27	Search results for genetic algorithm model runs CODE11-CODE15	214
6.28	Search results for genetic algorithm model runs CODE21-CODE25	215
6.29	Search results for genetic algorithm model runs CODE31-CODE35	215
6.30	Search results for genetic algorithm model runs CROSS1-CROSS5	236
6.31	Search results for genetic algorithm model runs CROSS11-CROSS15	236
6.32	Search results for genetic algorithm model runs CROSS21-CROSS25	237
6.33	Search results for genetic algorithm model runs CROSS31-CROSS35	237
6.34	Search results for genetic algorithm model runs CROSS41-CROSS45	238
6.35	Search results for genetic algorithm model runs CROSS51-CROSS55	238
6.36	Search results for genetic algorithm model runs CROSS61-CROSS65	239
6.37	Search results for genetic algorithm model runs CREEP1-CREEP5	248
6.38	Search results for genetic algorithm model runs CREEP11-CREEP15	248
6.39	Search results for genetic algorithm model runs CREEP21-CREEP25	249
6.40	Search results for genetic algorithm model runs CREEP31-CREEP35	250
6.41	Search results for genetic algorithm model runs CREEP41-CREEP45	251
6.42	Search results for genetic algorithm model runs CREEP51-CREEP55	252
7.1	GA parameter sets D1-D5 for the optimisation of two Gessler problems	262
7.2	Variation of fitness scaling exponent, n for the GA runs D1-D5	263
7.3	Improved GA results for the optimisation of two Gessler problems	265
7.4	Occurrences of optimal solutions for GA run D1	268
7.5	Parameter sets T1-T5 for the optimisation of three Gessler problems	269
7.6	Variation of the fitness scaling exponent, n for the GA runs T1-T5	269
7.7	Improved GA results for the optimisation of three Gessler problems	270
7.8	A comparison of the various expanded solution spaces	273
7.9	Parameter sets F1-F5 for the optimisation of five Gessler problems	274
7.10	Variation of the fitness scaling exponent, n for the GA runs F1-F5	276
7.11	Results of the optimisation of five Gessler problems using the improved GA (with elitism)	276
7.12	Results of the optimisation of five Gessler problems using the improved GA (without elitism)	279
7.13	Results of the optimisation of five Gessler problems using the traditional GA	280
8.1	Nodal data for the New York City water supply tunnels	284
8.3	Existing tunnel data for the New York City water supply tunnels	285

8.3 Available tunnel sizes and construction costs for New York tunnels duplications and the corresponding coded substrings	286
8.4 The improved GA compared to the traditional GA formulation	287
8.5 Improved GA runs NY1-NY5 with varying penalty multiplier	296
8.6 Improved GA runs NY6-NY10 with varying penalty multipliers and new random number generator seed	297
8.7 Improved GA runs NY11-NY19	304
8.8 Improved GA runs NY21-NY29 (without elitism)	305
8.9 Improved GA runs NY31-NY39 (with substrings of binary codes)	308
8.10 Improved GA runs NY41-NY49 (not including fitness scaling)	311
8.11 Improved GA runs NY51-NY59 (not including creeping mutation)	314
8.12 Traditional GA runs NY61-NY69	317
8.13 The five lowest cost feasible GA designs and three low cost infeasible GA designs	320
8.14 Hydraulic heads for GA designs	321
8.15 Balanced tunnel flows for design GA(1)	322
8.16 Balanced node hydraulic heads for design GA(1)	323
8.17 Previous studies of the New York City tunnels problem	324
8.18 Designs achieved by previous studies	325
8.19 Hydraulic heads for previous designs using KYPIPE	327
8.20 Hydraulic heads for previous designs using the simulation model developed in this research	328
9.1 Pump station operation for the 2015 peak hour demands	335
9.2 Storage tank water levels for the 2015 peak hour	338
9.3 Mapping of values of decision variables to corresponding integer code	340
9.4 The Master Plan pipe network design for 2015	342
9.5 The GA design pipe network design for 2015	347
9.6 Summary of pipe costs	348
9.7 Master Plan recommendations for PRVs	349
9.8 Range of pressure settings considered in GA optimisation for PRVs	350
9.9 Pressure reducing valves and selected pressure settings	351
9.10 Summary of low pressures for the proposed designs	353
9.11 Pressure zone inflows for the 2015 peak hour demands	354
9.12 Demand variation assumed for EPS of the 2015 maximum day	357
9.13 Tank dimensions and initial water levels assumed for the EPS	357
9.14 Tanks available operating storage	358
9.15 Tank water level variations for the EPS of the 2015 maximum day	358
9.16 Pump station operation for the EPS analyses	361
9.17 Average pump station power output for the 2015 maximum day	362

9.18	The HGL of sources of supply for EPS	363
9.19	Estimated total supplies from the sources of supply for EPS	363
A1	Tank and reservoir input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands	388
A2	Node input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands	389
A3	Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands	394
A4	Pump station (PS) input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands	401
A5	Pressure reducing valve (PRV) and flow control valve (FCV) input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands	401
B1	Tank and reservoir input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands	402
B2	Node input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands	403
B3	Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands	408
B4	Pump station (PS) input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands	415
B5	Pressure reducing valve (PRV) and flow control valve (FCV) input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands	415

Principal Notations

Many of the symbols that are commonly used throughout the thesis are presented below. Each symbol is also defined when first encountered in the text.

A = cross-sectional area

A_p, B_p = constants for a pump p

BFS = breadth first search

C_j = Hazen-Williams roughness coefficient of pipe j

CV = check valve

D_j = diameter of pipe j

D_e = equivalent diameter of pipe

$\Delta E_{mn} = E_m - E_n$ = elevation difference between fixed-grade nodes m and n for pseudo loops

EGL = energy grade line

EPS = extended period simulation

f = Darcy-Weisbach friction factor

f_i = raw fitness of coded string i

f_i' = scaled fitness of coded string i

FCLWD = Fort Collins - Loveland Water District

FCV = flow control valve

g = acceleration due to gravity

GA = genetic algorithm

h_{fj} = friction head loss in pipe j

h_p = pumping head

H_i = total hydraulic head (pressure head+elevation head) at node i

H_{mini} = minimum allowable head at node i

H_0 = pump shut-off head

HGL = hydraulic grade line

J = the Jacobian matrix of partial derivatives (evaluated at the values of the flow corrections for the loop flow correction equations)

k = pressure violation penalty multiplier

L_j = length of the pipe j

LP = linear programming

n = fitness scaling exponent (in the fitness function)

N = population size (the working population)

N' = elite population size

NF = number of fixed-grade nodes

$(NF-1)$ = number of pseudo loops

NJ = number of junction nodes

NL = number of natural loops

NLP = nonlinear programming

NP = number of pipes

NPJ = number of pipes connected to a given junction node

NPL = number of pipes forming a given loop

NPP = number of parallel pipes

NPR = number of pipes in a path from a fixed-grade node to a given junction node

p_a = the probability of creeping mutation

p_c = the probability of crossover

p_d = the probability of creeping down

p_e = the probability of an elite mate

p_i = the probability of selection of string i

p_m = the probability of random bit-wise mutation

p_t = probability that the fittest individual is selected as the winner in a binary tournament

P_p = pump power input

PC = penalty cost

PC_{max} = penalty cost based on the maximum violation of the pressure constraints

PC_{sum} = penalty cost based on the sum of the violations of the pressure constraints

PRV = pressure reducing valve

PS = pump station

$q = (q_1, q_2, \dots, q_{NL+NF-1})$ = flow corrections for each natural loop and pseudo loop

$q^{(m)}$ = computed flow corrections for iteration m

Δq_{avg} = specified accuracy of average change in flow corrections between iterations

Δq_{max} = specified accuracy of maximum change in flow corrections between iterations

Q_j = flow in pipe j

Q_{exi} = demand at node i

Q_p = pump flow

R_j = resistance or loss coefficient of pipe j

R = Reynolds Number

s = tournament size (tournament selection)

S_j = hydraulic gradient for pipe j

V = pipe velocity

z_i = elevation of node i

γ = unit weight of water

η_p = pump efficiency

ν = viscosity



1 Introduction

This thesis presents an application of the relatively new and powerful genetic algorithm search to the problem of the optimisation of the design (and operation) of water distribution systems.

1.1 Water Distribution Systems

A water supply system is a fundamental component of the infrastructure of a community. A water supply system for water transmission and distribution is essentially a network of pipes connecting sources (such as reservoirs, wells or connections to adjacent systems) of water to demand points (nodes). To facilitate efficient and reliable operation, the sources are connected to the demand nodes via a complex arrangement of system components such as pipes, pumps, balancing tanks (often used to store water for peak demand periods and emergency conditions), and control devices such as pressure reducing valves (often used to separate a system into pressure zones).

1.2 System Expansions

A community's water use patterns increase and diversify with population and economic growth. The water supply authority responsible for the ongoing maintenance and operation of a water distribution system assesses the reliability and the quality of service provided by their system and periodically updates plans for future system expansions to meet projected future demands. Recommendations for future system expansions may include a host of design and operational decisions such as:

- the identification of future sources of water such as wells
- the upgrade of water treatment facilities (since the capacity of the source is constrained by the capacity of the water treatment facility)
- the sizing of pumps and their pumping schedules
- the settings and operating rules of pressure regulating valves
- the expansion or rehabilitation of a water distribution pipe network,
- the location and capacity of system storages

The design of future system expansions and / or changes to the operation of existing facilities to meet changing demand patterns is the responsibility of the water authority. For complex city systems, and even for simple residential subdivisions sub-systems or irrigation systems, the number of alternative designs is very large. The number of combinations of options to meet the

changing demands in the best way is often so large that it is impossible to consider all possible solutions.

1.3 Hydraulic Simulation of Water Distribution Systems

A prerequisite step to optimisation is the accurate hydraulic simulation of proposed pipe network designs to assess hydraulic feasibility. A hydraulic simulation model predicts water distribution system behaviour, including flow and pressure distributions, given some instantaneous or time-dependent pattern of water demands.

The hydraulic simulation of a water distribution system is itself a complex mathematical problem. Methods of hydraulic analysis are reviewed in this thesis in Chapter 2 in a search for a reliable and efficient analysis. The hydraulic simulation model is a very important part of the genetic algorithm formulation developed in this thesis.

There are a number of commercial packages available for the hydraulic analysis of water distribution systems. In engineering practice, the designer develops a pipe network design by considering a handful of trial and error hydraulic simulations of proposed designs using a commercial hydraulic simulation package. Proposed designs are based on local knowledge of the design and operation of the system, engineering judgement, design guidelines and rules of thumb. The designer uses a simulation model to determine workable solutions which are then compared in terms of cost, reliability and other objectives.

1.4 Optimisation of the Design of Water Distribution Systems

The optimisation of a water distribution system design attempts to achieve the best possible design (and operation) for a specified level of system performance such that system expansion costs (and life-cycle operating costs) are a minimum. The pipe network optimisation problem is defined in Chapter 3.

Usually, the foremost objective of the optimisation is to minimise the initial construction costs of new system components and the present value of system operation costs for the lifetime of the design. Millions of dollars are spent annually expanding or rehabilitating water supply system infrastructure. Jeppson (1985) reported the 40,000 water services in the U.S.A. had invested \$200 billion in water supply facilities and were investing in new facilities at the rate of \$2 billion per year. The ongoing costs of operating water supply systems are also significant (and will depend on the system layout and the capacity of system components). In the U.S.A., the electricity consumed by water supply utilities (for pumping water) makes up about 7% of all the electricity consumed in the country (Ormsbee et al., 1989).

1 Introduction

In addition to network design and operating costs, there are usually several other competing objectives (some non-quantifiable) such as system reliability and possible future system expansions that should be considered in the optimisation.

The decision variables of the optimisation are the physical and operational characteristics of the system components to be defined in the design such as: the pipe network layout; the diameters of new pipes; the cleaning, duplication or deletion of existing pipes; the introduction of new pump station installations or the upgrade of existing pump stations; the capacity and proposed operating policies for individual pump units; the location, volume and operating water levels of new storage tanks; and valve settings.

The water distribution system design is subject to a series of demand conditions at the nodes such as: peak instantaneous flows; emergency flows for fire fighting or in the event of a pipe breakage; and/or time-dependent peak day or peak week demand patterns. The design is required to exhibit a specified system performance for the demand patterns considered. Some system performance constraints which may be considered include: minimum allowable pressure heads achieved at the nodes for all demand conditions; tanks refill by the end of some demand period in preparation for the next demand period; and pumps operate within their limits of operation. A design which does not meet the system performance constraints is said to be an infeasible solution.

1.5 Pipe Network Optimisation Techniques

The optimisation of pipe networks is a problem which has received more attention in recent years. In Chapter 3 of this thesis, a selection of models are reviewed, covering a broad range of approaches to pipe network optimisation. Some models formulate the pipe network optimisation problem using traditional mathematical optimisation techniques such as linear programming, nonlinear programming or dynamic programming. There are other innovative models that are essentially random search methods or enumeration algorithms. Many of the models are hybrid schemes of two or more methods and the models often incorporate heuristic processes. Lansey and Mays (1989b) and Walski (1985) provide comprehensive reviews of the development of pipe network optimisation models in the last 25 years.

Most of the approaches presented in the literature first concentrate on the simpler problem of the optimisation of a gravity-fed distribution network of pipes subject to one critical instantaneous demand pattern and then make recommendations for the treatment of complexities such as pumped systems, multiple demand patterns and the design of other system components such as storage tanks. Often assumptions and simplifications are required due to the complicated nature of the pipe network optimisation problem.

Optimisation approaches which are linked to stand-alone hydraulic simulation models are convenient as proposed designs can be simulated and hydraulic feasibility evaluated in the same way a designer manually simulates and confirms a workable design.

1.6 Genetic Algorithms (GAs)

A genetic algorithm (GA) model for the practical optimisation of the design and operation of water distribution systems is developed in this thesis. The framework for the GA search model is represented by the flowchart of processes in Figure 1.1.

The GA search is a simplified simulation of the evolution process. Evolution is the established optimisation process used by nature, by which species grow and develop from earlier forms and adapt to their environment. A firm theoretical basis for genetic algorithms was established by Holland (1975).

In the natural evolution of a species, a chromosome of genetic information characterises a unique individual. Similarly, in the GA search, trial solutions to the search or optimisation problem are represented by a unique coded structure such as a binary string of 1's and 0's.

In nature, according to Darwin's survival-of-the-fittest philosophy, an individual's chances of survival and reproduction are ultimately regulated by the fitness of the individual. The fitness may be measured relative to the conditions imposed by the environment. The prospect of reproduction can be measured by the fitness of the individual relative to the fitness of fellow members within the competing population. In the artificial evolution of the GA, coded solutions are assigned a value of fitness which measures the worth of the solution relative to a set of objectives.

The GA search employs operators such as selection, crossover and mutation which simulate Darwin's rules of natural selection and genetic mechanisms acting on an evolving population of coded structures. The selection or reproduction operator selects parent coded structures from the current population with some chosen preference to fitter solutions. The genetic code of selected parent coded structures is combined to form offspring coded structures for the new population by the crossover operator. Occasionally, the mutation operator applies subtle genetic variations to the offspring code. Crossover and mutation imitate the exchange of genetic information and the minor variations that occur to genetic information from parent to child.

In nature, useful developments and adaptations are inherited and stored in chromosomes which carry the blueprint of a living thing. In the GA, small pieces of useful code are reproduced and combined with other small pieces of useful code to produce longer pieces of highly fit code.

Over many generations, a population of coded solutions with average fitness (the starting population may be randomly selected) evolves to a population of highly fit coded solutions.

The traditional GA formulation (Holland, 1975) is reviewed in Chapter 4 of the thesis. The simple, yet powerful traditional GA considers populations of strings coded in the binary alphabet and three standard GA operators of selection, crossover and mutation. DeJong (1975) demonstrated the far-reaching possibilities of GAs for function optimisation by applying the traditional GA and some variations to a diverse set of solution spaces (including discontinuous, many-peaked and highly-dimensionality spaces). The simple, robust nature of the GA formulation makes it suitable for a number of applications - some applications for which the best solutions may be difficult to obtain using traditional optimisation techniques. Goldberg (1989) presented a comprehensive review of genetic-based techniques and their applications, and an analysis of the mechanics and the fundamental theory of GAs. Some GA applications of particular interest to this research are reviewed in Chapter 4.

1.7 A Traditional GA for Pipe Network Optimisation

There is a good degree of freedom in the way a GA model may be formulated. DeJong (1985) presents an overview of the issues facing researchers implementing a GA model for a new application area, including the choice of an appropriate coding representation, fitness functions, GA operators and parameters. A traditional GA approach to the pipe network optimisation problem is presented in Chapter 5.

A coding scheme is selected to represent pipe network design solutions as unique coded strings of finite length which simulate chromosomes. The decision variables of the optimisation are represented by genes and the genes are assigned positions in the coded string. A set of unique symbols for a gene (string position) maps to the choices for the corresponding decision variable such as available pipe diameters or allowable PRV settings. The traditional GA uses substrings of binary codes to represent decision variable choices for the pipe network design.

A fitness value provides information about a string's fitness to produce offspring. The fitness of a pipe network design is measured relative to objectives such as low cost and adequate hydraulic performance. The cost of proposed designs may be estimated and penalty costs may be applied to infeasible designs which do not achieve a specified level of system performance. The evaluation scheme of the GA model (in Figure 1.1) is linked to the hydraulic simulation model (developed in Chapter 2) which tests the hydraulic feasibility of the proposed design.

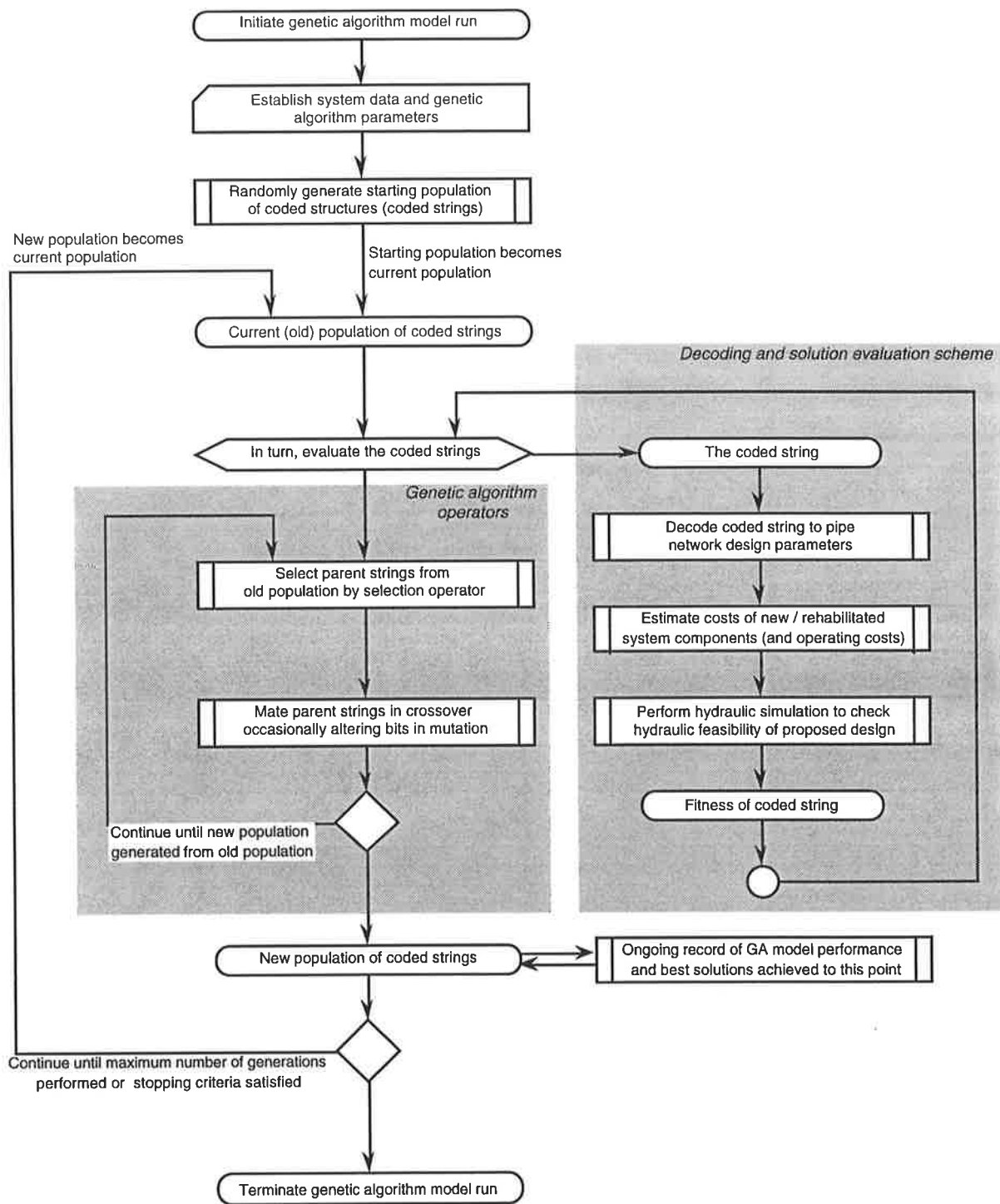


Figure 1.1 Flow chart for the proposed genetic algorithm model

The GA model operates with a population of coded solutions at any time. The initial population of coded strings is usually randomly generated. GA operators generate new populations using the code and fitness information of coded solutions in the old population. GA operators such as proportionate (roulette-wheel) selection, one-point crossover and random bit-wise mutations are employed by the traditional GA. The GA operators and GA parameters such as population size, probability of crossover and of mutation are selected to control and guide the GA search. Simpson, Dandy and Murphy (1994) presented a detailed procedure for applying the traditional GA approach to optimise a relatively simple hypothetical pipe network design which is referred to as the two-reservoir Gessler network.

1.8 The Two-Reservoir Gessler Network

The two-reservoir Gessler network (Gessler, 1985) is the case study chosen to investigate the application of the GA model to pipe network optimisation. The 14-pipe Gessler network introduced in Chapter 5 is a looped, gravity-fed distribution system. The Gessler network expansions require the sizing of five new pipes and the upgrade (cleaning, duplication or 'do nothing') of three existing pipes. The network expansions are required to satisfy three demand patterns, including a peak loading condition and two emergency loading conditions.

It was feasible to perform an exhaustive enumeration of every possible pipe network design solution (about 4 million) for the Gessler problem. The exhaustive enumeration identified the two global optimum solutions and other characteristics of the solution space such as relative proportions of feasible and infeasible solutions, critical nodes in the system and critical demand patterns. Gessler (1985) used a partial enumeration of a pruned search space of about 900 combinations to optimise the problem.

First, a small-scale GA is applied by hand to the Gessler problem. A small population size (only 10 members) is used for a close examination of the operations of the GA search. Then, full-scale GA model runs are performed with realistic population sizes for the application to the Gessler problem. The GA runs were allowed a maximum of 10,000 new solution evaluations (for example, about 100 generations of a population of 100 coded string members). The least cost solutions determined by the GA may be compared with the best solutions identified by the exhaustive enumeration. The traditional GA is found to be effective, but it becomes apparent that some modifications may improve performance.

1.9 Modifications to the Traditional GA

Many researchers have found it necessary to experiment with variations of the traditional GA, introducing innovative coding schemes, alternative fitness evaluation schemes and advanced GA operators to tailor the GA to a specific problem. Although, there is considerable freedom to express and formulate the GA search, it would be unwise to depart too far from the theoretical foundations of the GA search established by Holland (1975).

Variations of the traditional GA are applied to the search for the known optimal solutions to the Gessler network expansions problem in Chapter 6. The performance of specific elements of the GA formulation is observed, including various penalty functions and fitness functions and alternative coding representations, parent selection methods, crossover and mutation mechanisms. The experiments suggest the operators, coding and evaluation schemes likely to lead to improved performance in the search of the solution space to the pipe network optimisation problem. An improved GA formulation for the application to pipe network optimisation begins to emerge as a result of the study of the Gessler problem in Chapter 6.

1.10 Multiple Gessler Problems

The exhaustive enumeration of the relatively small Gessler problem (14-pipe network) in Chapter 5 positively identifies the lowest cost solutions. The effectiveness of various forms of the GA is measured in Chapter 6 by the ability to find these solutions. Larger water system design optimisation problems with many decision variables are required for further development and testing of the GA application. As problems increase in size and complexity, it soon becomes impossible to perform an exhaustive enumeration.

In Chapter 7, a larger pipe network optimisation problem with known global optimum solutions is devised by considering the simultaneous optimisation of two Gessler problems. The coded string solutions are separated into two component substrings for evaluation, representing two solutions to independent Gessler problems. Even larger solution spaces are manufactured by considering the simultaneous optimisation of three and five Gessler problems. The improved GA, incorporating many of the recommendations of Chapter 6 is applied to these problems. In addition, an elitist concept (DeJong, 1975; Goldberg, 1989) is introduced to the GA model. A population of the best solutions obtained in earlier generations are maintained in a parallel elite population and elite mates are occasionally crossed with members of the working population.

1.11 The New York City Water Supply Tunnels

In Chapter 8, the GA model is applied to the optimisation of the expansions of the New York City tunnels network. The classic New York tunnels problem was first introduced and optimised by Schaake and Lai (1969). Since then, the New York tunnels network has become a benchmark for researchers of pipe network optimisation techniques.

The five least cost feasible designs and three low cost infeasible designs identified by the GA are presented. The designs are compared with the designs obtained by traditional optimisation methods such as linear programming, nonlinear programming and enumeration techniques. There are estimated to be 1.93×10^{25} possible (discrete tunnel size) solutions to the New York tunnels problem.

The performance of the traditional three-operator GA and the new GA developed in Chapters 6 and 7 and various intermediate GA formulations are compared for the application to the New York tunnels problem. Dandy, Simpson and Murphy (1996a) presented an improved GA approach for the application to the New York tunnels problem (the GA used decision variable substrings of Gray codes, fitness scaling and decision-variable-wise creeping mutations, but not the elitist concept).

1.12 The Fort Collins - Loveland System Expansion Plan

Ultimately, the objective of this research is to develop the GA model for pipe network optimisation as a practical design tool. A final case study is intended to demonstrate the usefulness of the GA approach in a realistic design situation. In Chapter 9, the GA model is applied to the optimisation of aspects of the expansion plan for the Fort Collins - Loveland water transmission and distribution system.

The system provides water to an area of about 60 square miles between the cities of Fort Collins and Loveland in Colorado, U.S.A. The system will require expansions to meet anticipated increased agricultural and municipal water needs. A Master Plan prepared recently by a local engineering consultant predicted future water demands and outlined a proposed system expansion plan for 2015. The Master Plan design was determined using design guidelines, experience and a hydraulic simulation model.

The GA optimises the diameters of new and duplicate pipes and pressure reducing valve (PRV) pressure settings. The Fort Collins - Loveland water system is a complex system of source and booster pump stations, storage tanks, about 330 pipes (of which 49 are proposed new or duplicate pipes) and 13 major PRVs which isolate the system into 5 major pressure zones.

1 Introduction

The water is supplied from several alternative sources of supply. The GA model helps to identify the preferred water sources by considering increasing the supply (up to allowable limits) from the connections to adjacent city systems. The Fort Collins - Loveland system expansion plan presents the GA model with many real problems facing optimisation models for pipe network design. The design generated by the GA model is compared to the original Master Plan design.

The GA model is reconstructed for the Fort Collins - Loveland system expansion problem to link with a commercial hydraulic simulation package. The simulation model is employed to confidently test the hydraulic feasibility of proposed expansions to the complex system of multiple pressure zones.

2 Development of a Hydraulic Simulation Model

2.1 Introduction

A water distribution pipe network supplies some pattern of flows to demand points in the network. A hydraulic simulation (or hydraulic analysis) of a pipe network involves the prediction of system behaviour (such as pipe flows, nodal pressures and tank water level variations) for a given system configuration and operation, subject to some expected demand condition. The accurate simulation of a water distribution system to assess hydraulic performance is a prerequisite step to pipe network optimisation. The genetic algorithm (GA) model for pipe network optimisation developed in this thesis incorporates a simulation model to evaluate the hydraulic feasibility of trial designs generated by the GA search.

At present, hydraulic simulation models have a much higher use in practice than pipe network optimisation models. There now exist a number of extensively tested hydraulic simulation models that perform efficient analyses of the behaviour of complex water distribution systems. It makes sense to utilise the well developed capabilities of hydraulic simulation models as a first step towards the problem of optimisation. A number of effective pipe network optimisation techniques are linked to stand-alone hydraulic simulation models to test for hydraulic feasibility including linear programming (Morgan and Goulter, 1985), nonlinear programming (Lansey and Mays, 1989a) and enumeration procedures (Gessler, 1982).

The evaluation scheme of the GA model should be linked to an efficient and reliable method of hydraulic analysis. In this chapter, a hydraulic simulation model is developed specifically for the purpose of the development of the GA model.

In the GA evolution of a population of coded solutions representing pipe network designs, hundreds of thousands of pipe network designs may be generated and evaluated. It is most important that the simulation model is efficient, since the hydraulic analyses of trial pipe network designs usually requires the most significant proportion of the total computational effort for the GA search. The pipe network simulation model is integrated with the GA optimisation model in one computer program in order to minimise computational times.

The hydraulic solution method should display good convergence characteristics in all situations, since the GA can propose unusual pipe network designs (more often in early generations). The simulation model should be capable of performing multiple hydraulic analyses for different demand patterns including instantaneous peak loading cases, loadings for emergency conditions such as pipe breakage or fire fighting needs, and maximum day or average day extended period

2 Development of a hydraulic simulation model

simulations. The results of the simulations are used to check that water is supplied with an acceptable pressure. The simulation model can be used to check other hydraulic constraints such as acceptable pipe velocities, to estimate annual power costs for pumping or to check the fluctuations of storage tank water surface levels over a period of time. The simulation model could potentially be modified to simulate and check other conditions such as pressure surges or water quality constraints.

The alternative to developing a hydraulic simulation model from the ground up is to create an external link between the GA evaluation scheme and a stand-alone commercial hydraulic simulation package such as WATSYS (Olde, 1985), KYPIPE (Wood, 1974) or EPANET (Rossman, 1994). If the source code for a well developed and tested simulation model is available, it may be more practical and efficient to embed the hydraulic simulation code within the code of the GA routines.

A hydraulic analysis is essentially the determination of the steady-state pipe network flows and node pressures for a defined system configuration and operation, subject to a pattern of instantaneous nodal demands. The behaviour of the system over a period of time (extended period simulation) is usually determined from a series of steady-state hydraulic analyses performed at suitable time intervals.

The steady-state behaviour of the system is governed by the physical laws of mass continuity at the junction nodes and conservation of energy around the loops of the network, and the nonlinear head loss / flow relationship in the pipes. The hydraulic analysis of a pipe network for steady-state conditions reduces to the solution of a set of pipe network equations which may be formulated as:

- pipe flow equations (Q -equations) constructed in terms of unknown pipe flows
- node equations (H -equations) constructed in terms of the unknown nodal heads
- loop corrective flow equations (ΔQ_L -equations) constructed with flow corrections around the loops as the unknowns

The equations form a set of simultaneous nonlinear algebraic equations which cannot be solved directly. A number of methods have been proposed to solve this system of equations including:

- the Hardy Cross method
- the Newton-Raphson method
- the linear theory method

2 Development of a hydraulic simulation model

In this chapter, the pipe network equations are formulated and proposed solution methods are reviewed in the search for an efficient and reliable method of hydraulic analysis. The adopted method of hydraulic analysis is presented and the implementation of the method and various algorithms which are developed to supplement the method are described for a case study hydraulic simulation.

A hydraulic simulation model is developed which can perform a steady-state analysis of a looped pipe network which contains system components such as reservoirs, tanks and pumps. The inclusion of other system components such as pressure regulating valves and the modification of the model for extended period simulation (EPS) is discussed briefly. Finally, time-saving techniques are described which speed up the hydraulic simulation of systems for the GA search, particularly by exploiting the results of previous simulations.

2.2 Pipe Network Components

A pipe network is composed of a layout of nodes interconnected by pipes. The pipes provide the principal framework for the water transmission and distribution system. The transmission pipelines convey large flows from water sources to the balancing storages and distribution pipes then convey the flows to the water demand points.

A pipe is characterised by physical properties such as length, roughness and diameter. The roughness of the pipe inside wall may depend on the material, size and condition of the pipe. The condition of the pipe is influenced by time in operation and water quality. Common pipe materials include DICL (ductile iron concrete lined), CICL (cast iron concrete lined), RC (reinforced concrete), HOBAS and uPVC. The pipe size may be indicated by a nominal diameter or an internal diameter.

The nodes of the pipe network are either junction nodes (demand nodes) or fixed-grade nodes (source nodes). The elevations of the nodes are determined from network topography. The demands at junction nodes are determined by forecasted consumer water needs. The demands at the junction nodes should be supplied with a pressure above some specified minimum (and below some allowable maximum pressure). A fixed-grade node is a point where a constant pressure head is maintained such as a reservoir, storage tank or connection to a constant pressure region. The water surface level at some nodes (such as elevated storage tanks) which are assumed to be fixed-grade nodes for the steady-state hydraulic analysis may actually vary with time. Elevated storage tanks are used to help smooth peak water demand periods and store water for emergency water demands such as fire fighting needs.

2 Development of a hydraulic simulation model

Pipe networks may contain flow devices such as pumps, check valves and pressure reducing valves. Source pumps are used in association with water sources such as tanks, reservoirs, wells and rivers, and connections to adjacent systems. Booster in-line pumps are used within a length of pipe in the network. Pumping station facilities usually consist of multiple stages of centrifugal (radial-flow) pumps.

Check valves only allow flow in one direction. Check valves are installed in the discharge line of pumps to prevent flow against the direction of pumping. A pressure reducing valve (PRV) maintains a constant pressure on the downstream side of the valve assuming upstream pressure is greater than or equal to the valve pressure setting. Should the pressure upstream of a PRV be less than the valve pressure setting, then the PRV has no effect and flow through the valve is unrestricted. Should the pressure at the downstream side of a PRV be greater than the upstream pressure, then the PRV acts as a check valve and closes to prevent reverse flow. PRVs are used to reduce the pressure in regions of the system, to adjust flow distributions or to control the sources of supply (El-Bahrawy and Smith, 1987). Other flow devices include flow control valves (FCVs) used to maintain a set rate of flow, pressure sustaining valves (PSVs) used to maintain a set upstream pressure and throttle control valves or orifice plates.

2.3 Tree Networks and Looped Networks

Pipe networks may be tree networks or they may contain loops. A tree network (or branched network) is such that any two nodes are connected by one and only one path of pipes. There is always a unique flow pattern in a tree network governed by the law of mass continuity. The hydraulic analysis of looped networks is more complicated since the flow may be conveyed to a single demand node by several alternative routes. The flow pattern throughout the looped pipe network is governed by the physical laws of mass continuity and energy conservation.

Irrigation systems are often branched networks. Looped networks are usually preferred to provide some degree of system reliability and flexibility. Urban water supply systems are usually extensively looped. Loops can help meet downstream demands in the event of a pipe breakage or pipe maintenance. In addition, looped networks do not allow water to become stagnant in dead-end pipes and have greater flexibility to meet abnormal demands such as fire fighting demands.

2.4 Natural Loops and Pseudo Loops

Natural loops (or primary loops) in a pipe network are identified as any closed path of pipes that does not contain another closed path of pipes within it.

Pseudo loops (or paths) are formed for systems with multiple fixed-grade nodes to account for unknown outflows and inflows at the storages (Streeter and Wylie, 1981). If there is only one fixed-grade node, then there are no pseudo loops since the total network demand is carried by one source. Pseudo loops are created by the introduction of an imaginary link between pairs of fixed-grade nodes. The pseudo loops are treated in the hydraulic analysis in a similar way to natural loops by taking the head loss in the imaginary link as the difference in elevation between the fixed-grade nodes.

Wood and Rayes (1981) state the following general relationship for a pipe network:

$$NP = NJ + NL + (NF - 1) \quad (2.1)$$

in which NP = number of pipes, NJ = number of junction nodes, NL = number of natural loops, NF = number of fixed-grade nodes and where $(NF - 1)$ = number of pseudo loops that need to be considered for the hydraulic analysis.

There are actually $NF! / [(NF - 2)! 2!]$ possible pseudo loop configurations for a system containing $NF > 1$ fixed-grade nodes, however, only $(NF - 1)$ pseudo loops need to be considered to formulate the equations for the hydraulic analysis. The $(NF - 1)$ selected pseudo loop configurations should involve each of the NF fixed-grade nodes (at least once).

2.5 Hydraulic Grade Line (HGL)

The water at any point in the water distribution system possesses an energy equal to the sum of the elevation head, pressure head and velocity head at the point. The velocity head and minor head losses in system components are usually small compared to head losses due to friction for long pipes and therefore these are assumed to be negligible in this case. This implies the hydraulic grade line (HGL) equals the energy grade line (EGL). The pressure head at a junction node is the energy possessed by the water above the elevation of the junction node (measured above some arbitrary datum).

The hydraulic grade line (HGL) is the profile of the variation of head across network components such as pipes as shown in Figure 2.1.

2 Development of a hydraulic simulation model

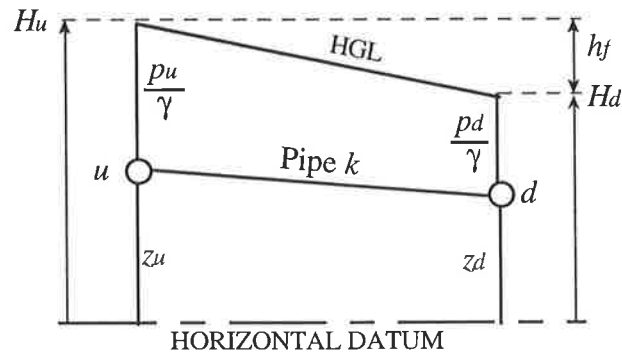


Figure 2.1 The HGL for a length of pipe

The pressure head at a reservoir is equal to the depth of water and the HGL is the reservoir surface as shown in Figure 2.2.

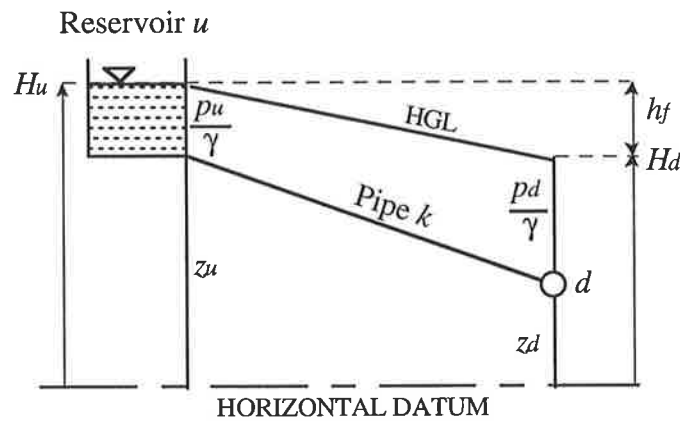


Figure 2.2 The HGL at a reservoir

According to the Bernoulli equation in Eq. 2.2 (neglecting velocity heads) and demonstrated in Figures 2.1, 2.2 and 2.3, the HGL at some downstream point d equals the HGL at an upstream point u less head losses due to friction ($h_{f_{u-d}}$) plus head gains due to pumping ($H_{p_{u-d}}$) between u and d . The Bernoulli equation is based on the physical law of conservation of energy between points u and d .

$$z_u + \frac{p_u}{\gamma} = z_d + \frac{p_d}{\gamma} + h_{f_{u-d}} + H_{p_{u-d}} \quad (2.2)$$

in which z_u = elevation of node u (ft, m), z_d = elevation of node d , p_u = pressure at node u (psi, Pa), p_d = pressure at node d and γ = specific unit weight of water (62.4 lb/ft³ at 60°F, 9,789 N/m³ at 15°C).

2.6 Pipe Friction Head Losses

The flow in a pipe results in an energy loss or head loss due to friction over the length of the pipe. Friction head loss in a pipe is expressed in terms of the flow by the following relationship:

$$h_f = R Q |Q|^{n-1} \quad (2.3)$$

in which h_f = friction head loss in pipe (ft, m), Q = pipe flow (cfs, m³/s), R = the resistance or loss coefficient for the pipe which is a function of pipe parameters and flow conditions and depends on the head loss equation and the units used, and n = the exponent in the head loss equation.

The equation chosen to relate head loss to flow will depend on the pipe roughness data available for the pipe network under study. The widely used Hazen-Williams equation is of the form of Eq. 2.3 with resistance R given by Eq. 2.4 or Eq. 2.5 and with $n=1.852$ (Streeter and Wylie, 1981).

$$R = \frac{4.73 L}{C^{1.852} \left(\frac{D}{12}\right)^{4.8704}} \quad (\text{US customary units}) \quad (2.4)$$

$$R = \frac{10.67 L}{C^{1.852} \left(\frac{D}{1000}\right)^{4.8704}} \quad (\text{SI units}) \quad (2.5)$$

in which L = length of the pipe (ft, m), C = Hazen-Williams roughness coefficient and D = diameter of pipe (in, mm).

The Darcy-Weisbach head loss formula is also commonly used, and iterative pipe roughness values are calculated in this case since the Darcy-Weisbach friction factor is a function of the flow in a pipe. The Darcy-Weisbach head loss equation computes the friction head losses in a pipe, h_f (m) for a pipe velocity, V (m/s).

$$h_f = \frac{f L V^2}{D 2 g} = \frac{f L}{D 2 g} \frac{Q^2}{A^2} = \frac{f L}{D^5} \frac{8 Q^2}{g \pi^2} \quad (2.6)$$

in which f = Darcy-Weisbach friction factor, L = pipe length (m), D = pipe diameter (m), Q = pipe flow (m³/s) and A = pipe cross-sectional area (m²).

2 Development of a hydraulic simulation model

The pipe roughness may be expressed in terms of the Colebrook-White roughness k (mm). The Swamee-Jain formula (Swamee and Jain, 1976) in Eq. 2.7 is used to determine the Darcy-Weisbach friction factor f from the Colebrook-White k and Reynolds number $R = VD / \nu$ where $\nu =$ viscosity (1.007×10^{-6} m²/s for water at 20°C).

$$f = \frac{1.325}{\left[\ln \left(\frac{k}{3.7D} + \frac{5.74}{R^{0.9}} \right) \right]^2} \quad (2.7)$$

2.7 Parallel Pipes

Two or more pipes may be connected in parallel between the same two nodes. This situation often occurs when existing pipe systems are reinforced to meet increased demands by installing new pipes (of some selected pipe size) parallel to existing pipes (called pipe 'duplication'). The parallel pipes may be regarded as one equivalent pipe with equivalent hydraulic properties for the hydraulic analysis. The flow in the equivalent pipe, Q_e is the sum of the constituent flows in NPP parallel pipes:

$$Q_1 + Q_2 + \dots + Q_{NPP} = Q_e \quad (2.8)$$

The relationship in Eq. 2.9 is obtained by substituting the friction head loss equation in Eq. 2.3 into Eq. 2.8.

$$\left(\frac{h_{f1}}{R_1} \right)^{\frac{1}{n}} + \left(\frac{h_{f2}}{R_2} \right)^{\frac{1}{n}} + \dots + \left(\frac{h_{fNPP}}{R_{NPP}} \right)^{\frac{1}{n}} = \left(\frac{h_{fe}}{R_e} \right)^{\frac{1}{n}} \quad (2.9)$$

The physical law of energy conservation is observed between the nodes and therefore the energy losses across the parallel pipes must be equal:

$$h_{f1} = h_{f2} = \dots = h_{fNPP} = h_{fe} \quad (2.10)$$

Consequently, the equivalent pipe may be assigned a resistance R_e which is a function of the resistances of the NPP parallel pipes as given by Eq. 2.11.

$$\left(\frac{1}{R_1} \right)^{\frac{1}{n}} + \left(\frac{1}{R_2} \right)^{\frac{1}{n}} + \dots + \left(\frac{1}{R_{NPP}} \right)^{\frac{1}{n}} = \left(\frac{1}{R_e} \right)^{\frac{1}{n}} \quad (2.11)$$

2.8 Pumping Heads

The purpose of a pump is to increase the pressure head in the system by adding energy head to the flow. The hydraulic grade line (HGL) rises across a pump as shown in Figure 2.3. The energy added by pumps is expressed in the energy equation in Eq. 2.2 as a head gain, H_p .

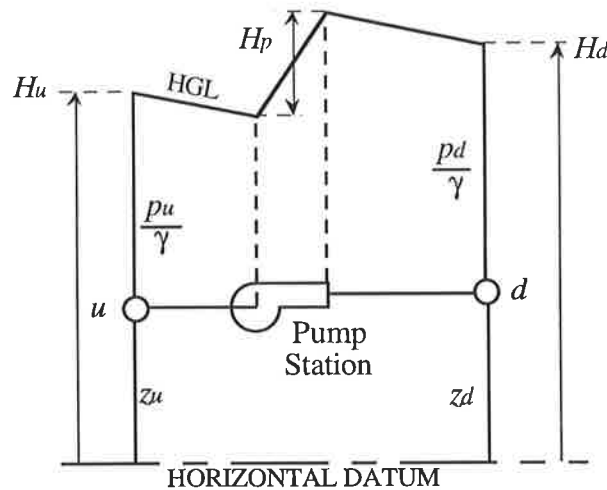


Figure 2.3 The head added by a pump

The centrifugal (or radial-flow) pump with multiple (serial or parallel) stages is the best suited and most common in water distribution systems. A pump is described by characteristic curves for pumping head, power requirements and efficiency as functions of flow.

The pump head characteristic curve relates the flow through the pump to the head produced by the pump. Some alternatives for approximating actual pump head, H_p as a function of pump flow, Q_p include a quadratic equation or an exponential equation. The quadratic equation in Eq. 2.12 will approximate the head-discharge curve within the normal working range of the pump (Jeppson, 1976).

$$H_p = A Q_p^2 + B Q_p + H_0 \quad (2.12)$$

in which H_p = pumping head, Q_p = pump flow, H_0 = pump shut-off head and A and B are constants for a given pump.

Pumping head decreases with flow for a centrifugal pump as shown in Figure 2.4. The pumping head at zero flow is the shut-off head. As the flow increases from zero flow, the pumping head may rise just above the shut-off value before falling as the flow increases further. A unique operating point is ensured if the pump head characteristic curve is strictly monotonic decreasing (the constants A and B are negative for this condition).

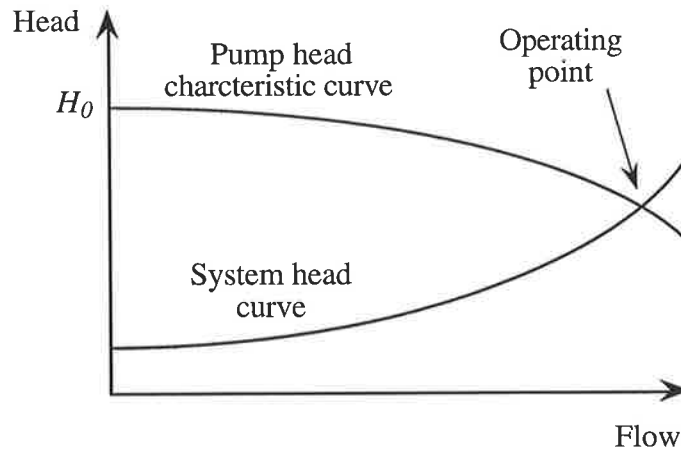


Figure 2.4 Pump head and system head curves

The efficiency characteristic curve specifies the ability of the pump to transmit power to pump head for a given flow. Head losses due to circulatory flow, torque losses and turbulence and shock losses reduce the pump efficiency. The point of best efficiency occurs when turbulence and shock losses are minimised (Streeter and Wylie, 1981). The flow at this point of peak efficiency is called the best efficiency point (BEP).

The pump power input P_p (kW) is given by Eq. 2.13 (based on Lansey and Mays, 1989c).

$$P_p = \frac{\gamma Q_p H_p}{1000 \eta_p} \quad (2.13)$$

in which η_p = pump efficiency.

A pump operates against a system head composed of the static head, friction losses in the pipes and the velocity head. Since head loss increases approximately proportional to Q^2 , as demand flows increase the system head required increases approximately quadratically. The steady-state flow through the pump (operating point) is the one at which the pump head equals the system head. This represents the intersection of the pump head characteristic curve and the system head curve as shown in Figure 2.4.

A pumping station facility may consist of multiple stages of pumps arranged in series or parallel. Given some set of pumps, the pumps operating at any time will depend on the system demands and other conditions such as water levels in balancing tanks. In this way, pumps may be operated close to their best efficiency.

2.9 The Set of Pipe Network Equations

The steady-state hydraulic analysis of a looped pipe network involves the determination of the flows in the pipes and the pressure heads at the nodes given the physical description of the network and the demand pattern to be supplied to the nodes. A set of simultaneous, nonlinear algebraic equations are formulated by applying the two fundamental physical laws of the conservation of mass to the nodes and the conservation of energy to the loops. These laws are analogous to Kirchoff's first and second laws in electrical network theory. The pipe network equations are usually formulated in one of three ways:

- the pipe flow equations (Q -equations)
- the node equations (H -equations)
- the loop corrective flow equations (ΔQ_L -equations)

The set of pipe flow equations (Q -equations) express the mass continuity equations at the nodes and the energy conservation equations around the loops in terms of the unknown flows in the pipes. The set of node equations (H -equations) express the mass continuity equations in terms of the unknown heads at the junction nodes. The set of loop equations (ΔQ_L -equations) express the energy conservation equations in terms of the unknown flow corrections around the loops (flow corrections to an assumed initial pipe flow pattern which satisfies continuity).

Since the relationship between flow in a pipe and friction head (energy) loss in the pipe is nonlinear (Section 2.6), the system of pipe network equations for looped networks is nonlinear (regardless of formulation). The equations cannot be solved directly and must be solved using an iterative method. Three widely used methods for solving the system of nonlinear network equations are the following iterative numerical techniques:

- (1) Linear theory method
- (2) Newton-Raphson method
- (3) Hardy Cross method (pre-computer method)

The linear theory method developed by Wood and Charles (1972) may be applied to the pipe flow equations or the node equations. The method is described in detail in Section 2.9.1.

The Newton-Raphson method is widely used for solving systems of nonlinear equations. Consider a system of N functions f_i in the variables x to be set equal to zero.

$$f_i(x) = 0 \quad \text{for } i=1, \dots, N \quad (2.14)$$

2 Development of a hydraulic simulation model

in which the vector of unknowns is:

$$\mathbf{x} = (x_1, x_2, \dots, x_N) \quad (2.15)$$

If $\mathbf{x}^{(0)}$ is an approximate solution to Eq. 2.14, then $\mathbf{x}^{(0)} + \delta\mathbf{x}$ is an improved approximation. Thus:

$$\delta\mathbf{x} = (\delta x_1, \delta x_2, \dots, \delta x_N) \quad (2.16)$$

Expanding the functions f_i in Taylor series and neglecting the higher order terms produces a set of N linear equations in terms of the corrections $\delta\mathbf{x}$ that move each function closer to zero simultaneously.

$$\frac{\partial f_i}{\partial x_1} \delta x_1 + \frac{\partial f_i}{\partial x_2} \delta x_2 + \dots + \frac{\partial f_i}{\partial x_N} \delta x_N = -f_i \quad \text{for } i=1, \dots, N \quad (2.17)$$

The linear system of equations in Eq. 2.17 may be expressed using matrices by Eq. 2.18. The N by N matrix of partial derivatives is known as the Jacobian matrix of coefficients. The functions and the partial derivatives in Eq. 2.18 are evaluated at the current approximation of \mathbf{x} .

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \dots & \frac{\partial f_N}{\partial x_N} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_N \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \\ \vdots \\ -f_N \end{bmatrix} \quad (2.18)$$

The corrections $\delta\mathbf{x}$ can be determined by solving Eq. 2.17 by applying standard matrix procedures for the solution of systems of linear equations to the Jacobian matrix in Eq. 2.18. The improved values of \mathbf{x} are computed using Eq. 2.19 for the $(m+1)^{\text{th}}$ iteration.

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \delta\mathbf{x} \quad (2.19)$$

The set of linear equations in Eq. 2.18 are set up for each iteration by evaluating the functions and partial derivatives at the improved approximations of \mathbf{x} to determine the corrections $\delta\mathbf{x}$ to the improved approximation. The iterative procedure is repeated until convergence of the functions or the variables is exhibited. The Newton-Raphson method may be used to solve the pipe flow equations, the node equations or the loop equations.

Martin and Peters (1963) introduced the Newton-Raphson method (or Newton's method) for the simultaneous solution of the nonlinear algebraic hydraulic network equations. The

2 Development of a hydraulic simulation model

application of the Newton-Raphson method for efficiently simulating pipe networks was made possible by the development of the computer. Prior to the development of computers, the Hardy Cross method (suitable for hand computation) of systematic relaxation was the standard numerical procedure for solving the pipe network equations.

The Hardy Cross method is the oldest method for balancing pipe networks. It was devised by Hardy Cross in 1936. The Hardy Cross method uses Newton's method to solve one nonlinear equation at a time in one unknown (in contrast to the Newton-Raphson method that solves the set of nonlinear equations simultaneously). The Hardy Cross method may be applied to the node equations, but is more commonly applied to the loop equations. The Hardy Cross method has often been reported to converge slowly (if convergence does occur) and a number of variations of the method have been suggested to improve convergence. In a general application of the Hardy Cross method to the loop equations (Jeppson, 1976):

- (1) an initial set of pipe flows is determined which satisfies continuity at the junction nodes,
- (2) Newton's method is used to solve one loop equation at a time for the unknown flow correction around the loop assuming the remaining flow corrections are temporarily known,
- (3) usually the flow corrections are computed once for each loop (or path) in the system and the process is repeated using the improved solution until convergence is exhibited.

2.9.1 Pipe flow equations (Q -equations)

The set of pipe flow equations are formed with the flows in each pipe as the unknowns. A mass continuity balance equation must be satisfied for each junction node which states the volumetric flow rate into the node must equal the volumetric flow rate out of the node. There are NJ independent linear mass continuity equations in the form of Eq. 2.20.

$$\sum_{k=1}^{NPJ} Q_k = Q_{exi} \quad \text{for junction nodes } i=1, \dots, NJ \quad (2.20)$$

in which NPJ = total number of pipes k connected to junction node i , Q_k = flow in pipe k connected to node i (flow towards node i is positive) and Q_{exi} = demand at node i .

An energy conservation equation can be written for each natural loop and pseudo loop which states the net energy gain around a loop is zero. There are $NL+(NF-1)$ independent nonlinear energy conservation equations in the form of Eq. 2.21.

$$\sum_{k=1}^{NPL} [h_{f_k} \pm H_{P_k}] - \Delta E_{mn} = 0 \quad \text{for loops } l=1, \dots, NL+(NF-1) \quad (2.21)$$

2 Development of a hydraulic simulation model

Therefore,

$$\sum_{k=1}^{NPL} [R_k |Q_k| Q_k^{n-1} \pm (A_P |Q_k|^2 + B_P |Q_k| + H_{0P})] - \Delta E_{mn} = 0$$

for loops $l=1, \dots, NL+(NF-1)$ (2.22)

in which

NPL = total number of pipes k that make up loop l

h_{fk} = the pipe friction head loss in pipe k

H_{P_k} = the pump head of pump P in pipe k ($H_{P_k}=0$ if pipe k does not contain a pump)

Q_k = flow in pipe k in an assumed direction (clockwise or anti-clockwise)

R_k = loss coefficient of pipe k

A_P, B_P and H_{0P} are the pump curve coefficients for pump P

$\Delta E_{mn} = E_m - E_n$ = elevation difference between fixed-grade nodes m and n in the case of a pseudo loop ($\Delta E_{mn}=0$ for a natural loop) where m is the node connected to pipe $k=1$ and n is the node connected to pipe $k=NPL$

The pumping head, H_{P_k} is considered when a pump exists in the pipe k within the loop l . The sign of the pumping head will depend on the direction of pumping. The flow through the pump, Q_k must be positive in the direction of pumping. There are NJ mass continuity equations and $NL+(NF-1)$ energy conservation equations forming the set of NP simultaneous equations in terms of the NP unknown pipe flows (Eq. 2.1).

Wood and Charles (1972) proposed a method for a hydraulic network analysis using the linear theory method based on the simultaneous solution of the pipe flow equations. The method proposed to transform the nonlinear energy equations represented by Eq. 2.21 into linear equations by approximating the head loss as shown in Eq. 2.23.

$$h_{fk} = R_k Q_k^n = R_k |Q_{k0}|^{n-1} Q_k = R_k' Q_k \quad (2.23)$$

in which Q_{k0} = a first guess of the flow in pipe k .

An initial flow is assumed in each pipe. The $NL+(NF-1)$ linearised energy equations are combined with the NJ continuity equations to yield NP simultaneous linear equations in terms of flows in each pipe which can be solved using standard matrix procedures. At each iteration, the computed values of flow are used as the new estimates of the flows and the process is repeated. The iterations continue until convergence is exhibited.

2 Development of a hydraulic simulation model

Wood and Charles (1972) propose a first approximation of $Q_{k0}=1.0$ (i.e., $R_k'=R_k$) provides a reasonable initial guess for the flow distribution as the solution approximates a laminar flow distribution (since the head loss is assumed to vary linearly with flow). They suggest this initial guess will give feasible estimates of the turbulent flow distribution. To protect against oscillations about the final solution, Wood and Charles suggest the new approximate flow, $Q_k^{(m)}$ in pipe k for the m^{th} iteration be calculated using Eq. 2.24. This averaging process showed faster convergence to the solution.

$$Q_k^{(m)} = \frac{Q_k^{(m-1)} + Q_k^{(m-2)}}{2} \quad (2.24)$$

Wood and Charles analysed a network with 19 pipes, 12 nodes and 8 natural loops using their proposed linear theory method, a Hardy Cross method and a Newton-Raphson method. The linear theory method provided excellent convergence, requiring 3 iterations to achieve a practical level of accuracy. The Newton-Raphson method converged in 3 iterations, provided the initial guesses for flow were good. The Newton-Raphson method required about 2 additional iterations, if the initial guesses for flow were poor. The Hardy Cross method required about 20 iterations to approach a similar accuracy. Wood and Charles concluded that the linear theory method shows rapid and accurate convergence. The approach does not require initial estimates of flows.

Jeppson (1976) found the linear theory method did not give rapid convergence when analysing networks with pumps using the pipe flow equations with energy equations written in the form of Eq. 2.21. Jeppson did however describe modifications to the pipe flow equations which improved the convergence. Jeppson did not recommend the use of the linear theory method for solving the node equations or the loop corrective flow equations.

Wood and Rayes (1981) examined the reliability of five widely used methods that have been proposed to perform a steady-state analysis of pressure and flow in a pipe network including:

- (1) the linear method applied to the pipe flow equations
- (2) the Hardy Cross method applied to the node equations
- (3) the Newton-Raphson technique applied to the node equations
- (4) the Hardy Cross method applied to the loop equations
- (5) the Newton-Raphson technique applied to the loop equations

2 Development of a hydraulic simulation model

Wood and Rayes (1981) applied the five methods to a large number of real pipe networks (60 networks with less than 100 pipes and 31 networks with more than 100 pipes). A convergence criterion was specified which stated the average change in flows between successive iterations must be less than 0.5%. The iterations were continued until the convergence criterion was met, however, convergence does not guarantee an accurate solution. The accuracy of the methods based on the pipe flow equations or loop equations was indicated by the unbalanced heads for the energy equations. The accuracy of the methods based on the node equations was indicated by the unbalance in continuity at junction nodes. Liberal upper limits were placed on the maximum number of iterations allowed (from 20 iterations for the linear method up to 200 iterations for the Hardy Cross methods).

Wood and Rayes (1981) recorded a 'failure' if convergence was observed and the solution was inaccurate (average deviation of flow or head exceeded 10% or the maximum deviation of flow or head exceeded 30%). A 'failure' was also recorded if convergence was not observed within the maximum number of iterations.

The method referred to as the *linear method* by Wood and Rayes (1981) and as the *modified linear method* by Ormsbee and Wood (1986) is not the original *linear theory method* described by Wood and Charles (1972) as reported by Salgado et al. (1988) and Ellis and Simpson (1996). The *linear method* (Wood and Rayes, 1981), which is the solution technique adopted in the KYPIPE hydraulic simulation computer program (Wood, 1974), is actually quite similar to the Newton-Raphson method (Ellis and Simpson, 1996).

Wood and Rayes investigated the reliability of the linear method for the solution of the pipe flow equations. Excellent convergence characteristics were shown and no failures were reported using the method. The linear method was used to determine the exact solutions for all the comparisons, by performing one further iteration after the specified convergence criterion was reached, which is an indication of the reliability of this method. For the 31 networks with more than 100 pipes (up to 509 pipes), the linear method computed solutions with good accuracy in an average 6.4 iterations (which did not depend on the size of the system) and using an average 5.4 seconds of computer time. The reliability study conducted by Wood and Rayes (1981) was revisited by Wood and Funk (1993).

2.9.2 Node equations (*H*-equations)

The set of node equations are formed with the heads at the junction nodes in the pipe network as the unknowns. The friction head loss in a pipe connecting nodes *i* and *j* represents the head difference between nodes *i* and *j*:

$$h_{fij} = H_i - H_j \quad (2.25)$$

in which h_{fij} = the friction head loss in the pipe connecting nodes *i* and *j*, H_i = HGL or total head (pressure head+elevation head) at node *i* and H_j = HGL or total head at node *j*.

Rearranging the head loss relationship given by Eq. 2.3 and substituting Eq. 2.25 gives:

$$Q_{ij} = \left(\frac{H_i - H_j}{R_{ij}} \right) \left| \frac{H_i - H_j}{R_{ij}} \right|^{\frac{1}{n} - 1} \quad (2.26)$$

The node equations are based on the continuity relationship in Eq. 2.20. Substituting Eq. 2.26 into Eq. 2.20 gives:

$$\sum_{j=1}^{NPJ} \left(\frac{H_i - H_j}{R_{ij}} \right) \left| \frac{H_i - H_j}{R_{ij}} \right|^{\frac{1}{n} - 1} = Q_{exi} \quad \text{for junction nodes } i=1, \dots, NJ \quad (2.27)$$

in which NPJ = total number of pipes connecting nodes *j* to node *i*.

There are a total of NJ nonlinear simultaneous node equations in terms of the NJ unknown heads at the junction nodes. The node equations in Eq. 2.27 are formed assuming the pipes do not contain pumps. Wood and Rayes (1981) described a method to expand the node equations to include pumps by introducing two junction nodes at the suction and discharge sides of the pump. Two additional equations are written in terms of the two additional unknown heads at the new junction nodes. A flow continuity equation is formed expressing flow using Eq. 2.26 since the flow in the suction line (upstream) is equal to the flow in the discharge line (downstream). Another equation relates the head gain across the pump to the flow in either the suction line or the discharge line.

Martin and Peters (1963) reported oscillations of the corrections to the head approximations when the Newton-Raphson method was applied to the node equations. They identified possible reasons for the oscillations and overcame the convergence difficulties by halving the corrections when oscillations were observed.

2 Development of a hydraulic simulation model

Shamir and Howard (1968) proposed a generalised method for determining the steady-state solution of a water distribution network by solving the node equations using the Newton-Raphson method. Shamir and Howard allowed the NJ unknowns to be computed to be some solvable combination of the node heads H , the demands, Q_{ex} and the pipe loss coefficients (or resistance), R . A combination of unknowns which produces a Jacobian matrix with a row(s) which contains only zeros is not solvable.

The method proposed by Shamir and Howard (1968) may be used to determine pipe resistances for some specified system performance. Unfortunately, the optimum combination of node heads which produces the lowest cost combination of network elements is difficult to predict. Shamir and Howard suggest any network element which may be represented by a function relating flow and head changes such as pipes, pumps and valves may be included in the network analysis, however, convergence is not guaranteed when a characteristic function of an element does not have a continuous derivative. Shamir and Howard reported good convergence to a solution using their method provided reasonable initial guesses are made for the set of unknowns.

Gessler's (1982) enumeration algorithms required the repeated hydraulic analyses of network pipe size combinations in the search for the minimum cost pipe network. Gessler believed methods based on the node equations where the pressure heads at the nodes are the unknowns are superior to methods based on the loop equations especially for the purposes of optimisation where the critical constraint is the pressure head distribution. The iterations of a method based on the node equations can be terminated as soon as the pressure heads converge to some specified accuracy which may avoid unnecessary iterations. In addition, Gessler suggested the average absolute head adjustment is a more reliable measure of the pressure head accuracy than the average absolute residual of the head losses around the loop. Gessler found the successive head adjustments consistently reduced by a factor of about 0.5. Under some conditions, over-relaxation by a factor of about 1.85 produced the best results.

Gessler suggested that for large networks where computer time for a solution is important, there are almost as many loops ($NL+(NF-1)$) as nodes (NJ) for realistic layouts. This claim is true for extensively interconnected layouts, however in general, the number of node equations is greater than the number of loop equations. Gessler argued the node equations are especially suited to analysing systems with pumps, pressure reducing valves and check valves. The pipe flow equations and loop equations require the formation of the loop structure compared to the node equations which do not require this information. Gessler presented several reasons why the node equations may be favoured, however, Wood and Rayes (1981) questioned the reliability of methods based on the node equations.

2 Development of a hydraulic simulation model

Wood and Rayes (1981), in their study of the reliability of five commonly used methods of pipe network analysis (Section 2.9.1), examined the Newton-Raphson method (Shamir and Howard, 1968) and the Hardy Cross method for the solution of the node equations. Both methods require an initial realistic assumption of the node heads. The Hardy Cross method computes a head adjustment factor for each junction node consecutively. The adjusted heads approach the solution of Eq. 2.27 at the node assuming the heads at adjacent nodes are fixed. An iteration is complete when the heads at all the junction nodes in the pipe network have been adjusted. By comparison, the Newton-Raphson technique applied to the node equations adjusts the heads at the junction nodes simultaneously.

Significant convergence problems were exhibited by both the Hardy Cross and Newton-Raphson methods applied to the node equations. The Hardy Cross method applied to the node equations proved to be the least reliable of the five methods examined by Wood and Rayes. For the 60 networks analysed with less than 100 pipes, 51 failures (a 'failure' was defined in Section 2.9.1) were recorded using the method. In most cases, the convergence criterion of 0.5% was satisfied, but the solution was found to be in error. The convergence criterion was not satisfied for 7 networks, yet a maximum of 200 iterations were allowed. Four of 5 networks tested further, failed to converge after 1000 iterations and the solution which did converge was found to be in error. Often substantial errors were found in solutions when the convergence criterion was attained. The probability of an accurate solution was increased by applying a more stringent convergence criterion, however, this did not promise an accurate solution. Given a convergence criterion of 0.05% and a maximum of 400 iterations, 15 of 15 networks tested converged, however, 2 failures were noted.

The Newton-Raphson technique applied to the node equations is capable of providing a highly accurate solution in relatively few iterations, however, Wood and Rayes (1981) recorded a number of problems for this method. A total of 18 failures were reported using the method, from the 60 networks with less than 100 pipes analysed. Convergence was not observed for 8 networks within the allowed 40 iterations. In some cases, oscillations occurred which prevented the attainment of the convergence criterion regardless of the number of iterations. Five of 9 networks tested further, failed to converge after 400 iterations and 6 failures were noted.

Wood and Rayes (1981) found the reliability of the node methods was hindered in some cases by the inability to handle pipes which carry very high flows at very low head losses. In these situations, small errors in head calculations may produce serious errors in flow calculations. Shamir and Howard (1968) previously had reported that the Hardy Cross method may converge very slowly, or even diverge, given some conditions such as pipes with low resistance or very low flows (i.e., very low head losses). The node methods require an initial

set of heads and the chance of obtaining a good solution are increased by using initial values which are closer to the correct values, however, convergence is not assured and it is difficult to consistently estimate close initial values. Nielsen (1989) also discouraged the use of the node equations reasoning it is difficult to determine a good initial set of node heads and convergence to the solution may be slow.

2.9.3 Loop corrective flow equations (ΔQ_L -equations)

The set of loop equations are formed with the flow corrections for each of the loops as the unknowns. Eq. 2.28 is derived from the nonlinear energy equation in Eq. 2.21. Eq. 2.28 is a set of nonlinear equations in the unknown flow corrections q for each loop.

$$F_l(q) = 0 \quad \text{for loops } l=1, \dots, NL+(NF-1) \quad (2.28)$$

$$\sum_{k=1}^{NPL} \left[R_k(Q_{0k} \pm q) |Q_{0k} \pm q|^{n-1} \pm (A_P |Q_{0k} \pm q|^2 + B_P |Q_{0k} \pm q| + H_{0p}) \right] - \Delta E_{mn} = 0$$

for loops $l=1, \dots, NL+(NF-1)$ (2.29)

in which

F_l = net head loss around the loop l

$q = (q_1, q_2, \dots, q_{NL+(NF-1)})$ = one flow correction for each loop

NPL = total number of pipes k that make up loop l

R_k = loss coefficient of pipe k

Q_{0k} = initial flow in pipe k

A_P, B_P and H_{0p} are the pump curve coefficients for pump P (if pipe k contains a pump)

$\Delta E_{mn} = E_m - E_n$ = elevation difference between fixed-grade nodes m and n in the case of a pseudo loop ($\Delta E_{mn}=0$ for a natural loop) where m is the node connected to pipe $k=1$ and n is the node connected to pipe $k=NPL$

The flow correction q_l is added (or subtracted depending on the sign convention adopted) to each assumed initial flow, Q_{0k} when the pipe k forms part of loop l . If pipe k is common to several loops, the flow, Q_{0k} is adjusted by the corrections for each loop.

An initial flow pattern must be specified such that continuity at each of the junction nodes is satisfied. The initial flow pattern generally will not satisfy the energy equations and therefore the flow corrections are determined such that each loop energy equation is satisfied. The final adjusted flow pattern will continue to satisfy the continuity equations. There are a total of $NL+(NF-1)$ nonlinear simultaneous equations in terms of the unknown flow corrections in each

2 Development of a hydraulic simulation model

of the loops (including natural loops and pseudo loops). The number of loop equations is generally less than the number of node equations which in turn is less than the number of pipe flow equations, since the number of loops ($NL+(NF-1)$) is generally less than the number of nodes (NJ) which in turn is less than the number of pipes (NP) in a looped pipe network.

Epp and Fowler (1970) present the outline of a computer program for the steady-state analysis of pipe networks. The algorithm features the Newton-Raphson method to solve the loop corrective flow equations. The Newton-Raphson method applied to the loop equations gives fast (quadratic) convergence.

The algorithm proposed by Epp and Fowler (1970) was designed to minimise computer time and storage requirements. The algorithm utilises a 'minimal spanning tree' to assign the initial flow pattern to the pipe network. This provides an initial flow pattern which satisfies continuity of flow at the nodes and Epp and Fowler believed that it assures convergence. The algorithm utilises a 'minimum path algorithm' to define the natural loops and pseudo loops in the pipe network given node connectivity information. The number of nonlinear equations to be solved equals the number of loops. The loops are numbered so as to obtain a symmetric and banded Jacobian matrix of coefficients of near minimum bandwidth. This type of matrix minimises storage requirements and is easily solved. The algorithm developed by Epp and Fowler was used successfully on a wide variety of networks. The method was applied to some networks where convergence by the Hardy Cross method or the Newton-Raphson method for the node equations was not exhibited, the largest network consisting of 307 pipes, 170 nodes, 135 natural loops and 4 pseudo loops.

Wood and Rayes (1981) in their investigation into the reliability, accuracy and efficiency of methods of hydraulic analysis (formulation of pipe network equations and numerical solution methods) considered the linear method applied to the pipe flow equations (Section 2.9.1), the Newton-Raphson technique and Hardy Cross method applied to the node equations (Section 2.9.2) and also the Newton-Raphson technique and the Hardy Cross method for the solution of the loop equations.

Wood and Rayes (1981) found that significant accuracy problems were exhibited by the Hardy Cross method applied to the loop equations. Given a maximum 200 iterations, the method converged in every case for the 60 networks with less than 100 pipes. Although the convergence criterion was reached, substantial errors were found in some solutions. A total of 8 failures were documented for the 60 networks using the convergence criterion of 0.5%. The probability of an accurate solution was increased by specifying a more stringent convergence criterion, however, this did not guarantee a good solution. Five of the 8 previous failures were improved by applying a convergence criterion of 0.05% and allowing a maximum of 400

2 Development of a hydraulic simulation model

iterations. The 3 remaining failures featured significant unbalanced heads and errors in flows to and from storage nodes. For most of the failures, the average unbalanced head was relatively low. The Hardy Cross method encountered difficulty with a very high head loss and a very low head loss in the same energy equation. The method requires an initial set of flows that satisfy continuity at the nodes and the chances of obtaining a good solution are increased by using initial flows which are closer to the balanced flows, however, convergence is not assured.

Wood and Rayes (1981) also examined the Newton-Raphson method applied to the loop equations. The method was developed to improve the convergence characteristics of the Hardy Cross method. The Newton-Raphson method also requires as input an initial set of flows which satisfy continuity at the nodes. A flow adjustment factor is computed for each natural loop and pseudo loop simultaneously. A set of $NL+(NF-1)$ simultaneous linear equations are constructed in terms of the loop flow adjustment factors. The set of linear equations can be solved using standard matrix procedures. The flow adjustment factors tend to balance the energy equations for a loop by considering the energy relationship for the particular loop and also for all other loops with pipes in common with this loop. The computation of a new set of loop flow adjustments to the initially assumed pipe flow distribution constitutes an iteration. Iterations continue until convergence is exhibited. The loop flow adjustments correct the assumed flows in all pipes in each loop and so continuity is maintained.

Wood and Rayes found the performance of the Newton-Raphson method applied to the loop equations to be comparable to that of the linear method (for the pipe flow equations) computing very accurate solutions in few iterations and using reasonable computer times. For 31 networks with more than 100 pipes, the Newton-Raphson method computed accurate solutions in an average 8.5 iterations and required an average 6.9 seconds of computer time (by comparison, the linear method computed accurate solutions in an average 6.4 iterations and required an average 5.4 seconds of computer time). Wood and Rayes found the Newton-Raphson method for the loop equations showed excellent convergence characteristics. For 60 networks with less than 100 pipes analysed, the method failed once to converge within 30 iterations. The single failure is attributed to a pipe network containing a constant power pump characterised by a steep head-discharge curve. The low horsepower pump operated at low flow rates. The steepness of the characteristic curve led to convergence problems for all the methods investigated by Wood and Rayes except the linear method.

Nielsen (1989) formulated the hydraulic equations in matrix notation and observed the behaviour of the linear theory method and the Newton-Raphson method. Nielsen found the iterates for the successive solution of the system of pipe flow equations are identical to the iterates for the successive solution of the system of loop equations. The system of loop

2 Development of a hydraulic simulation model

equations requires a starting set of flows that satisfies continuity. The corresponding iterates are identical regardless of the choice of loops. Nielsen used linear algebra to show that the linear theory method is a variation of the Newton-Raphson method. He suggests using the linear theory method to determine a starting point (the first iteration) for a Newton-Raphson method (used for second and subsequent iterations) and demonstrated the accuracy and reliability of this modification. Nielsen suggests the use of the loop equations since there are a smaller number of unknowns ($NL+(NF-1)$) and convergence is efficient.

2.10 The Adopted Method of Hydraulic Analysis

The hydraulic simulation model developed in this study to be linked with the genetic algorithm optimisation model uses the Newton-Raphson numerical solution technique applied to the loop corrective flow equations (Epp and Fowler, 1970) as outlined in Section 2.9.3. The reliability of the method is demonstrated in the valuable work of Wood and Rayes (1981). The method was extensively tested and consistently exhibited excellent convergence to accurate solutions.

Since it is expected that thousands of network analyses need to be performed by the genetic algorithm search to optimise a proposed pipe network design, the efficiency of the adopted method of hydraulic analysis is critical. Few iterations are required by the Newton-Raphson method to reach accurate solutions owing to the quadratic convergence characteristics. The number of simultaneous equations to be solved using the loop equations are usually fewer (than the node equations and pipe flow equations) since in general there are less loops than pipes or nodes in a pipe network. The extensively interconnected *Anytown* network in Figure 2.5 (Walski et al., 1987) is an exception. Jeppson (1976) found the Newton-Raphson method required less computer storage for a given number of equations.

The adopted method of hydraulic analysis solves for the flow corrections around the loops. The loop corrective flow equations or ΔQ_L equations (Eq. 2.29) are derived from the energy conservation equations around the loops (Eq. 2.22). The method requires initial flows in the pipes to be specified which satisfy continuity at the nodes given some set of consumer water demands at the nodes. The method requires the pipes in the loops and the loop structure to be identified. A computer program to perform a hydraulic analysis may incorporate algorithms which undertake these tasks in order to save the users additional effort and minimise errors in data input. In the development of the optimisation model in this research, the motivation for the development of algorithms to compute a set of initial flows and to identify the loops arose as the proposed network layout and operation may be considered as design variables.

The adopted method of hydraulic analysis is described in the following sections and particularly with respect to the hypothetical *Anytown* water distribution system.

2.10.1 The *Anytown* water distribution system

Walski et al. (1987) presented the water distribution system for the hypothetical community of *Anytown* in the 'Battle of the Network Models' study. The existing water distribution system before the proposed system expansions is represented by the pipe network in Figure 2.5.

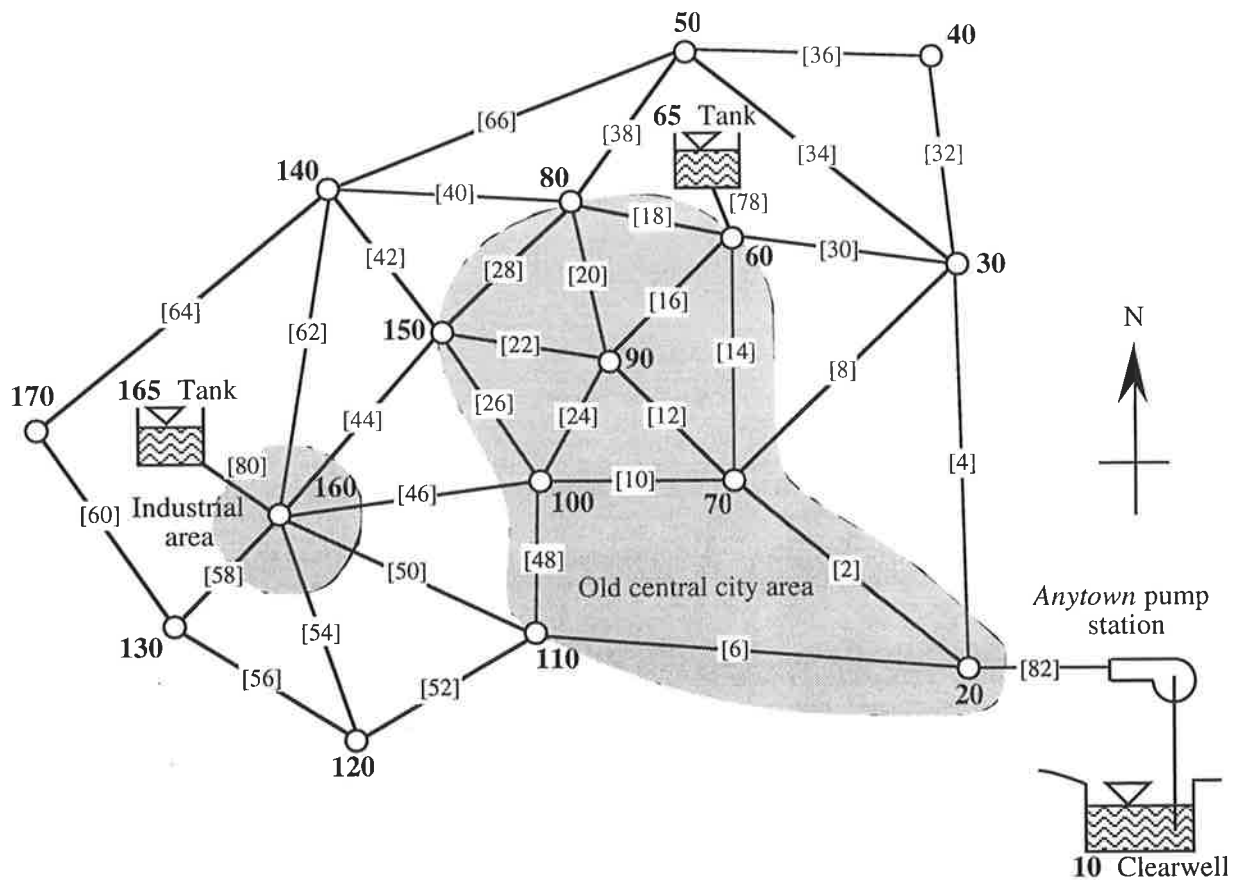


Figure 2.5 The Anytown water distribution system

The lengths, diameters and roughness coefficients of the pipes are given in Table 2.1. The lengths and diameters of the pipes refer to the existing network, however, the roughness coefficients are values projected for 20 years time. Pipes [78] and [80] are riser pipes to elevated tanks and pipe [82] is the pumping main. The loss coefficients for the pipes (Eq. 2.4) in the *Anytown* system are given in Table 2.1. The Hazen-Williams friction head loss formula (Eq. 2.3) and US customary units are used for this study.

The current average daily demand flows for the junction nodes of the *Anytown* system are given in Table 2.2. The nodes within the old central city area and the industrial area around node 160 are associated with higher water requirements. The demands at the junction nodes should be met while maintaining minimum pressures of 40psi.

The fixed-grade nodes are numbered 10, 65 and 165 in the *Anytown* system. Node 10 is a clearwell at the water treatment plant where water from a nearby river is treated. The elevation of the water level at the clearwell is maintained at 10 ft. Nodes 65 and 165 are elevated storage tanks both with an elevation of 215 ft and a minimum water level of 10 ft and both with a volume of 250,000 gal.

Table 2.1 The pipes of the Anytown network

[Pipe]	Start node	End node	Length, L (ft)	Diameter, D (in)	Hazen-Williams roughness, C	Loss coefficient (Resistance, R)
[2]	20	70	12000	16	70	5.348
[4]	20	30	12000	12	120	8.002
[6]	20	110	12000	12	70	21.713
[8]	70	30	9000	12	70	16.285
[10]	70	100	6000	12	70	10.857
[12]	70	90	6000	10	70	26.384
[14]	70	60	6000	12	70	10.857
[16]	90	60	6000	10	70	26.384
[18]	60	80	6000	12	70	10.857
[20]	90	80	6000	10	70	26.384
[22]	90	150	6000	10	70	26.384
[24]	90	100	6000	10	70	26.384
[26]	100	150	6000	12	70	10.857
[28]	150	80	6000	10	70	26.384
[30]	60	30	6000	10	120	9.723
[32]	30	40	6000	10	120	9.723
[34]	30	50	9000	10	120	14.585
[36]	40	50	6000	10	120	9.723
[38]	80	50	6000	10	120	9.723
[40]	80	140	6000	10	120	9.723
[42]	150	140	6000	8	120	28.828
[44]	160	150	6000	8	120	28.828
[46]	100	160	6000	8	120	28.828
[48]	110	100	6000	8	70	78.223
[50]	110	160	6000	10	120	9.723
[52]	110	120	6000	8	120	28.828
[54]	160	120	9000	2	120	37000.000
[56]	120	130	6000	8	120	28.828
[58]	160	130	6000	10	120	9.723
[60]	130	170	6000	8	120	28.828
[62]	160	140	6000	8	120	28.828
[64]	170	140	12000	8	120	57.656
[66]	140	50	12000	8	120	57.656
[78]	65	60	100	12	120	0.0667
[80]	165	160	100	12	120	0.0667
[82]	10	20	100	30	140	0.00058

The Anytown source pump station pumps water to the Anytown system from the water treatment plant at the clearwell (node 10). The pump station consists of three identical pumps arranged in parallel. The pump characteristics for the individual pumps are given in Table 2.3. The rated flow (point of peak efficiency) for each pump is 4000 gpm with 65% efficiency. The efficiencies of the pumps are wire-to-water efficiencies which includes motor and pump efficiencies.

Table 2.2 The nodes of the Anytown network

Node	Degree (number of connected pipes)	Elevation (ft)	Demand (cfs)	Minimum allowable pressure head (psi)
10	1	10	Clearwell	
65	1	225	Elevated storage tank	
165	1	225	Elevated storage tank	
20	4	20	1.114	40
30	5	50	0.446	40
40	2	50	0.446	40
50	4	50	0.446	40
60	5	50	1.114	40
70	5	50	1.114	40
80	5	50	1.114	40
90	5	50	2.228	40
100	5	50	1.114	40
110	4	50	1.114	40
120	3	120	0.446	40
130	3	120	0.446	40
140	5	80	0.446	40
150	5	120	0.446	40
160	7	120	1.782	40
170	2	120	0.446	40

Table 2.3 Anytown pump characteristics

Pump flow, Q_p (gpm)	Pump head, H_p (ft)	Pump efficiency, η_p (%)	Pump power, P_p (horsepower)
0	300	0	-
2000	292	50	295.2
4000	270	65	419.9
6000	230	55	634.2
8000	181	40	914.9

The pump head characteristic curve for a pump in the *Anytown* pump station is approximated by Eq. 2.30.

$$H_p = -0.372 Q_p^2 - 0.056 Q_p + 300 \quad (2.30)$$

The set of equivalent characteristic curves for r identical operating parallel pumps in the *Anytown* pump station are approximated by Eq. 2.31.

$$H_p = -0.372 \left(\frac{Q_p}{r}\right)^2 - 0.056 \left(\frac{Q_p}{r}\right) + 300 \quad (2.31)$$

2.10.2 Determination of natural loops and pseudo loops

An algorithm has been developed in this research which defines the pipes in the natural loops and pseudo loops. The algorithm is outlined below.

In the following discussion, a Breadth First Search (Even, 1979) or BFS is a method of traversing a network by fanning out from some start node s . The start node s corresponds to a level 0. Every node adjacent to the start node is visited and these nodes are located on level 1. The level of a node corresponds to the minimum number of pipes connecting the start node and the node in question. The nodes which are connected to the nodes on level 1 and which have not already been visited are visited and located on level 2. The sweep of the network in levels of nodes continues until all the nodes have been visited or until some specified destination node d is encountered. The BFS will determine the minimum number of pipes in the path between start node s and destination node d .

The Natural Loops

The structure of the NL natural loops may be deduced from the node connectivity information using the following procedure.

Step 1. *Determine the degree of the nodes.*

The 'degree' of a node in the looped network equals the number of pipes (excluding imaginary pipes used to form pseudo loops) connecting the node to adjacent nodes.

Step 2. *Disconnect 'tails' from the looped network.*

A node with degree=1 and the pipe connecting the node to the looped network may be excluded from further consideration by the algorithm since the pipe is a 'tail' which is not an edge of a natural loop.

Step 3. *Modify the degree of the remaining nodes.*

The degree of nodes adjacent to disconnected nodes are modified (reduced by one for each adjacent node that is disconnected).

Step 4. *Repeat Steps 2 and 3 until there are no nodes of degree=1 remaining.*

Step 5. *Establish a 'key' node and a 'start' node.*

A key node is a node in the pruned network of least degree, preferably of degree=2, but occasionally only nodes of degree ≥ 3 will remain (in rare situations, using a key node with degree ≥ 3 will identify a loop which is not a natural loop). The start node is a node adjacent to the key node. A start node of least degree is selected if the key node has degree ≥ 3 .

Step 6. *Perform a Breadth First Search (BFS) extending from the start node in search of the key node.*

A BFS originates from the start node without backtracking until the key node is determined. The BFS identifies the minimum path of pipes and consequently the natural loop.

Step 7. Store the path of pipes constituting a loop.

The loop is defined as the path of pipes determined in Step 6 plus the pipe connecting the start node and key node.

Step 8. Remove the pipe connecting the start node to the key node.

Step 9. Repeat Steps 2 to 8 until the network is completely removed in Step 2.

The Pseudo Loops

The number of pseudo loops ($NF-1$) equals the number of fixed-grade nodes NF less one.

Step 1. Establish a key node.

The key node may be any fixed-grade node within the network which does not already belong to a pseudo loop.

Step 2. Perform a Breadth First Search (BFS) through the complete original looped network until another fixed-grade node is reached.

The minimum number of pipes between the fixed-grade nodes and an imaginary pipe between the fixed-grade nodes forms the pseudo loop. The pipes constituting the pseudo loop are stored. The energy loss in the imaginary pipe is equal to the difference in total head from the end fixed-grade to the start fixed-grade node.

Step 3. Repeat Steps 1 and 2 until every fixed-grade node belongs to at least one pseudo loop.

The *Anytown* pipe network contains 36 pipes ($NP=36$), 16 junction nodes ($NJ=16$) and 3 fixed-grade nodes ($NF=3$). By rearranging Eq. 2.1, there are $NL=NP-NJ-(NF-1)=18$ natural loops and there are $(NF-1)=2$ pseudo loops. The algorithms described above were used to find the 18 natural loops and 2 pseudo loops of the *Anytown* network.

The natural loops **A** to **R** (in the order they were found) are shown in Figure 2.6. First, the 'tails' are removed from the network which includes the three source nodes **10**, **65** and **165** and their respective adjoining pipes. A node of degree=2 (node **170**) is chosen as the 'key' node and an adjacent node (node **140**) is chosen as the 'start' node. A BFS extends from node **140** outwards in all directions until the key node **170** is reached. The path of pipes [64], [62], [58] and [60] form the natural loop **A**. Pipe [64] is now removed from the network. Consequently, pipe [60] represents a tail and is removed also. The nodes **130** and **160** are selected to be the new key node and start node respectively and a BFS extending from node **160** in search of node **130** identifies natural loop **B**. The process continues until the set of natural loops is defined.

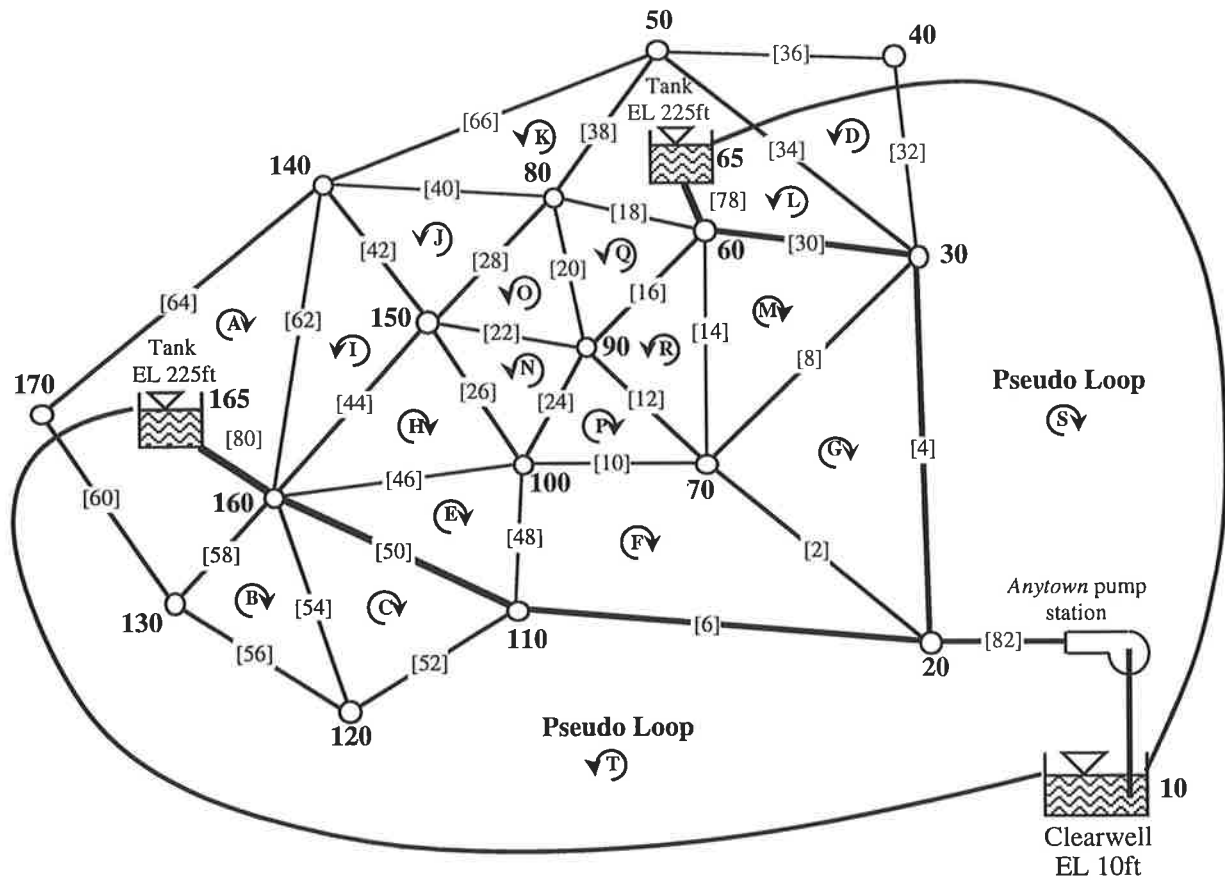


Figure 2.6 The natural loops (A-R) and pseudo loops (S-T) identified for the *Anytown* pipe network

After the natural loops A to M have been identified, the network has reduced to a bicycle-wheel network centred about node 90. Only nodes of degree ≥ 3 remain. Node 150 is arbitrarily chosen as the key node. If node 90 is chosen as the start node, then the new loop N (pipes [22], [26] and [24]) is a natural loop but the next loop O (pipes [26], [24], [20] and [28]) is not a natural loop. Similarly, if node 90 is chosen as the key node, the set of natural loops is not found. Under these circumstances, the algorithm chooses the key node of least degree and adjacent start node of least degree from the potential key nodes and start nodes. Therefore, node 150 is chosen as the key node and node 100 is chosen as the start node. The path of pipes in the natural loops may be in a clockwise or an anti-clockwise direction. The direction of the loops is shown in Figure 2.6 and the path of pipes in the loops are given in Table 2.4.

Table 2.4 The loops, initially assumed loop flow corrections and final loop flow corrections (after 6 iterations)

Loop	Path of pipes for the loops*	Initial flow corrections, $q^{(0)}$ (cfs)	Adjustments (Iteration 1) δq (cfs)	Improved flow corrections, $q^{(1)}$ (cfs)	Final flow corrections, $q^{(6)}$ (cfs)
1	64 -62 58 60	0.05	-0.157	-0.107	-0.176
2	-58 54 56	0.1	-0.711	-0.611	-0.904
3	44 42 -62	0.15	-0.164	-0.014	-0.239
4	-54 -50 52	0.2	-0.766	-0.556	-0.913
5	44 -26 46	0.25	0.092	0.342	0.654
6	-42 28 40	0.3	-0.185	0.115	-0.226
7	50 -46 -48	0.35	0.439	0.789	1.025
8 (Pseudo)	82 6 50 -80	0.4	-2.602	-2.202	-3.527
9	-26 -24 22	0.45	-0.505	-0.055	-0.939
10	-40 38 -66	0.5	-0.513	-0.013	-0.205
11	-22 20 -28	0.55	-0.211	0.339	-0.219
12	48 -10 -2 6	0.6	0.257	0.857	1.253
13 (Pseudo)	82 4 -30 -78	0.65	-1.328	-0.678	-0.608
14	-24 -12 10	0.7	-0.068	0.632	1.274
15	-38 -18 30 34	0.75	-1.305	-0.555	-0.690
16	-20 16 18	0.8	-0.095	0.705	0.056
17	2 8 -4	0.85	-0.791	0.059	0.259
18	-8 14 30	0.9	-1.270	-0.370	0.268
19	-12 14 -16	0.95	0.411	1.361	0.606
20	36 -34 32	1.0	-0.516	0.484	0.470

* A negative sign indicates the assumed pipe flow direction is opposite to the assumed loop flow direction

The 2 pseudo loops labelled **S** and **T** in Figure 2.6 were both found while performing a BFS starting at node **10**. The equivalent head loss in the imaginary pipes from node **65** to node **10** and from node **165** to node **10** is equal to 215 ft for both pseudo loops (elevation difference between water levels).

The algorithm developed in this thesis resembles the 'loop defining algorithm' proposed by Epp and Fowler (1970). Epp and Fowler suggest choosing the natural loops will generally reduce the bandwidth of the Jacobian matrix, although not necessarily improve convergence of the Newton-Raphson method.

2.10.3 Assumed initial flows

An algorithm was devised in this research which utilised node connectivity and junction node demands to determine a set of initial flows in the pipes such that continuity was satisfied. The algorithm is outlined in Steps 1 to 9 below.

Step 1. Establish a tree network originating from a principal source node.

The principal source node is usually the source node expected to supply the greatest proportion of flow to the system. A tree network is derived from the original looped network by starting at the principal source node and systematically visiting all the downstream nodes such that any two nodes are connected by just one path of pipes. The downstream nodes are visited by a Breadth First Search (BFS). Additional supply nodes (such as tanks) may not be a part of the tree network. The pipe flows in the tree (assumed initial flows in the looped network) are determined from the node demands and continuity in Steps 2 to 9.

Epp and Fowler (1970) used a tree network called a 'minimal spanning tree'. The minimal spanning tree was built by starting at any node and appending the pipe of minimum length to the tree network which connects the tree to any isolated node. Epp and Fowler claim use of the minimal spanning tree will lead to a reasonable approximation of the flow distribution, reduce the number of iterations required and assure convergence of the iterative Newton-Raphson solution technique, since pipe resistance is proportional to pipe length.

Step 2. Assign a cumulative demand to each node initially equal to the demand at this node.

Step 3. Count the degree of the junction nodes in the tree network.

The degree of a node in the tree network equals the number of pipes connecting the node to an adjacent node in the tree.

Step 4. Consider a node with degree=1.

The algorithm considers all junction nodes with degree=1 in the tree consecutively.

Step 5. The flow in the pipe connected to the node of degree=1 under consideration equals the current cumulative demand at the junction node.

Step 6. The junction node of Steps 4 and 5 and the adjacent pipe are disconnected from the tree.

Step 7. The degree of the node immediately upstream in the tree is reduced by 1.

Step 8. The cumulative flow imbalance at the node immediately upstream is adjusted by the flow in the disconnected pipe.

Step 9. Repeat Steps 4 to 8. Proceed upstream through the tree until the root source node is reached.

2 Development of a hydraulic simulation model

The flow pattern established in the tree network in the above procedure and a zero flow in the pipes that were not part of this tree represents an initial flow pattern (that satisfies continuity) for the original looped network to be simulated.

Consider the application of the procedure described above to the *Anytown* pipe network. Figure 2.7 represents the tree network generated by conducting a BFS on the *Anytown* looped pipe network originating from the clearwell at source node 10. The nodes of degree=1 in the tree network (excluding the root source node 10) are the extreme downstream junction nodes (nodes 40, 60, 80, 100, 140, 150, 160 and 170). Therefore, the initial flow in pipe [66] from node 50 to node 140 is equal to the demand at node 140 (0.446 cfs). Similarly, the initial flow in pipe [38] from node 50 to node 80 is equal to the demand at node 80 (1.114 cfs). The nodes 140 and 80 and the adjacent pipes [66] and [38] are removed from the tree and node 50 has a modified degree=1. The cumulative flow imbalance at node 50 is the sum of the demand at node 50 and the flows in pipes [66] and [38]. The initial flow in pipe [34] from node 30 to node 50 is equal to the cumulative flow imbalance at node 50 (2.005 cfs). The algorithm retreats upstream from the furthest downstream nodes until the root source node 10 is reached.

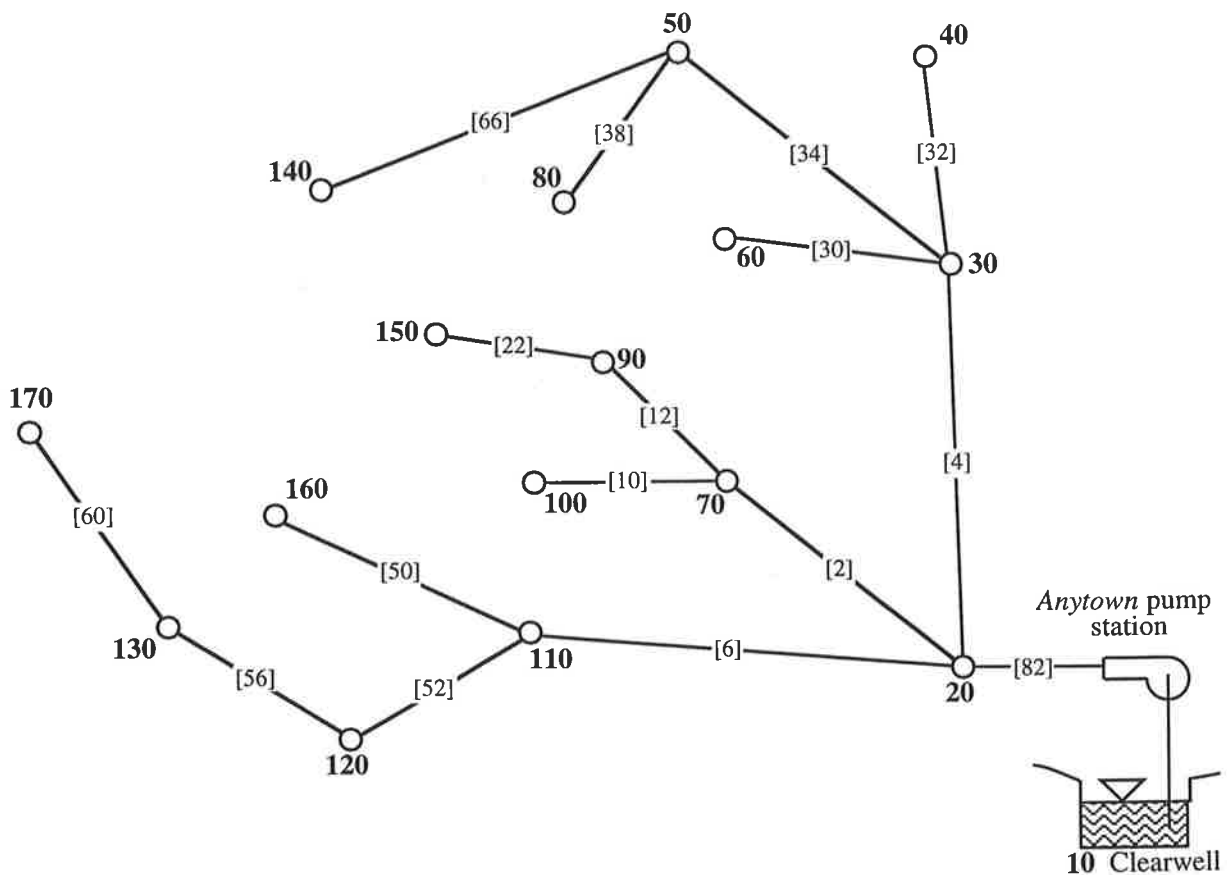


Figure 2.7 A tree network generated by a Breadth First Search (BFS) originating from the principal source node 10

2 Development of a hydraulic simulation model

The set of assumed initial flows for the *Anytown* system are given in Table 2.5. It may be anticipated that a tree network which fans out from a principal source node will reasonably guess the patterns of the flow. The initial flows assigned to the main supply branches in pipes [2], [4] and [6] is approximately one third the total flow into the system (supplied from the principal source node 10) which appears to be a rational estimate.

Table 2.5 The initially assumed flows and the balanced flows in the *Anytown* system

Pipe	Loss coefficient (Resistance) R	Initial flow Q_0 (cfs)	Final flow Q (cfs)	Head loss h_f (ft)	Head loss/1000ft
[2]	5.348	4.901	3.907	66.722	5.560
[4]	8.002	4.010	3.143	66.720	5.560
[6]	21.713	4.233	1.958	75.385	6.282
[8]	16.285	0.0	-0.0093	-0.0028	-0.0003
[10]	10.857	1.114	1.135	13.718	2.286
[12]	26.384	2.673	0.794	17.190	2.865
[14]	10.857	0.0	0.874	8.462	1.410
[16]	26.384	0.0	-0.550	-8.727	-1.455
[18]	10.857	0.0	0.746	6.315	1.053
[20]	26.384	0.0	-0.275	-2.412	-0.402
[22]	26.384	0.446	-0.275	-2.410	-0.402
[24]	26.384	0.0	-0.335	-3.472	-0.579
[26]	10.857	0.0	0.285	1.062	0.177
[28]	26.384	0.0	-0.0068	-0.0025	-0.0004
[30]	9.723	-1.114	-0.928	-8.465	-1.411
[32]	9.723	0.446	0.916	8.260	1.377
[34]	14.585	2.005	0.844	10.663	1.185
[36]	9.723	0.0	0.470	2.403	0.401
[38]	9.723	-1.114	-0.629	-4.117	-0.686
[40]	9.723	0.0	-0.020	-0.0071	-0.0012
[42]	28.828	0.0	-0.013	-0.0097	-0.0016
[44]	28.828	0.0	0.415	5.656	0.943
[46]	28.828	0.0	-0.371	-4.594	-0.766
[48]	78.223	0.0	0.228	5.056	0.843
[50]	9.723	1.782	0.193	0.462	0.077
[52]	28.828	1.337	0.424	5.873	0.979
[54]	37000.000	0.0	0.0092	6.252	0.695
[56]	28.828	0.891	-0.013	-0.0090	-0.0015
[58]	9.723	0.0	0.728	5.403	0.900
[60]	28.828	0.446	0.270	2.547	0.425
[62]	28.828	0.0	0.415	5.646	0.941
[64]	57.656	0.0	-0.176	-2.304	-0.192
[66]	57.656	-0.446	-0.240	-4.110	-0.343
[78]	0.0667	0.0	0.608	0.027	0.266
[80]	0.0667	0.0	3.527	0.688	6.884
[82]	0.00058	14.257	10.122	-290.158*	-2901.600*

* head loss includes pump head lift ($h_f - H_p$)

2.10.4 Near minimum bandwidth of the Jacobian matrix

A Jacobian matrix (Eq. 2.18) is reduced for each iteration of the Newton-Raphson technique applied to solve the loop equations. The size of the Jacobian is n by n , where n equals the number of loops, $NL+(NF-1)$. The zero-nonzero structure of the Jacobian will depend on the chosen set of loops (preferably the natural loops and minimum path pseudo loops) and the loop numbering scheme.

The loops are assigned loop identification numbers which associate the loops with rows (and columns) in the Jacobian matrix of coefficients. The coefficients of the Jacobian on the leading diagonal are nonzero. The other nonzero entries in a row of the Jacobian occur in the columns corresponding to loops with common edges. The bandwidth of the matrix is $2k+1$, where k (the half bandwidth) is the maximum number of parallel diagonals traversed by any row between the leading diagonal and a nonzero entry (Martin and Peters, 1963). The set of natural loops and pseudo loops may be reordered (assigned alternative loop identification numbers) in such a way that minimises the bandwidth of the Jacobian matrix. The bandwidth is minimised by ordering the loops so that the nonzero entries in the rows are as close as possible to the leading diagonal. Epp and Fowler (1970) described a 'loop labelling algorithm' which determines a near minimum bandwidth and hence reduces total computational time required. Epp and Fowler estimated the computational time to solve the Jacobian matrix is proportional to $(n)(k)^2$ in which n is the number of rows (i.e., the number of loops) and k is the half bandwidth. Knowledge of the bandwidth is used to speed the reduction of the Jacobian, since arithmetic computations need only be considered inside the bandwidth. The procedure described by Epp and Fowler is summarised in Steps 1 to 7 below.

Step 1. Choose a starting loop to label Loop 1 that is either a pseudo loop or a natural loop containing a node with degree=2 (a 'corner' node).

The loops containing a node with degree=2 tend to occur at 'corners' in the network.

Step 2. Consider the loop with the smallest identification number which has adjacent loops which are not yet labelled.

Step 3. The adjacent unlabelled loops are labelled by observing the rule:

The adjacent unlabelled loop next to be assigned an identification number is the one with the fewest adjacent labelled loops.

Step 4. Repeat Steps 2 to 3 until all the loops have been labelled.

Step 5. Calculate the bandwidth of the Jacobian matrix generated by the loop numbers.

The bandwidth of the Jacobian is $2k+1$, where k (the half bandwidth) is the maximum difference between the loop numbers of any two adjacent loops.

Step 6. Repeat the Steps 1 to 5 for all pseudo loops and for all natural loops containing 'corner nodes'.

Step 7. The chosen set of labelled loops is the one corresponding to the smallest bandwidth. The resulting bandwidth is a near minimum bandwidth. It is not guaranteed to be the minimum possible bandwidth.

The algorithm for determining a near minimum bandwidth is applied to the 20 loops of the Anytown pipe network (18 natural loops and 2 pseudo loops) referenced as Loops A to T in Figure 2.6. There are four potential loops to be labelled Loop 1 including the natural loops A and D with 'corner' nodes 170 and 40 respectively and the pseudo loops S and T. Loop A labelled as Loop 1 yields the smallest bandwidth of 11 (compared to 15 for loop D, 15 for loop S and 17 for loop T). The algorithm chooses the loops as 1 to 20 as shown in Figure 2.8 starting with loop A. The bandwidth equals $2k+1$, where k is equal to the maximum difference between the loop numbers of any two adjacent loops. For the configuration of loops in Figure 2.8, $k=5$ since loop 7 is adjacent to loop 12 (and loop 12 is adjacent to loop 17, etc.). Therefore, the bandwidth is $2k+1=11$.

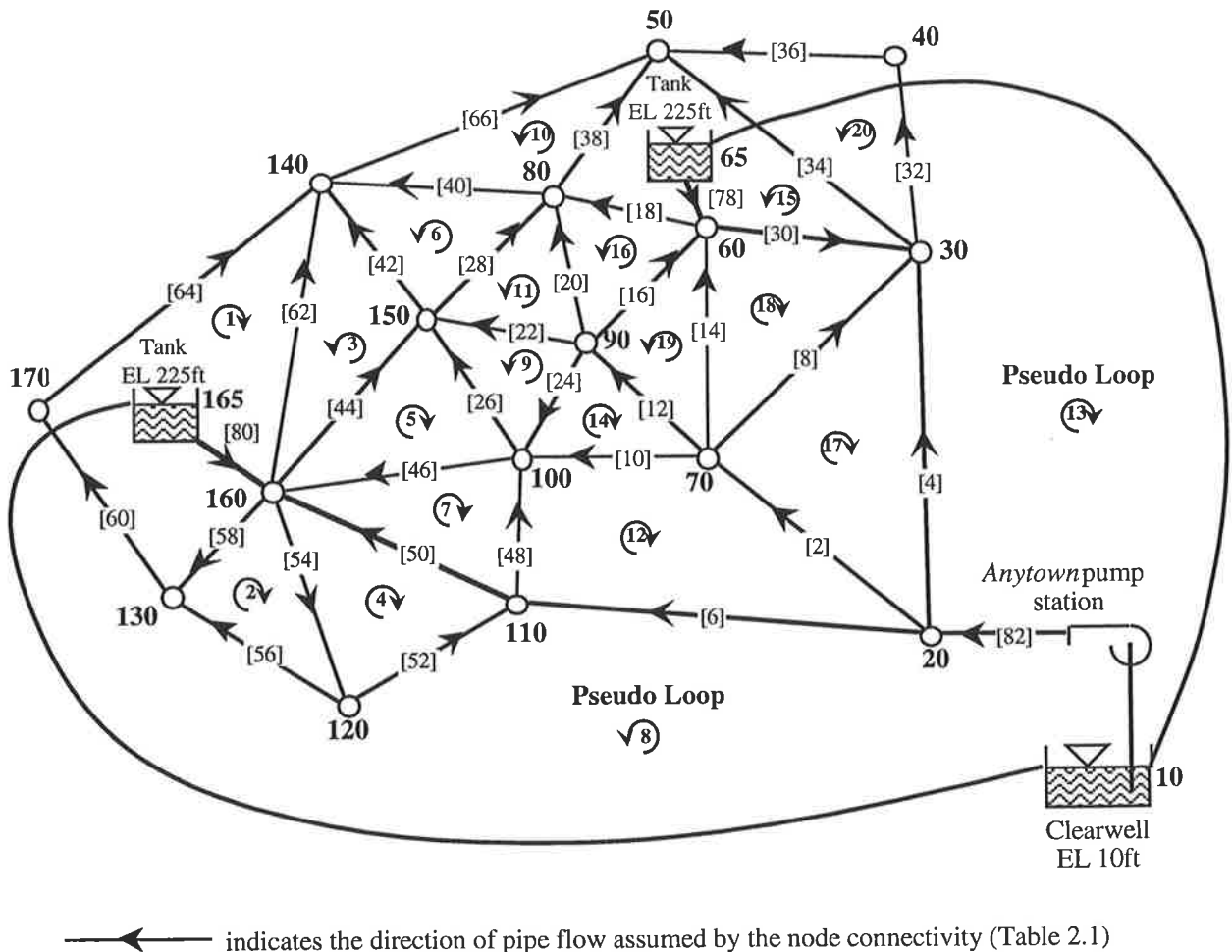


Figure 2.8 The loops ordered for near minimum bandwidth

2.10.5 The Newton-Raphson method applied to the loop equations

The Newton-Raphson method is applied to the set of $NL+(NF-1)$ nonlinear loop equations in Eq. 2.29. The loop equations are formed with the set of flow corrections around a loop of pipes as the unknowns. Flow corrections around the loops q modify assumed initial flows in the pipes Q_0 such that the balanced flow distribution is given by $(Q_0 \pm q)$. The initial flows in the network are selected such that continuity of flow at each node is satisfied. Since the flow corrections are added (or subtracted) from all the pipes in a loop, the improved flow approximations will continue to satisfy continuity. According to Eq. 2.17, the Newton-Raphson method generates the set of $NL+(NF-1)$ simultaneous linear equations in Eq. 2.32 with a set of adjustments to the flow corrections around the loops δq as the unknowns. The linear system of equations in Eq. 2.32 is equivalent to the matrix expression in Eq. 2.33 (for the *Anytown* network).

$$\frac{\partial F_l}{\partial q_1} \delta q_1 + \frac{\partial F_l}{\partial q_2} \delta q_2 + \dots + \frac{\partial F_l}{\partial q_{NL+(NF-1)}} \delta q_{NL+(NF-1)} = -F_l$$

for loops $l=1, \dots, NL+(NF-1)$ (2.32)

$$\begin{bmatrix} \frac{\partial F_1}{\partial q_1} & \frac{\partial F_1}{\partial q_2} & \dots & \frac{\partial F_1}{\partial q_{20}} \\ \frac{\partial F_2}{\partial q_1} & \frac{\partial F_2}{\partial q_2} & \dots & \frac{\partial F_2}{\partial q_{20}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_{20}}{\partial q_1} & \frac{\partial F_{20}}{\partial q_2} & \dots & \frac{\partial F_{20}}{\partial q_{20}} \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \vdots \\ \delta q_{20} \end{bmatrix} = \begin{bmatrix} -F_1 \\ -F_2 \\ \vdots \\ -F_{20} \end{bmatrix}$$

(2.33)

The coefficients in the system of linear equations in Eq. 2.32 are the functions $F_l(q)$ and partial derivatives of $F_l(q)$ evaluated at the current values of the flow corrections q . Appropriate matrix techniques are used to reduce the Jacobian matrix to solve for the adjustments to the flow corrections around the loops δq . The flow corrections q are superseded by the adjusted flow corrections $q + \delta q$ to complete one iteration of the Newton-Raphson technique applied to the loop equations. The adjusted flow corrections $q + \delta q$ move the $NL+(NF-1)$ functions $F_l(q)$ closer to zero simultaneously.

As an illustration, the functions of the loop flow corrections (i.e., the loop corrective flow equations) and the partial derivatives with respect to the loop flow corrections for natural loop 20 and pseudo loop 13 of the *Anytown* pipe network are formulated below. The functions and partial derivatives are evaluated for the first iteration of the Newton-Raphson method.

Natural Loop 20

The loop equation for natural loop **20** of the Anytown network is expressed by function F_{20} in Eq. 2.34 (according to Eq. 2.28). The natural loop **20** is formed by the path of pipes [36], [34] and [32] in an anti-clockwise direction as shown in Figure 2.8. The function F_{20} is equal to the net sum of the head losses in the pipes [36], [34] and [32].

$$F_{20} = h_{f_{36}} - h_{f_{34}} + h_{f_{32}} \quad (2.34)$$

in which h_{fk} = friction head loss in pipe k .

The direction of a pipe as indicated in Figure 2.8 represents the arbitrarily assumed direction of positive pipe flow. The actual pipe flow (and head loss) will be negative where the pipe flow direction is assumed incorrectly. The direction of a loop (clockwise or anti-clockwise) also indicated in Figure 2.8 is the arbitrarily assumed direction of positive loop flow correction.

The head losses in Eq. 2.34 are expressed in terms of pipe flows using the head loss formula such that the loop equations for natural loop **20** are expressed by the function of loop flow corrections in Eq. 2.35. The flow in pipe [36] equals the initial flow in pipe [36] plus the loop flow correction for loop **20**. Similarly, the flow in pipe [32] equals the initial flow in pipe [32] plus the flow correction for loop **20**. Since pipe [34] is common to loops **20** and **15**, the flow in pipe [34] is equal to the initial flow in pipe [34] plus the flow correction for loop **20** minus the flow correction for loop **15**. The flow correction for the common loop **15** is subtracted since the assumed direction of positive flow correction in loop **15** (clockwise) opposes the direction of positive loop flow correction in loop **20** (anti-clockwise).

$$F_{20} = R_{36}(Q_{0_{36}}+q_{20})|Q_{0_{36}}+q_{20}|^{n-1} + R_{34}(-Q_{0_{34}}+q_{20}-q_{15})|-Q_{0_{34}}+q_{20}-q_{15}|^{n-1} + R_{32}(Q_{0_{32}}+q_{20})|Q_{0_{32}}+q_{20}|^{n-1} \quad (2.35)$$

in which Q_{0k} = initial flow in pipe k , R_k = resistance of pipe k and q_l = flow correction for loop l and n = the exponent in the head loss formula.

The initial flow corrections $q^{(0)}$ for the loops are chosen as $q_l^{(0)}=q_{l-1}^{(0)}+0.05$ (with $q_1^{(0)}=0.05$) as shown in Table 2.4. The flow corrections $q^{(0)}$ are chosen in this way so that there was no possibility of a loop equation becoming redundant for the first iteration. The pipe resistances and initial pipe flow assumptions are given in Table 2.5. Hence, the function F_{20} is evaluated for the first iteration of the Newton-Raphson method:

$$F_{20} = (9.723) + (-41.334) + (19.250) = -12.37$$

2 Development of a hydraulic simulation model

The partial derivative of F_{20} with respect to q_{20} is expressed by Eq. 2.36 and evaluated for the first iteration of the Newton-Raphson technique. The sign of the partial derivatives is independent of the sign of the pipe flow (Martin and Peters, 1963) and is only dependent on the sign of the flow correction.

$$\frac{\partial F_{20}}{\partial q_{20}} = nR_{36}|Q_{0,36}+q_{20}|^{n-1} + nR_{34}|Q_{0,34}+q_{20}-q_{15}|^{n-1} + nR_{32}|Q_{0,32}+q_{20}|^{n-1} \quad (2.36)$$

$$\text{Hence, } \frac{\partial F_{20}}{\partial q_{20}} = (18.007) + (43.619) + (24.655) = 86.28$$

The partial derivative of F_{20} with respect to the flow correction in the common loop **15**, q_{15} is evaluated using Eq. 2.37. The remaining partial derivatives of F_{20} with respect to q are equal to 0.0 as stated by Eq. 2.38.

$$\frac{\partial F_{20}}{\partial q_{15}} = -nR_{34}|Q_{0,34}+q_{20}-q_{15}|^{n-1} = -43.62 \quad (2.37)$$

$$\frac{\partial F_{20}}{\partial q_l} = 0.0 \quad \{l:l=1,\dots, NL+(NF-1); l \neq 20; l \neq 15\} \quad (2.38)$$

Pseudo Loop 13

The pseudo loop **13** is formed by the path of pipes [82], [4], [30] and [78]. The function F_{13} in Eqs. 2.39 and 2.40 must account for the head losses in pipes [82], [4], [30] and [78], the pumping head gain in pipe [82], HP_{82} (with r identical parallel pumps operating), and the elevation difference between the water levels of fixed-grade nodes **10** and **65**, ΔE_{10-65} .

$$F_{13} = h_{f_{82}} - HP_{82} + h_{f_4} - h_{f_{30}} - h_{f_{78}} - \Delta E_{10-65} \quad (2.39)$$

$$\begin{aligned} F_{13} = & R_{82}(Q_{0,82}+q_{13}+q_8)|Q_{0,82}+q_{13}+q_8|^{n-1} - \left(A_{82} \left| \frac{Q_{0,82}+q_{13}+q_8}{r} \right|^2 + B_{82} \left| \frac{Q_{0,82}+q_{13}+q_8}{r} \right| + H_{0,82} \right) + \\ & R_4(Q_{0,4}+q_{13}-q_{17})|Q_{0,4}+q_{13}-q_{17}|^{n-1} + R_{30}(-Q_{0,30}+q_{13}-q_{15}-q_{18})|-Q_{0,30}+q_{13}-q_{15}-q_{18}|^{n-1} + \\ & R_{78}(-Q_{0,78}+q_{13})|-Q_{0,78}+q_{13}|^{n-1} - (E_{10} - E_{65}) \end{aligned} \quad (2.40)$$

$$\text{Hence, } F_{13} = (0.0904) - (277.780) + (95.291) + (0.174) + (0.030) - (-215.0) = 32.81$$

2 Development of a hydraulic simulation model

The partial derivatives of F_{13} with respect to q are given by Eqs. 2.41-2.46 and evaluated for the first iteration of the Newton-Raphson method. The evaluated functions and partial derivatives are inserted in the complete matrix equation shown in Eq. 2.47.

$$\frac{\partial F_{13}}{\partial q_8} = nR_{82}|Q_{0_{82}}+q_{13}+q_8|^{n-1} - \left(\frac{2A_{82}|Q_{0_{82}}+q_{13}+q_8|}{r^2} + \frac{B_{82}}{r} \right) = 2.88 \quad (2.41)$$

$$\begin{aligned} \frac{\partial F_{13}}{\partial q_{13}} = nR_{82}|Q_{0_{82}}+q_{13}+q_8|^{n-1} - \left(\frac{2A_{82}|Q_{0_{82}}+q_{13}+q_8|}{r^2} + \frac{B_{82}}{r} \right) + nR_4|Q_{0_2}+q_{13}-q_{17}|^{n-1} + \\ nR_{30}|Q_{0_{30}}+q_{13}-q_{15}-q_{18}|^{n-1} + nR_{78}|Q_{0_{78}}+q_{13}|^{n-1} \end{aligned} \quad (2.42)$$

$$\text{Hence, } \frac{\partial F_{13}}{\partial q_{13}} = (0.0109) - (-2.875) + (46.322) + (2.828) + (0.0857) = 52.12$$

$$\frac{\partial F_{13}}{\partial q_{15}} = -nR_{30}|Q_{0_{30}}+q_{13}-q_{15}-q_{18}|^{n-1} = -2.83 \quad (2.43)$$

$$\frac{\partial F_{13}}{\partial q_{17}} = -nR_4|Q_{0_2}+q_{13}-q_{17}|^{n-1} = -46.322 \quad (2.44)$$



$$\frac{\partial F_{13}}{\partial q_{18}} = -nR_{30}|Q_{0_{30}}+q_{13}-q_{15}-q_{18}|^{n-1} = -2.83 \quad (2.45)$$

$$\text{In general, } \frac{\partial F_{13}}{\partial q_l} = 0.0 \quad \{l:l=1,\dots, NL+(NF-1); l \neq 8; l \neq 13; l \neq 15; l \neq 17; l \neq 18\} \quad (2.46)$$

The complete Jacobian matrix equation of functions and partial derivatives evaluated at the starting loop flow corrections $q^{(0)}$ is formulated in Eq. 2.47 for the first iteration of the Newton-Raphson method. The Jacobian matrix is reduced using sparse matrix routines (Section 2.10.7) to solve for the adjustments δq to the loop flow corrections. The adjustments δq to the loop flow corrections $q^{(0)}$ and the improved loop flow corrections $q^{(1)}=q^{(0)}+\delta q$ for the first iteration are given in Table 2.4. A new matrix equation of functions and partial derivatives evaluated at the improved flow corrections $q^{(1)}$ is set up and solved and the procedure is repeated until some specified convergence criterion (Section 2.10.6) is met.

Loop	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
1	52.62	-1.40	13.55	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	δq_1	-9.50
2	-1.40	9688.6	0.0	-9634.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	δq_2	491.82
3	13.55	0.0	48.61	0.0	24.46	-10.60	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	δq_3	-5.89
4	0.0	-9634.2	0.0	9748.3	0.0	0.0	-37.05	-37.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	δq_4	-537.43
5	0.0	0.0	24.46	0.0	46.80	0.0	-7.51	0.0	14.84	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	δq_5	-10.49
6	0.0	0.0	-10.60	0.0	0.0	30.17	0.0	0.0	0.0	-4.57	-15.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	δq_6	1.66
7	0.0	0.0	0.0	-37.05	-7.51	0.0	89.02	37.05	0.0	0.0	0.0	-44.47	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	δq_7	-41.06
8	0.0	0.0	0.0	-37.05	0.0	0.0	37.05	204.70	0.0	0.0	0.0	164.71	2.88	0.0	0.0	0.0	0.0	0.0	0.0	0.0	δq_8	-449.32
9	0.0	0.0	0.0	0.0	14.84	0.0	0.0	0.0	89.64	0.0	-19.76	0.0	0.0	55.04	0.0	0.0	0.0	0.0	0.0	0.0	δq_9	-43.47
10	0.0	0.0	0.0	0.0	0.0	-4.57	0.0	0.0	0.0	129.83	0.0	0.0	0.0	0.0	-23.46	0.0	0.0	0.0	0.0	0.0	δq_{10}	-35.20
11	0.0	0.0	0.0	0.0	0.0	15.00	0.0	0.0	-19.76	0.0	49.76	0.0	0.0	0.0	0.0	-15.00	0.0	0.0	0.0	0.0	δq_{11}	3.69
12	0.0	0.0	0.0	0.0	0.0	0.0	-44.47	164.71	0.0	0.0	0.0	272.92	0.0	-23.72	0.0	0.0	-40.03	0.0	0.0	0.0	δq_{12}	-344.49
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.88	0.0	0.0	0.0	0.0	52.12	0.0	-2.83	0.0	-46.32	-2.83	0.0	0.0	δq_{13}	-32.78
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	55.04	0.0	0.0	-23.72	0.0	128.59	0.0	0.0	0.0	0.0	49.83	0.0	δq_{14}	-22.19
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	23.46	0.0	0.0	-2.83	0.0	71.47	-1.57	0.0	2.83	0.0	-43.62	δq_{15}	-58.39
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-15.00	0.0	0.0	0.0	-1.57	26.27	0.0	0.0	-9.71	0.0	δq_{16}	-1.28
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-40.03	-46.32	0.0	0.0	0.0	88.70	-2.35	0.0	0.0	δq_{17}	-15.98
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-2.83	0.0	2.83	0.0	-2.35	39.14	33.96	0.0	δq_{18}	-33.81
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	49.83	0.0	-9.71	0.0	33.96	93.50	0.0	δq_{19}	-7.18
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-43.62	0.0	0.0	0.0	0.0	86.28	δq_{20}	12.37

Equation 2.47 The complete matrix equation representing the first iteration of the Newton-Raphson method applied to the loop equations of the Anytown pipe network (assuming the starting loop flow corrections in Table 2.4 and assuming the initial pipe flows in Table 2.5)

 Leading diagonal
 Bandwidth

2.10.6 Convergence test

The purpose of a convergence test is to determine whether the Newton-Raphson method has converged to a solution. In addition, a maximum number of iterations is specified. Wood and Rayes (1981) used a convergence criterion which stopped iterations when the average change in flows in the pipes between successive iterations is less than 0.5%. A similar convergence test is employed in this thesis. Computations are terminated when the average change in flow corrections around the loops between successive iterations, m is less than some specified relative accuracy, Δq_{avg} and the maximum change in flow corrections between successive iterations is less than some other specified accuracy, Δq_{max} . The convergence test is summarised by Eqs. 2.48 and 2.49.

$$\frac{\sum_{l=1}^{NL+(NF-1)} (q_l^{(m)} - q_l^{(m-1)})}{\sum_{l=1}^{NL+(NF-1)} q_l^{(m)}} \leq \Delta q_{avg} \quad (2.48)$$

$$\max \left(\frac{q_l^{(m)} - q_l^{(m-1)}}{q_l^{(m)}} \right) \leq \Delta q_{max} \quad \text{for loops } l=1, \dots, NL+(NF-1) \quad (2.49)$$

in which $q_l^{(m)}$ = computed flow correction for loop l for the current iteration m and $q_l^{(m-1)}$ = computed flow correction for loop l for the previous iteration $m-1$.

Based on an average change $\Delta q_{avg}=0.1\%$ and a maximum change $\Delta q_{max}=0.5\%$, calculations were terminated after only 6 iterations of the Newton-Raphson method applied to the loop corrective flow equations for the *Anytown* network. The convergence to the solution for the analysis of the *Anytown* system is shown in Table 2.6.

Table 2.6 Convergence of the Newton-Raphson method applied to the loop equations for the *Anytown* network

Iteration	Maximum change in q (%)	Average change in q (%)
1	260.2 (loop 8)	114.1
2	123.5 (loop 8)	56.7
3	13.8 (loop 5)	7.26
4	1.70 (loop 3)	0.72
5	0.73 (loop 2)	0.18
6	0.27 (loop 2)	0.065

2 Development of a hydraulic simulation model

The final computed flow corrections for the loops after 6 iterations $q^{(6)}$ are presented in Table 2.4 and the balanced pipe flows are presented in Table 2.5. The *Anytown* source pump station contributes most (71.0%) of the total network demand for the average daily demand flows with the elevated tanks **65** and **165** at their low water levels (225.0 ft). Tank **65** supplies 4.3% of the network demand while tank **165** supplies 24.7% of the network demand. The flow fans out from node **20** and then from nodes **30**, **70** and **110**. Pipes [28], [40] and [42] and pipes [54], [56] and [8] convey relatively small flows.

The accuracy of the solution is checked by computing the unbalanced heads around the loops as shown in Table 2.7. The relatively high unbalanced heads for loops **2** and **4** may be attributed to the very high resistance of pipe [54] which is common to both loops. The very high resistance of pipe [54] results in a significant head loss for a very low flow. The unbalanced heads in loops **2** and **4** could be reduced by applying a more stringent convergence criterion.

Table 2.7 The unbalanced head losses around the loops

Loop	Path of pipes	Unbalanced head losses (ft)
1	64 -62 58 60	0.00002
2	-58 54 56	0.83986
3	44 42 -62	0.00000
4	-54 -50 52	-0.84020
5	44 -26 46	0.00000
6	-42 28 40	0.00000
7	50 -46 -48	0.00001
8 (Pseudo)	82 6 50 -80	-214.99999
9	-26 -24 22	0.00000
10	-40 38 -66	0.00000
11	-22 20 -28	0.00000
12	48 -10 -2 6	0.00000
13 (Pseudo)	82 4 -30 -78	-215.00000
14	-24 -12 10	0.00000
15	-38 -18 30 34	0.00000
16	-20 16 18	0.00000
17	2 8 -4	0.00000
18	-8 14 30	0.00000
19	-12 14 -16	0.00000
20	36 -34 32	0.00000

2.10.7 Sparse matrix routines

The chosen set of loops (preferably natural loops and minimum path pseudo loops) and the numbering of the loops (preferably for near minimum bandwidth) determines the structure of the Jacobian matrix to be reduced to solve the set of loop equations. To increase the efficiency of the adopted method of hydraulic analysis, sparse matrix routines were introduced to efficiently reduce the sparse Jacobian matrices. The Jacobian matrices are scaled and a practical and efficient order of pivot elements is determined before the matrices are reduced.

A matrix is said to be sparse if it is comprised of only a small percentage of nonzero elements. An n by n matrix is classified as sparse if it is comprised of to the order of only n nonzero elements. The Newton-Raphson method applied to the loop equations requires the solution of a system of $n=NL+(NF-1)$ linearised equations:

$$J \delta q = -F \quad (2.50)$$

or

$$\delta q = -J^{-1} F \quad (2.51)$$

in which J = the $(NL+(NF-1))$ by $(NL+(NF-1))$ Jacobian matrix of partial derivatives evaluated at the current values of the flow corrections, J^{-1} = the inverse of the Jacobian, F = the $(NL+(NF-1))$ by 1 matrix of functions evaluated at the current values of the loop flow corrections, δq = the $(NL+(NF-1))$ by 1 matrix of the unknown loop flow corrections.

The $(NL+(NF-1))$ by $(NL+(NF-1))$ Jacobian matrix of coefficients J is usually sparse. Eq. 2.50 was written in full in Eq. 2.47 for the first iteration of the Newton-Raphson method applied to the 20 loop equations of the *Anytown* pipe network. The Jacobian produced by the Newton-Raphson method applied to the loop equations typically comprises of 3 or 4 nonzero elements per row of $NL+(NF-1)$ elements. The rows (and columns) of the Jacobian correspond to loops of the network. The number of nonzeros in a row of the Jacobian depends on the number of loops adjacent (with common pipes) to the loop associated with this row. The Jacobian matrix will usually have many zeros and it is worthwhile using sparse matrix techniques that avoid operating with the zeros.

The augmented Jacobian matrix J' is comprised of the matrix $-F$ appended to the Jacobian J such that:

$$J' = [J \mid -F] \quad (2.52)$$

Gauss-Jordan Elimination

The method of solution of Eq. 2.50 considered is based on Gauss-Jordan elimination. The pivot elements $J'(R,C)$ are chosen on the leading diagonal (where rows $r = R$ and columns $c = C$ such that $R = C$) using the strategy described below. The nonzero elements above and below the pivot element are eliminated by row reduction operations. A set of row reduction operations for a sequence of pivot elements are applied to the augmented Jacobian matrix J' which reduces J to the identity matrix. The same row operations applied to the right hand side of the augmented matrix J' yields the solution δq .

The pivot elements were chosen on the leading diagonal since in general these elements are the most suitable pivots. Pivots on the leading diagonal help to maintain a minimum bandwidth and directly reduce the Jacobian matrix to the identity matrix. Having chosen an appropriate pivot element $J'(R,C)$, the Gauss-Jordan elimination considers the rows r with nonzero elements in the pivot column C . The rows r are operated on by the pivot row R for all the columns c with nonzero elements in the pivot row $J'(R,c)$. The element $J'_{i+1}(r,c)$ for the i th operation of the Gauss-Jordan reduction overwrites the element $J'_i(r,c)$ according to the formula:

$$J'_{i+1}(r,c) = J'_i(r,c) - \frac{J'_i(R,c) J'_i(r,C)}{J'_i(R,C)} \quad r \neq R \quad (2.53)$$

The nonzero elements in the pivot column are reduced to 0.0 since if the column c equals the pivot column C in Eq. 2.53 then:

$$J'_{i+1}(r,C) = J'_i(r,C) - J'_i(r,C) = 0.0 \quad (2.54)$$

The pivot row R is divided by the pivot element such that the pivot element $J'(R,C)=1.0$.

$$J'(R,c) = \frac{J'(R,c)}{J'(R,C)} \quad (2.55)$$

The operations for columns c in rows r ($r \neq R$) given by Eq. 2.53 and for columns c in row R given by Eq. 2.55 need only be performed for elements within the bandwidth of the Jacobian. The half bandwidth k is the maximum number of columns traversed between the leading diagonal and another nonzero element in the same row. The half bandwidth of the Jacobian for the *Anytown* network in Eq. 2.47 was reduced to $k=5$ by rearranging the loop identification numbers in Section 2.10.4. No arithmetic operations are performed beyond the bandwidth in the pivot column and pivot row. It is unnecessary to visit elements beyond the bandwidth when the Jacobian is originally generated and scaled. Should an operation in Eq. 2.53 overwrite an

2 Development of a hydraulic simulation model

element $J'(r,c)$ where $|r-c|>k$, then the half bandwidth becomes $k=|r-c|$. The operations in Eq. 2.53 and 2.55 are repeated for a chosen sequence of $NL+(NF-1)$ pivot elements.

Gaussian elimination pivots about the elements of the leading diagonal working from the top left hand corner and systematically eliminating the elements below the leading diagonal in forward elimination. The unknowns are computed by a back substitution through the modified linear equations above the leading diagonal. In Gaussian elimination, the number of operations are halved compared to Gauss-Jordan elimination, however Gaussian elimination requires a series of row and column interchanges to bring the i th pivot element to the leading diagonal in the i th row. The row and column interchanges will not help to maintain the minimum bandwidth.

Row Scaling

The magnitude of the nonzero elements of the Jacobian matrix may differ significantly from the magnitude of other nonzero elements in the same row (or column). Numerical instability may arise when large numbers are added to small numbers during the reduction. The difference in the magnitudes of element $J(2,1)$ and element $J(2,2)$ in the Jacobian for the *Anytown* network (before scaling) in Eq. 2.47 would indicate the potential danger of numerical instability during the row reduction operations. A simple method of scaling the matrix before the reduction begins can be applied to assure the matrix elements are of comparable magnitude. Row scaling consists of dividing each row by the element having the largest absolute value in that row (Tewarson, 1973). The scaled, augmented Jacobian J' for the *Anytown* network derived from the matrix equation in Eq. 2.47 is presented in Eq. 2.56.

Any computational effort spent to efficiently reduce the sparse Jacobian matrix J is justified since Jacobian matrices for each iteration of the Newton-Raphson technique have identical zero-nonzero structures with numerical values of a comparable scale (after scaling). The Jacobian matrices associated with identical pipe network layouts (proposed by the optimisation model) have identical zero-nonzero structures. A sequence of pivot elements to reduce the sparse matrix by Gauss-Jordan elimination may be established and stored to be applied to all the identical matrices encountered. A substantial amount of time can be saved if trivial operations involving zeros are not performed. The sequence of pivots helps to maintain numerical stability and at the same time limits the amount of fill-in.

Loop	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
1	1.0	-0.027	0.257	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.181	
2	-0.00014	1.0	0.0	-0.994	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0508
3	0.279	0.0	1.0	0.0	0.503	-0.218	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.121
4	0.0	-0.988	0.0	1.0	0.0	0.0	-0.0038	-0.0038	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0551
5	0.0	0.0	0.523	0.0	1.0	0.0	-0.160	0.0	0.317	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.224
6	0.0	0.0	-0.351	0.0	0.0	1.0	0.0	0.0	0.0	-0.151	-0.497	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0550
7	0.0	0.0	0.0	-0.416	-0.0843	0.0	1.0	0.416	0.0	0.0	0.0	-0.499	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.461
8	0.0	0.0	0.0	-0.0825	0.0	0.0	0.0825	0.456	0.0	0.0	0.0	0.367	0.0064	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
9	0.0	0.0	0.0	0.0	0.166	0.0	0.0	0.0	1.0	0.0	-0.220	0.0	0.0	0.614	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.485
10	0.0	0.0	0.0	0.0	0.0	-0.0352	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	-0.181	0.0	0.0	0.0	0.0	0.0	0.0	-0.271
11	0.0	0.0	0.0	0.0	0.0	-0.301	0.0	0.0	-0.397	0.0	1.0	0.0	0.0	0.0	0.0	-0.301	0.0	0.0	0.0	0.0	0.0	0.0741
12	0.0	0.0	0.0	0.0	0.0	0.0	-0.129	0.478	0.0	0.0	0.0	0.792	0.0	-0.0688	0.0	0.0	-0.116	0.0	0.0	0.0	0.0	-1.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0553	0.0	0.0	0.0	0.0	1.0	0.0	-0.0543	0.0	-0.889	-0.0543	0.0	0.0	0.0	-0.629
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.428	0.0	0.0	-0.184	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.388	0.0	-0.173
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.328	0.0	0.0	-0.0396	0.0	1.0	-0.0219	0.0	0.0396	0.0	-0.610	0.0	-0.817
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.571	0.0	0.0	0.0	-0.0596	1.0	0.0	0.0	-0.369	0.0	0.0	-0.0488
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.451	-0.522	0.0	0.0	0.0	1.0	-0.0265	0.0	0.0	0.0	-0.180
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0723	0.0	0.0723	0.0	-0.0600	1.0	0.868	0.0	0.0	-0.864
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.533	0.0	-0.104	0.0	0.363	1.0	0.0	0.0	-0.0768
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.506	0.0	0.0	0.0	0.0	0.0	1.0	0.143

Equation 2.56 The scaled, augmented Jacobian matrix J' representing the first iteration of the Newton-Raphson method applied to the loop equations of the Anytown pipe network

 Leading diagonal
 Bandwidth

Strategy for Selecting Suitable Pivot Elements

Markowitz threshold pivoting is a successful and widely used criterion for choosing which element is the next most suitable pivot (Parisi, 1981). The pivot is chosen as the nonzero element $J'(R,C)$ which minimises the product in Eq. 2.57 and which satisfies one of the two conditions in Eqs. 2.58 and 2.59.

$$(NZC)(NZR) \tag{2.57}$$

in which NZC = the number of (columns with) nonzero elements in row R and NZR = the number of (rows with) nonzero elements in column C for the proposed pivot element $J'(R,C)$.

$$|J'(R,C)| \geq \max(|J'(r,C)| \cdot u) \quad \text{for all rows } r=1, \dots, NL+(NF-1) \tag{2.58}$$

or

$$|J'(R,C)| \geq \max(|J'(R,c)| \cdot u) \quad \text{for all columns } c=1, \dots, NL+(NF-1) \tag{2.59}$$

The threshold parameter, u is specified in the interval [0.0, 1.0]. Choosing $u=1.0$ is to pivot to maintain numerical stability alone while choosing $u=0.0$ is to pivot to maintain sparsity alone. A value of $u=0.1$ is recommended as being a good compromise (Parisi 1981).

In this research, a modified form of Markowitz threshold pivoting is employed to determine the next pivot element. The pivot is chosen as the element $J'(R,C)$ such that:

- (1) $R = C$ (i.e. the pivot element $J'(R,C)$ is on the leading diagonal)
- (2) one of the two conditions in Eqs. 2.58 and 2.59 is satisfied
- (3) the product in Eq. 2.60 is minimised:

$$\left(\sum_{c'=1}^{NZC} |c' - C| \right) \left(\sum_{r'=1}^{NZR} |r' - R| \right) \quad \text{for columns } c' \text{ such that } J'(R,c') \neq 0.0 \text{ and} \\ \text{for rows } r' \text{ such that } J'(r',C) \neq 0.0 \tag{2.60}$$

In words, Eq. 2.57 is the product of the number of nonzero elements in the pivot row R and the number of nonzero elements in the pivot column C . The minimisation of this product helps to minimise the amount of fill-in caused by the row operations. The minimisation of the modified product in Eq. 2.60 may be appropriate for the reduction here since the modified product takes into account the distance of the nonzero elements $J'(R,c')$ and $J'(r',C)$ from the pivot element $J'(R,C)$. The further these elements are from the pivot element on the leading diagonal, the greater the chance the half bandwidth will be extended when a nonzero element is placed beyond the bandwidth by a row reduction operation.

The first pivot element is chosen for the scaled Jacobian in Eq. 2.57 as shown in Table 2.8. The pivot element is chosen as $J'(20,20)$ by Markowitz threshold pivoting and $J'(2,2)$ by the modified Markowitz threshold pivoting. The elements of the leading diagonal are usually the element of greatest magnitude in a row or column. The modified product in Eq. 2.60 favours element $J'(1,1)$ or $J'(2,2)$ over element $J'(20,20)$ for the first pivot element even though rows 1 and 2 contain more nonzero elements than row 20, since the nonzero elements in rows 1 and 2 are closer to the leading diagonal.

Table 2.8 Potential pivot elements for the scaled Jacobian matrix for the Anytown network in Eq. 2.56

Element $J'(R,C)^\dagger$	Product (Eq. 2.57)	Modified product (Eq. 2.60)	Max. element in row R	Max. element in column C
$J'(1,1)=1.0$	9	9	1.0	1.0
$J'(2,2)=1.0$	9	9*	1.0	1.0
$J'(3,3)=1.0$	16	49	1.0	1.0
$J'(4,4)=1.0$	16	81	1.0	1.0
$J'(5,5)=1.0$	16	64	1.0	1.0
$J'(6,6)=1.0$	16	144	1.0	1.0
$J'(7,7)=1.0$	25	121	1.0	1.0
$J'(8,8)=0.456$	25	196	0.456	0.478
$J'(9,9)=1.0$	16	121	1.0	1.0
$J'(10,10)=1.0$	9	81	1.0	1.0
$J'(11,11)=1.0$	16	144	1.0	1.0
$J'(12,12)=0.792$	25	256	0.792	0.792
$J'(13,13)=1.0$	25	256	1.0	1.0
$J'(14,14)=1.0$	16	144	1.0	1.0
$J'(15,15)=1.0$	36	256	1.0	1.0
$J'(16,16)=1.0$	16	81	1.0	1.0
$J'(17,17)=1.0$	16	100	1.0	1.0
$J'(18,18)=1.0$	25	100	1.0	1.0
$J'(19,19)=1.0$	16	81	1.0	1.0
$J'(20,20)=1.0$	4*	25	1.0	1.0

† the pivot element is selected on the leading diagonal such that $R=C$

* chosen pivot elements

2 Development of a hydraulic simulation model

The order of the pivot elements chosen by Markowitz threshold pivoting and the modified Markowitz threshold pivoting are given in Tables 2.9 and 2.10 respectively. Comparison of the adjusted half bandwidths throughout the matrix reduction confirm the modified pivoting strategy maintains a smaller bandwidth for longer, however, the total number of operations for the Markowitz threshold pivoting is 862 which is 4.1% less than the total number of operations for the modified Markowitz threshold pivoting which is 899. The average number of nonzero elements at each step of the reduction for Markowitz threshold pivoting is 83.55 which is just less than for the modified pivoting order which is 84.75.

Either strategy for selecting pivot elements is suitable for this research. Although more operations are performed during the matrix reduction using the alternative pivot order, the operations are confined to thinner bands of nonzero elements for longer.

The element on the leading diagonal always represents the element of greatest magnitude within the row for the chosen pivots which helps maintain numerical stability. The threshold parameter was set at $u=0.1$, however a value of $u=1.0$ would not have changed the chosen order of the pivot elements.

Table 2.9 Chosen pivot order for Markowitz threshold pivoting

Pivot element	Product Eq. 2.56	Max. row element	Max. column element	Adjusted half bandwidth	Number of operations	Number of nonzero elements
$J'(20,20)=1.0$	4	1.0	1.0	5	6	81
$J'(10,10)=1.0$	9	1.0	1.0	9	12	81
$J'(2,2)=1.0$	9	1.0	1.0	9	12	81
$J'(1,1)=1.0$	12	1.0	1.0	9	16	81
$J'(19,19)=1.0$	16	1.0	1.0	9	20	84
$J'(17,17)=1.0$	16	1.0	1.0	9	20	85
$J'(11,11)=1.0$	16	1.0	1.0	10	20	88
$J'(5,5)=1.0$	16	1.0	1.0	10	20	91
$J'(4,4)=0.017$	24	0.017	0.994	10	30	92
$J'(14,14)=0.794$	30	0.794	0.794	10	36	95
$J'(13,13)=0.536$	30	0.536	0.536	10	36	98
$J'(6,6)=0.845$	35	0.845	0.845	13	42	103
$J'(7,7)=0.895$	45	0.895	0.895	13	54	107
$J'(18,18)=0.586$	60	0.586	0.586	13	70	108
$J'(16,16)=0.716$	84	0.716	0.716	16	98	106
$J'(15,15)=0.621$	75	0.621	0.621	17	90	96
$J'(12,12)=0.596$	80	0.596	0.883	17	100	77
$J'(9,9)=0.363$	60	0.363	0.699	17	80	58
$J'(8,8)=0.027$	40	0.027	0.945	17	60	39
$J'(3,3)=0.300$	20	0.300	0.925	17	40	20
					TOTAL=862	AVG=83.55

Table 2.10 Chosen pivot order for modified Markowitz threshold pivoting

Pivot element	Modified product Eq. 2.58	Max. row element	Max. column element	Adjusted half bandwidth	Number of operations	Number of nonzero elements
$J'(2,2)=1.0$	9	1.0	1.0	5	12	82
$J'(20,20)=1.0$	25	1.0	1.0	5	6	81
$J'(1,1)=1.0$	30	1.0	1.0	5	16	81
$J'(3,3)=0.928$	54	0.928	0.928	5	30	86
$J'(19,19)=1.0$	81	1.0	1.0	5	20	89
$J'(10,10)=1.0$	81	1.0	1.0	9	12	89
$J'(17,17)=1.0$	100	1.0	1.0	9	20	90
$J'(9,9)=1.0$	121	1.0	1.0	9	20	93
$J'(7,7)=1.0$	121	1.0	1.0	9	30	95
$J'(16,16)=0.962$	130	0.962	0.962	9	36	98
$J'(13,13)=0.536$	221	0.536	0.536	10	36	101
$J'(4,4)=0.016$	360	0.016	0.994	11	54	104
$J'(14,14)=0.522$	589	0.522	0.614	14	63	109
$J'(11,11)=0.664$	943	0.664	0.664	14	70	106
$J'(8,8)=0.414$	1638	0.414	0.531	17	91	108
$J'(12,12)=0.075$	2024	0.075	0.983	17	108	91
$J'(6,6)=0.679$	2332	0.679	0.679	17	95	75
$J'(15,15)=0.623$	1560	0.623	0.623	17	80	58
$J'(5,5)=0.418$	1690	0.418	0.648	17	60	39
$J'(18,18)=0.177$	0	0.177	0.829	17	40	20
					TOTAL=899	AVG=84.75

Packed Forms of Storage

The zero-nonzero structure of the sparse Jacobian matrix J is symmetric when the Newton-Raphson method is applied to the loop equations. In general, the elements of the leading diagonal are nonzero elements. The pipes common to loop r and loop c yields nonzero elements $J(r,c)$ and $J(c,r)$. A packed form of storage is investigated which makes use of this characteristic. For this research, the pipes common to two or more loops were identified so that an array could be constructed which contained the addresses of the nonzero elements (other than the leading diagonal).

2.10.8 Junction node pressure heads

Pipe head losses are determined from the final balanced pipe flows using the friction head loss formula and pump heads are calculated using pump head characteristic curves. An algorithm finds the pressure heads at the junction nodes in the network by starting at any fixed-grade node and proceeding through the network using a Breadth First Search (BFS) to systematically visit all the junction nodes. The pressure head p_d at a downstream junction node, d is determined from the pressure head p_u at a connecting upstream node, u by the Bernoulli equation (Eq. 2.2). The computed pressure heads at the junction nodes in the existing *Anytown* system for the

average daily demand pattern are presented in Table 2.11. There are no violations of the minimum allowable pressure head constraints of 40 psi. The lowest pressure is 41.70 psi at node 170.

Table 2.11 The balanced pressure heads at the junction nodes for the Anytown system

Junction node	Actual pressure (psi)	Minimum allowable pressure (psi)	Surplus pressure (psi)
20	121.22	40.0	81.22
30	79.37	40.0	39.37
40	75.80	40.0	35.80
50	74.76	40.0	34.76
60	75.71	40.0	35.71
70	79.37	40.0	39.37
80	72.98	40.0	32.98
90	71.94	40.0	31.94
100	73.44	40.0	33.44
110	75.63	40.0	35.63
120	42.43	40.0	2.43
130	42.80	40.0	2.80
140	60.00	40.0	20.00
150	42.69	40.0	2.69
160	45.14	40.0	5.14
170	41.70	40.0	1.70

2.10.9 Analysis of pressure reducing valves and check valves

A pressure reducing valve (PRV) maintains a constant pressure immediately downstream of the valve equal to the valve pressure setting, assuming the pressure immediately upstream of the valve is greater than the valve pressure setting. There are three possible modes of operation of a PRV, however the mode of operation just described is the functional mode of the PRV.

PRVs may be used to reduce pressure in downstream regions or subnetworks of the water distribution system. In the absence of a PRV, uncontrolled pressures may be high and considerably variable with time. PRVs serving lower zones are sometimes considered as demand points in a water transfer system and the same PRV may represent the source node of a more detailed subnetwork¹ of distribution pipes. The PRV may control the pressure at the upstream end of the subnetwork and limit the available flow to the subnetwork (Perez et al., 1993). PRVs may be used to maintain pressure within operating limits to protect equipment against damage due to extreme pressures. For networks with more than one source of supply,

¹ Jeppson (1976) suggested the decomposition of pipe networks into subnetworks where possible to improve computational efficiency. For example, a subnetwork may be supplied by a single pipe or isolated by one or more PRVs.

2 Development of a hydraulic simulation model

PRVs may be operated to adjust the flow distribution and control the flows to and from the source nodes for some demand conditions.

The PRV with pressure setting (H_{set}) may assume one of three different modes of operation which depends on the inlet pressure (H_{in}) and the outlet pressure (H_{out}) of the PRV (El-Bahrawy and Smith, 1987).

(1) The operative mode

If the PRV inlet pressure is greater than or equal to the PRV setting ($H_{in} > H_{set}$) as shown in Figure 2.9, then the PRV is operating (throttling) and the PRV maintains a fixed outlet pressure equal to the PRV setting ($H_{out} = H_{set}$).

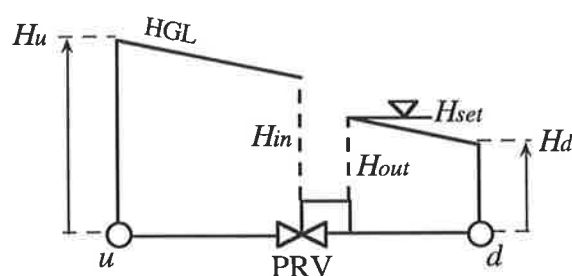


Figure 2.9 The operative mode for a PRV

(2) The inoperative mode

If the PRV inlet pressure is less than the PRV pressure setting ($H_{in} < H_{set}$) as shown in Figure 2.10, then the PRV outlet pressure equals the inlet pressure minus minor losses through the open valve which may be negligible ($H_{out} \approx H_{in}$).

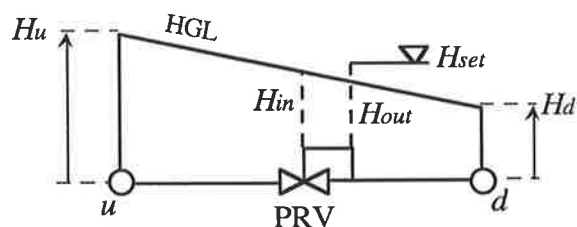


Figure 2.10 The inoperative mode for a PRV

(3) The closed check valve mode

If the outlet pressure exceeds the inlet pressure ($H_{out} > H_{in}$), the PRV acts as a check valve (CV) and prevents reverse flow.

2 Development of a hydraulic simulation model

The hydraulic simulation model described previously may be modified to allow for the inclusion of PRVs and to adjust the analysis to determine the eventual mode of operation of the PRVs. Jeppson (1976) described modifications to each of the formulations of the pipe network equations including the loop corrective flow equations (ΔQ_L -equations) to simulate PRVs and check valves (CVs) in the hydraulic analysis. The loop equations write $NL+(NF-1)$ energy equations in the unknown flow corrections around the NL natural loops and $(NF-1)$ pseudo loops. The energy equations for loops with PRVs are only valid if the PRV inlet pressure is less than the PRV setting (inoperative mode). The PRV in the inoperative mode has negligible effect on the pressure distribution in the network. The network may be analysed as though the PRV did not exist at all.

The head drop across an operational PRV is not a function of the flow through the PRV and is therefore independent of the flow corrections in any loop which contains the PRV. The energy equations for the loops which include an operational PRV are no longer valid. Natural loops and pseudo loops may be affected by an operating PRV.

Jeppson (1976) analysed the operating PRV as an artificial fixed-grade node, with HGL equal to the PRV pressure setting. The pipe upstream of the PRV is removed from the network. If the PRV is considered to be an artificial fixed-grade node, continuity is upset since flow is extracted from the PRV without being added to the upstream node. Therefore, the loop flow corrections are assumed to circulate around the original set of loops, l defined as if there were no PRVs. The corrections applied to the flows in the pipes in the original loops will produce a flow distribution which continues to satisfy continuity. However, the energy equations (Eq. 2.29) are written around the new set of loops, l' which are formed assuming the operating PRV is an artificial fixed-grade node. A new pseudo loop is formed and a natural loop removed for each operating PRV in the modified set of loops, l' . The final pipe flows satisfy continuity at the nodes for the original set of natural loops and pseudo loops, l and the head losses are zero around the modified set of natural loops and pseudo loops, l' .

The new set of loops may be determined using the procedures described in Section 2.5.2. Alternatively, the modified set of loops may be determined from the original set of loops by:

- (1) deleting the natural loop(s) which contains the pipe with the operating PRV,
- (2) defining a new natural loop if the pipe with the PRV is common to two original natural loops (the new natural loop is formed by the incomplete paths of the two original natural loops),
- (3) defining an alternative path of pipes for an original pseudo loop(s) if the original pseudo loop(s) contains the pipe with the PRV, and
- (4) by identifying a new pseudo loop between the PRV and a fixed-grade node.

2 Development of a hydraulic simulation model

These modifications to the original loops avoids the computational effort (especially for large networks) of defining and numbering a complete new set of loops which are quite similar to the original set of loops.

A PRV does not allow the direction of flow to be in the opposite direction and under these conditions the PRV operates as a closed check valve (CV).

There are two different modes of operation of a CV (open or closed). The inclusion of a CV does not affect the analysis if the flow is in the permitted direction as shown in Figure 2.11. If the flow is in the reverse direction, then the CV closes and there is no flow in the pipe as shown in Figure 2.12. A closed CV may be analysed by assuming the pipe with no flow does not exist. The network is analysed again with the pipe removed (the flow in this pipe must zero in the initial flow pattern).

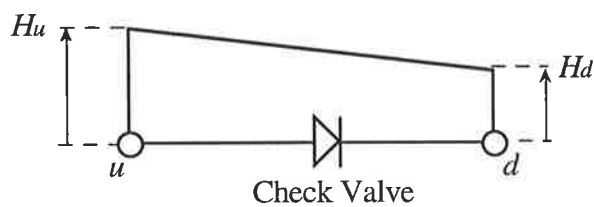


Figure 2.11 The open check valve

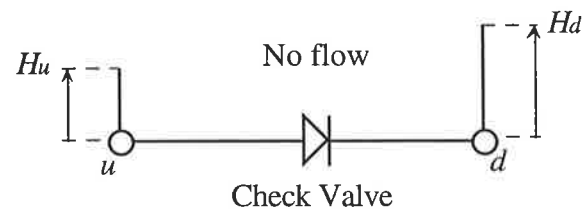


Figure 2.12 The closed check valve

Jeppson (1976) analyses a closed CV by adjusting the flow corrections in the loop with the closed CV by the initial flow in the closed pipe which need not be zero for this method of analysis. For example, if pipe j is closed and belongs to loop K then:

$$q_K = -Q_0 \pm q_L \quad (2.61)$$

in which L are the other loops which contain the closed pipe j . The flow Q_j in pipe j is:

$$Q_j = Q_0 \pm q = 0.0 \quad (2.62)$$

for a closed CV in pipe j . The number of equations is reduced by 1, since the expression in Eq. 2.61 is substituted for q_K where this correction is applied to pipes in the loops in the energy equations.

Salgado et al. (1988) reviewed a number of the traditional methods of hydraulic analysis of water supply systems and warned of a number of problems that may arise when using some methods proposed to simulate pressure regulating devices. The loop equations may experience

2 Development of a hydraulic simulation model

difficulty in dealing with multiple pressure regulating devices. The assumed mode of operation for these devices is checked after an appropriate number of iterations and the analysis must be repeated if the devices are not operating according to the initially assumed conditions. Using Jeppson's (1976) modifications to the loop equations to simulate PRVs, the loop structure must be modified if the mode of operation of the PRV is different to the assumed mode of operation.

The method of hydraulic analysis used by Salgado et al. (1988) is the gradient method. The gradient method (Todini and Pilati, 1987) is based on a "formal dissipated power approach" and operates on the heads and flows simultaneously. Salgado et al. (1988) propose an extension of the gradient method of hydraulic analysis for modelling regulating valves such as PRVs and CVs and other devices such as pressure sustaining valves (PSVs), pump on/off switches and closed pipes. The proposed algorithm is "physically based" and does not require the identification of a new loop structure when the mode of operation of the pressure regulating device is different to the assumed mode of operation. The gradient method is reported to have a number of other advantages compared to the traditional methods of hydraulic analysis. The hydraulic simulation model called EPANET (Rossman, 1994) developed by the U.S.A. Environmental Protection Agency uses the gradient method of hydraulic analysis.

2.10.10 Extended period simulation (EPS)

A steady-state hydraulic analysis determines instantaneous flow and pressure distributions for a system subject to an instantaneous demand pattern. An extended period simulation (EPS) predicts system behaviour (such as node pressure patterns and tank water levels) over a period of time, given changes in system operation (such as pump schedules and valve status), and subject to varying demand patterns with time. The time-dependent simulations are a series of steady-state hydraulic simulations carried out at some specified time step.

To develop a practical tool for pipe network optimisation, it is necessary to link the solution evaluation scheme with a hydraulic simulation model that can perform an efficient EPS. EPS provides a more comprehensive assessment of the feasibility of the integrated water supply system design and operation, subject to the anticipated maximum and/or average demand cycle (the demand cycle is usually 24 hours for an urban water distribution system). EPS is the most effective way to evaluate elements of system design such as proposed system storage capacity and system operating rules such as pump schedules and EPS is used to approximate power costs for pumping.

Storage tanks (or balancing tanks) are used to help 'equalise' or smooth peak water demand periods and store water for emergency purposes. Tank water surface levels are assumed to be constant for a steady-state hydraulic analysis. The water levels are updated during the EPS by

2 Development of a hydraulic simulation model

accounting for flows in and out of the tanks. The shape (eg., cylindrical, spherical) and volume of the tanks, and the maximum and minimum water levels and the starting water level (at the initiation of the EPS) should be known for the EPS.

The dimensions of the tanks are required to determine how the water levels will vary with incoming or outgoing volumes of water. The tank water levels should not exceed the maximum water level or fall below a specified minimum water level. The minimum water level is not usually the very bottom of the tank since some storage is usually retained as an emergency supply. As a tank water level reaches the maximum water level or falls to the minimum water level, the pipes connecting the supply to the network are closed and remain closed until flow is reversed such that the flow is out of a full tank or into an empty tank. Ideally, water levels should return to close to the initial water levels by the end of the demand cycle (in preparation for the next demand cycle) and fluctuating water levels are often required to demonstrate a reasonable amount of tank exercising (for water quality reasons).

The accuracy of the EPS to simulate time-dependent behaviour depends on the time step between steady-state hydraulic analyses, the magnitude of the flows leaving and entering the tanks and the size and shape of the tanks (Lansey and Mays, 1989c). The time step is the time elapsed since the last steady-state hydraulic analysis. The time step is very important since the accuracy of the EPS increases as the time between hydraulic analyses decreases and the cost of additional hydraulic analyses is a rapid increase in computational effort.

The variations in water level Δh in tanks and reservoirs between steady-state hydraulic analyses for a time step Δt is expressed by Eq. 2.63. The water levels are updated each time step. The time taken Δt to fill or drain a tank a height Δh is computed by rearranging Eq. 2.63. If the water level is approximately constant for the source for the duration of the EPS, then the surface area is assumed to be infinite.

$$\Delta h = \frac{\Delta t \cdot Q}{A} \quad (2.63)$$

in which Q = flow to or from the tank or reservoir and A = water surface area of the tank or reservoir.

A new steady-state hydraulic analysis is required during the EPS if:

- a tank becomes full or empty,
- a tank fills or drains by more than some specified height increment,
The change (rise or fall) in water level between hydraulic analyses should not exceed some specified change in water level regardless of the time since the last hydraulic analysis.
- a specified time increment elapses,
The time between hydraulic analyses is not allowed to exceed some specified time interval (the time step).
- the demands vary (the start of a new demand period), or
- system operation changes (for example, a pump is switched on or off, or the setting or status of a valve is adjusted).

2.11 Integration of the Hydraulic Simulation Model and the Genetic Algorithm Pipe Network Optimisation Model

In this chapter, a hydraulic simulation model is developed which is to be integrated with the genetic algorithm (GA) model for pipe network optimisation. The simulation model uses the Newton-Raphson numerical solution technique (Martin and Peters, 1963) applied to the loop corrective flow equations (ΔQ_L -equations) and is similar to the algorithm outlined by Epp and Fowler (1970). The simulation model is developed specifically for the purpose of the GA optimisation. The computer simulated GA evolution is expected to generate many thousands of proposed pipe network designs. The hydraulic feasibility of each trial design is evaluated, and this may involve multiple hydraulic simulations (instantaneous and time-dependent) of the many thousands of alternative configurations of system components and operational decisions for system components.

The efficiency of the hydraulic simulation model is critical. The extensive investigation of methods for hydraulic analysis carried out by Wood and Rayes (1981) showed that both the linear method applied to the pipe flow equations (Q -equations) and the Newton-Raphson method applied to the loop corrective flow equations exhibit efficient convergence to accurate solutions. The hydraulic solution methods which formulated the pipe network equations as node equations (H -equations) were found to be less reliable.

The reliability of the method of hydraulic analysis is also important. The simulation model will be expected to model many different hydraulic situations including complex arrangements of system components and extensively looped pipe networks. The random processes of the GA search (including the randomly generated starting population and the randomness of crossover

2 Development of a hydraulic simulation model

and mutation) are expected to occasionally yield unconventional pipe network designs, many of which will be infeasible combinations of network components.

Based on experience to date, the simulation model developed in this chapter obtains accurate solutions for the case study pipe networks considered in this thesis. An additional check for imbalances in the loop energy equations may be used to confirm the accuracy of the analysis. The accuracy of the simulation model was verified by comparison with the established KYPIPE hydraulic simulation package (Wood, 1974) for the simulation of the New York City tunnels network in Chapter 8. The HGL at the critical (extreme downstream) nodes determined by the hydraulic simulation model developed in this chapter and the KYPIPE model are compared, and closely agree, for all the near-optimal designs identified by the GA and for designs identified by past studies of the classic New York tunnels problem. In general, final GA designs are independently verified by comparison with a commercial hydraulic simulation package.

Algorithms have been developed to accompany the chosen method of hydraulic analysis (the Newton-Raphson method applied to the loop equations), including algorithms to define the paths of pipes in natural loops and pseudo loops (the loop structure), an algorithm to determine a set of initial flows in the network such that continuity is satisfied and a loop numbering scheme that leads to a Jacobian matrix of coefficients of near minimum bandwidth. Sparse matrix routines are employed to efficiently reduce the identical, sparse Jacobian matrices at each iteration of the Newton-Raphson method. Row scaling and a strategy for selecting a suitable sequence of pivot elements maintain numerical stability and minimise the number of row reduction operations by limiting the amount of fill-in. The use of a convergence test specifies the accuracy of the solutions required and avoids unnecessary iterations of the method.

Various time-saving procedures can be implemented to speed up the hydraulic analyses performed as part of the GA search. For many pipe network optimisation problems, sequences of hydraulic analyses are performed for which the structure of the loop set does not change from one simulation to the next. This is the case for the optimisation of pipe sizes in a pipe network of fixed layout (with multiple reservoirs), subject to instantaneous demand pattern(s), such as the two-reservoir Gessler network case study introduced in Chapter 5. If the loop structure does not change in consecutive analyses, it is not necessary repeat computations to determine the following:

- the natural loops and pseudo loops (consequently an appropriate loop numbering scheme, Jacobian zero-nonzero structure and near minimum bandwidth is known).
- a practical and efficient order for the pivots computed by the modified Markowitz threshold pivoting. The sparse Jacobian matrices will have an identical zero-nonzero structure and contain elements of comparable scale and may be efficiently reduced from

2 Development of a hydraulic simulation model

one iteration of the Newton-Raphson technique to the next, and from one GA coded string evaluation to the next using the same pivot order. A sequence of pivot elements to reduce the sparse matrices by Gauss-Jordan elimination is established and stored to be applied to all the identical matrices encountered.

- good initial assumptions of the flow patterns for the same demand conditions (associated with initial assumptions of flow corrections equal to zero).

If loop structure is identical for consecutive analyses using the loop equations, the balanced flows of the previous analysis may be used as the initial flows for the new analysis to improve efficiency of the simulation. For an extended period simulation (EPS), the final flows of the last analysis may be used as the initial flows for the next analysis for successive analyses (for the same loop structure). Shamir and Howard (1968) highlighted the advantage of good initial guesses and recommended this type of approach to significantly reduce the amount of computation required. This practice is of value for any sequence of hydraulic analyses and would be of exceptional value in the GA search as the population of solutions converges and the pipe network designs become similar.

The structure of the loop set may be modified if:

- the layout of the new pipe network is being optimised (in this case, possible new pipes may be assigned a zero diameter),
- tank or valve locations are being optimised,
- the status of pipes, pumps or valves changes,
- the mode of operation of a pressure or flow regulating valve is different to the assumed mode of operation or changes during an extended period simulation, or
- tanks become full or empty during an extended period simulation.

If the loop structure does change for consecutive simulations it may be possible to store a data bank containing information about alternative loop structures (such as the loop numbering scheme, the sequence of row reduction operations and good initial flow assumptions). The data bank may be updated for new loop structures and accessed if previously identified loop structures are encountered to avoid repetitive computations. The number of alternative loop sets may depend on the number of pipes that may be added or removed, the number of existing tanks and valves, or the number of possible locations for new tanks and new valves.

In addition, during the GA search, the old population of coded strings may operate as a temporary data bank of evaluated solutions. A child coded string solution produced for the new population is compared to parent strings in the old population for a match which may avoid the repeated evaluation of strings.

2 Development of a hydraulic simulation model

The function of pressure reducing valves (PRVs) and check valves (CVs) and the extension of the Newton-Raphson method applied to the loop equations for the consideration of these system components is briefly discussed. In Chapter 9, the GA search is applied to the optimisation of proposed expansions to the complex Fort Collins - Loveland water distribution system. The Fort Collins - Loveland system is separated into five pressure zones and contains $NP = 323$ pipes, $NJ = 253$ junction nodes, $NF = 10$ fixed-grade nodes and 13 PRVs. Therefore, the system is modelled as $NL (=NP-NJ-(NF-1)) = 61$ natural loops and $(NF-1) = 9$ pseudo loops assuming the PRVs are in the inoperative mode and 48 natural loops and 22 pseudo loops for 13 operating PRVs. It is a more difficult problem to analyse systems separated into multiple pressure zones by multiple pressure regulating devices where the mode of operation of the devices is not initially known and is identified as part of the simulation (Salgado et al., 1988). For this reason, the code of an established simulation model is embedded within the GA model evaluation scheme to apply the GA search to the optimisation of pipe diameters and PRV pressure settings for the Fort Collins - Loveland system. The code of the established simulation model is modified to some extent to suit the purposes of the GA search.

Extended period simulation (EPS) plays an important role in integrated water supply system optimisation and techniques are suggested for the extension of the steady-state simulation model to consider EPS. Time-dependent simulations are not performed for the case study pipe network optimisation problems considered in this thesis.

When this research commenced, computation times to switch between separate optimisation and simulation packages via an external link were excessive. In future studies, as computers become faster and hydraulic simulation becomes more complicated, it may be more practical to link the GA model with an established simulation model such as EPANET (Rossman, 1994). Reliable models for hydraulic simulation such as EPANET are becoming available which are better equipped to handle complexities such as multiple pressure and flow regulating devices. To link the GA model with an existing simulation model, familiarity with the simulation model is essential, to know how to obtain the results of the simulation in order to evaluate hydraulic feasibility and to increase efficiency of the simulation by the avoiding repetitive computations. It is also important to recognise when the simulation model has not converged to an accurate hydraulic solution. Pipe network designs for which a hydraulic solution cannot be obtained should be penalised in the GA search, however, a breakdown of the simulation model should not cause the GA to stop processing. The GA can afford to discard these designs since it operates with a population of designs at any time. These designs are not expected to be near-optimal designs. Near optimal solutions are expected to be realistic design situations, for which the pipe network simulation model will converge to a hydraulic solution.

2 Development of a hydraulic simulation model

Nevertheless, there were a number of advantages in developing a hydraulic simulation model from the ground up for the purposes of the development of the GA model. Firstly, this afforded a flexibility to model assumptions and peculiarities of system operation and simulation encountered in practice, and consider different system performance requirements and reliability measures used to evaluate alternative designs in practice. Secondly, special techniques could be used to improve the efficiency of the simulation. Finally, the simulation model could be modified to perform other tasks such as individual coded string solution evaluations, checks for local optimality and also the exhaustive enumeration of the two-reservoir Gessler network expansions problem in Chapter 5.

Hydraulic simulation is the prediction of system performance for a specified system design and operation, subject to some set of demand conditions. Simulation has a unique solution, which is obtained in the most efficient and accurate manner, although it is not a simple task to obtain this unique solution. Simulation is a prerequisite step to the more difficult task of optimisation. Optimisation is the determination of the most economical system design and operation, for a specified level of system performance for the system subject to one or more demand conditions. Optimisation has a unique solution, although it is much more difficult to identify. Given a pipe network design problem, ten water supply system designers may determine ten different designs and ten different pipe network optimisation techniques may identify ten different solutions. A hydraulic simulation model has been prepared in this chapter, to be coupled with a GA pipe network optimisation model. In the next chapter, the pipe network optimisation problem is introduced and a number of pipe network optimisation techniques from the literature are reviewed.

3 Optimisation of Water Distribution System Design

Water transmission and distribution systems are fundamental components of the infrastructure of a community. Water systems provide for a diversity of water needs. The demand for water varies with population and economic growth. Water system expansions and alternative system operating rules and settings are considered to satisfy changing water needs.

The design of a water distribution system can be separated into the *simulation* problem and the *optimisation* problem. Hydraulic *simulation* predicts system behaviour including instantaneous (steady-state) and time-dependent flow and pressure distributions and tank water level variations for some proposed water distribution system design. A series of manual hydraulic simulations of system designs can provide a workable design, however simulation models alone do not have the capability to determine optimum designs.

A hydraulic simulation model may be coupled with an optimisation model. The simulation model developed in Chapter 2 is linked to the genetic algorithm search in this research. The *optimisation* model determines water distribution system design (and operation), for a specified level of system performance, and such that system expansion costs (and operating costs) are a minimum. The optimisation problem is introduced in this chapter. A number of optimisation techniques have been developed to search for the optimal water distribution system design and some of these are reviewed in this chapter.

3.1 The Optimisation Problem

The optimisation of a water distribution system design is a process which strives to determine the minimum cost system design, which supplies the projected demands while maintaining an acceptable level of system performance. In general, the optimisation problem can be stated as:

Minimise system construction / expansion and operating costs

subject to system performance constraints
 hydraulic constraints
 general design constraints

In general, the objective function to be minimised and the constraints to be satisfied are a set of nonlinear functions in terms of the decision variables.

3.1.1 The decision variables

The decision variables are the physical and operational characteristics of the system components to be defined in the design. The decision variables represent design information concerning the layout, sizing and operation of system components such as pipes, tanks, sources of supply, pumping facilities and valves. The design may be an expansion or rehabilitation of an existing system, in which case the optimisation should evaluate the use of existing system components.

The design of the water distribution system may require the introduction of new pipes. The pipe network is the principal framework of the water distribution system composed of a set of nodes interconnected by a layout of pipes. The nodes represent junctions of pipes and may have associated demands and some nodes are sources of water. The pipes may be arranged in a tree network or in loops. A looped network may be preferred (particularly in urban systems) as loops can help meet downstream demands in the event of pipe breakage or pipe maintenance and loops provide greater flexibility to meet abnormal demands such as fire fighting loads. Pipe network reliability considerations may require that each demand node is connected to a source node by at least two paths.

The optimisation of the pipe network layout considers the placement of the possible new pipelines as decision variables. The proposed pipe network layout is often constrained by the street layout or by an existing layout of pipes. The new pipes should be constructed along accessible routes. Many pipe network optimisation techniques size new and duplicate pipes (pipes parallel to existing pipes, but may be of different diameter) for a fixed pipe network layout (minimum pipe diameters are specified for new pipes). A solution to the pipe network layout problem may be found by allowing new or duplicate pipes the option of a zero pipe diameter in the pipe sizing problem. Of course, demand nodes may not be disconnected from the network.

In its simplest form, the pipe network optimisation problem is the sizing of pipes of a predefined gravity-fed pipe network layout for one critical instantaneous demand pattern. The optimum pipe network layout for one demand pattern is a tree network (Swamee and Khanna, 1974; Quindry et al., 1981). Network redundancy (or reliability) often results from analysing multiple demand patterns.

The diameter, material and wall thickness (or pressure class) of the new pipes are decision variables which may be selected in the optimisation. The choices of pipe diameters are the discrete set of commercially available pipe sizes. Some pipe network optimisation techniques determine a *continuous diameter design* in which the optimised set of pipe diameters take on

3 Optimisation of water distribution system design

continuous real values. A *discrete diameter design* is a set of pipe diameters selected from among a specified set of commercially available pipe sizes. A *split pipe design* is derived from a continuous diameter design by decomposing a pipe section length of continuous diameter into partial lengths of discrete diameters to form a pipe with equivalent hydraulic properties. Internal pipe diameters should be used for the hydraulic simulations, since internal diameters may differ significantly from nominal diameters.

The old pipes in the system may be candidates for cleaning and/or relining, or duplication with a new parallel pipe in the optimisation problem. Alternatively, the existing pipes may be removed from the pipe network or retained without improvement. The deteriorating condition of existing system components such as increasing pipe roughnesses (and decreasing pump efficiencies) should be considered for the projected demand periods. The condition of the pipes inside wall is influenced by the pipe material, time in operation and the water quality. The cleaning and relining of existing pipes to improve the roughness of the pipe inside wall may be an economical design consideration.

The optimisation may select the site, shape, volume and elevation of new storage tanks. The optimisation observes the variation of the tank water levels for extended period demand patterns to ensure balancing tanks fill during low demand periods and empty when demands are high. Loubser and Gessler (1990) indicated that the pipe sizing problem and the tank sizing problem are interrelated. The tank volume should be adequate for balancing peak demands and the feeder pipes connected to the tank should have adequate capacity for refilling the tank during off-peak demand periods. The tanks should show a net inflow for average demand flows.

The pipe distribution network may be a gravity-fed system or a pumped system. The optimisation of a water distribution system with source pump stations or in-line booster pump stations may consider the installation of new pump stations or the upgrade of existing pump stations. The decision variables may include the location of new pump stations and the capacity and arrangement of individual pump units.

The pump facility operation schedule is the set of operating rules indicating when the individual pumps should be switched on and off over a specified period of time (the demand period cycle) such as a day for an urban system. The pump operating rules for a cycle of average demands may be decision variables so that average annual energy costs for pumping can be computed. The pump operation schedule for a cycle of maximum demands may also be decision variables to ensure tanks do not empty.

3 Optimisation of water distribution system design

Ormsbee et al. (1989) optimised the operation of water supply pumps for the minimisation of pump operation costs. The two phase technique may be applied to water distribution systems with a single dominant source and multiple pump stations. The first phase of the optimisation develops an optimal tank water level trajectory determined by dynamic programming. The second phase of the optimisation uses an enumeration scheme to develop an optimal pump operating policy to achieve the optimal tank trajectory. Tarquin and Dowdy (1989) and Jowitt and Germanopoulous (1992) considered the optimal pump scheduling problem for water distribution systems.

Devices such as pressure reducing valves, pressure sustaining valves or flow control valves are used in water distribution systems to regulate flows and/or pressures by adjusting settings to vary energy losses. Pressure reducing valves (PRVs) control the interface between pressure zones or subdue large pressure variations in adjacent systems. The optimisation should evaluate the use of existing PRVs and any proposed new PRVs. The location and pressure settings for proposed PRVs may be decision variables in the optimisation and existing PRVs may be relocated or removed. The pressure setting selected for a given PRV site is an indication to whether or not a PRV is required at the site. The operational mode of the PRV is the only economical mode since the system component is functioning (El-Bahrawy and Smith, 1987). The use of PRVs is not appropriate for some systems. The possible sites and settings of other valves and control devices (such as flow control valves, altitude control valves and orifice plates) may be considered in the optimisation in a similar way.

Perez et al. (1993) describe a procedure for the efficient utilisation of PRVs in tree networks. The procedure ensures the flows are delivered to the demand points above some minimum pressure and the pipe wall thickness class is selected to withstand internal pipe pressure. PRVs may be used to keep pressure within operating limits to protect equipment such as thinner, more economical pipes from excessive pressures and to reduce the danger of leaks.

In summary, the optimisation problem may consider some combination of the possible decision variables:

- Layout of components*
- introduce new pipes
 - remove or maintain existing pipes
 - parallel existing pipe routes with new pipes (duplicate pipes)
 - identify sites for new storage tanks
 - locate pump station facilities
 - locate valves and other control devices (consider pressure zone layouts)
 - identify future sources of supply (wells, connections to adjacent systems)

3 Optimisation of water distribution system design

- Sizing of components*
- select the size, material and class of new pipes and duplicate pipes
 - the capacity and number of pump units for new / upgraded pump stations
 - the shape, volume and elevation of new tanks
 - determine the capacity of other system facilities such as regulating devices and sources (often limited by the capacity of water treatment facilities or the amount of water that may be drawn from a connecting system)
- Rehabilitation*
- consider the cleaning / relining or replacing of existing pipes
 - consider the refurbishment of existing pump stations and the upgrade of other existing facilities (such as the redrilling of wells)
- Operation of components*
- select the pump station operating rules (the number and combination of individual pumps in operation)
 - determine normal operating water surface levels for tanks
 - select valve settings and controls (operating rules may be selected for some regulating devices such as a schedule of flow settings for flow control valves)

3.1.2 The objective function

The objective of the optimisation is typically to minimise the capital costs of new or upgraded system components plus the present value of system operating costs for the expected lifetime of the design. The cost of new pipes may include the cost of the pipe material, transport and laying costs and existing pipes may be considered for duplication or cleaning/relining. The costs of installing pipe may vary with pipe route (eg., dirt or paved roads, volume of traffic). The costs for new or upgraded pump station installations and tank construction (and land acquisition costs) may be considered where they are factors. The system operating costs may include energy costs for pumping, water treatment costs and perhaps the cost of purchasing water from an adjacent systems.

3.1.3 System performance constraints

The optimal water distribution system design should adequately satisfy the water requirements for a complex combination of water users. Several possible demand patterns may need to be investigated to ensure a reliable design:

- instantaneous peak hour demands
- emergency demand patterns such as fire fighting loadings

3 *Optimisation of water distribution system design*

- emergency conditions where a system component is out of operation such as pipe breakage, pipe maintenance or pump failure (to ensure reliability)
- the variation of demands on the maximum day (or appropriate critical period)
- the variation of demands on the average day (to estimate energy costs for pumping)

The design must demonstrate a specified level of system performance for the demand patterns considered. The system performance constraints are usually set by the managing water utility or company. The demand patterns and corresponding system performance constraints applied to a water distribution system depend on factors such as:

- consumer water use (eg. domestic, industrial or agricultural)
- system reliability to be achieved (eg. considering a peak demand pattern with the reservoirs and tanks at their low levels and one source pump unit out of operation)
- the probability of a demand situation occurring

The most important system performance constraints are the minimum acceptable pressures at the demand points for all demand conditions. System performance and reliability requirements may include:

- minimum and maximum allowable pressures at demand nodes
- pipe velocities within acceptable limits
- pumps should operate within acceptable operating limits for all demand situations
- minimum and maximum operating water levels for reservoirs and storage tanks
- the water levels at the end of a critical period (eg. maximum day) should be at least equal to the water levels at the start of the period (in preparation for the next period)
- fluctuating water levels should demonstrate a reasonable amount of tank exercising (for water quality reasons)
- the inflows from some sources may be restricted (eg. inflows from adjacent systems and inflows restricted by the capacity of a water treatment plant)
- the available hydraulic grade at some points may be limited (eg. transmission system turnouts or connections to adjacent zones or systems), and the system may be required to maintain a minimum grade at points which serve adjacent zones or systems

The water should be supplied to the consumer with an operational pressure head. Maximum pressure head constraints protect against exceeding the strength of the pipe material as well as consumer appliances. Velocity of the flow may be restricted to a permissible range since scouring of residue on the pipe wall may lead to poor water quality.

3.1.4 Hydraulic constraints

For a water distribution system subject to a given demand pattern, the physical laws of continuity of flow at the nodes and conservation of energy around the loops must be satisfied. A friction head loss formula (eg. Hazen-Williams, Darcy-Weisbach) describes energy loss as a function of flow in the pipes. Pump characteristics describe pump energy gain as a function of pump flow. Some optimisation models (eg. linear programming and nonlinear programming techniques¹) incorporate these nonlinear hydraulic constraints in the optimisation problem formulation while other models (eg. enumeration, genetic algorithms) are linked to a hydraulic simulation model (Chapter 2) which incorporate the hydraulic constraints.

3.1.5 General design constraints

There may be general design constraints which should be observed. The available sizes of new system components are usually discrete sets (corresponding to commercially available sizes). There may be minimum diameter constraints to ensure a looped network of pipes is maintained. Some specific design constraints may apply to the system being studied. For example, some existing pipe routes may not be accessible for duplication.

3.1.6 The solution space

A set of l values for the decision variables represents a network design solution and a point in an l -dimensional solution space. Solutions that satisfy the constraints are *feasible designs* and those that do not are *infeasible designs*. Feasible designs are restricted to a subspace of the solution space called the feasible region and the scope of the feasible region is shaped by the constraints.

The objective function value and the feasibility of a design are determined by performing a function evaluation. The objective function may have many local minima in the feasible region. An optimisation procedure searches for the global minimum or smallest local minimum in the feasible region. The optimum solution is likely to occur on or near the boundary of the feasible region.

¹ Some linear programming (Morgan and Goulter, 1985) and nonlinear programming (Lansley and Mays, 1989a) techniques are coupled with hydraulic simulation models.

3.2 The Benefits of Optimisation

Current design practices usually involve costing and simulating a handful of designs selected by the designer using engineering judgement and rules of thumb. The values for each of the decision variables are modified by a trial and error approach and a series of hydraulic simulations are performed until a feasible combination of decision variable values is determined.

Mathematical optimisation is a powerful tool for guidance in the design process, although it must always be supplemented by engineering experience. Optimisation eliminates the need for the uneconomical trial and error design practices and allows the designer to concentrate on other aspects of the design such as considering alternative pipe network layouts.

Optimisation can provide a number of near-optimal designs which may be evaluated by the designer in terms of other (perhaps non-quantifiable) engineering objectives such as expected future developments or environmental issues. The interpretation of the designs generated by an optimisation search lends insight into characteristics and peculiarities of the water system such as the limitations of system facilities, the critical demand conditions and system performance requirements, and the most limiting design constraints. The optimisation may suggest alternative design options and explore alternative system operating rules.

Water authorities are aware that water distribution system design is a complex optimisation problem. Many millions of dollars are spent providing a water supply system, and an optimisation search should be employed to explore potential cost savings. Even for a relatively small pipe network design, there may be millions of alternative designs. Experienced designers can often design a pipe network to be within 10% to 20% of the global optimum solution, however, better designs may not be obvious and may be overlooked. In some cases, lower cost pipe networks are superior hydraulic designs.

Research has shown that the cost differences between apparently good designs and the global optimum design may be substantial. Schaake and Lai (1969) applied a linear programming method to the proposed expansions to the New York City water supply tunnels (21 tunnel sections) and determined an optimised design cost of \$78.1 million. Researchers using a range of optimisation tools have studied the pipe network since 1969 and costs have been progressively cut to \$39.2 million. The significant savings identified by successive optimised designs demonstrates the elusiveness of the best designs. The genetic algorithm optimisation procedure developed in this research is applied to the benchmark New York tunnels problem in Chapter 8.

3.3 Pipe Network Optimisation Techniques

A search for the optimal design for a water distribution system can be achieved by a number of optimisation techniques including:

- Nonlinear programming (NLP)
- Linear programming (LP)
- Two-phase decomposition methods
- Random search methods
- Dynamic programming
- Enumeration algorithms
- Heuristic techniques
- Equivalent pipe methods
- Evolutionary strategy
- Genetic algorithms

Many of the procedures developed are a hybrid scheme combining two or more of the above techniques. Walski (1985) and Lansey and Mays (1989b) present extensive reviews of optimisation models for water distribution design in the literature. A selection of optimisation methods are now reviewed.

3.3.1 Nonlinear programming (NLP)

El-Bahrawy and Smith (1987) presented the general nonlinear programming (NLP) formulation of the pipe network optimisation problem. El-Bahrawy and Smith employed the powerful, large-scale, nonlinear, sparse-oriented MINOS package (Murtagh and Saunders, 1980) to solve the water distribution system design optimisation problem. Other nonlinear programming software packages which are available include GINO (Liebman et al., 1986) and GAMS (Brooke et al., 1988).

For a pipe network with NP pipes, the pipe network optimisation problem requires the determination of $3NP$ unknowns: including the combination of the diameters D_j , the flows Q_j and the head losses ΔH_j in the pipe sections j . The head losses may be found from the flows directly using the friction head loss relationship applied to each pipe section. El-Bahrawy and Smith (1987) formulated the problem as a nonlinear objective function subject to linear and nonlinear constraints considering three design variables for each pipe. These are pipe diameter D , pipe flow Q and pump shut-off head H_0 (for pipes which contain pumps). The NLP model is formulated as shown:

3 Optimisation of water distribution system design

$$\text{Minimise cost } Z = \sum_{j=1}^{NP} f_1(D_j, Q_j, H_{0j}) \quad (3.1)$$

subject to

$$\sum_{j=1}^{NPL} f_2(D_j, Q_j, H_{0j}) = \Delta E_{mn} \quad \text{for loops } l=1, \dots, NL+(NF-1) \quad (3.2)$$

$$\sum_{j=1}^{NPR} f_3(D_j, Q_j, H_{0j}) \geq H_{min_i} \quad \text{for junction nodes } i=1, \dots, NJ \quad (3.3)$$

$$\sum_{j=1}^{NPJ} Q_j = Q_{ex_i} \quad \text{for junction nodes } i=1, \dots, NJ \quad (3.4)$$

$$D_{min} \leq D_j \leq D_{max} \quad \text{for pipes } j=1, \dots, NP \quad (3.5)$$

$$Q_{min} \leq Q_j \leq Q_{max} \quad \text{for pipes } j=1, \dots, NP \quad (3.6)$$

$$0.0 \leq H_{0j} \leq H_{0_{max}} \quad \text{for pipes } j=1, \dots, NP \quad (3.7)$$

in which

D_j = diameter of pipe j

Q_j = flow in pipe j

H_{0j} = the shut-off head for a pump in pipe j

$\Delta E_{mn} = E_m - E_n$ = elevation difference between fixed-grade nodes m and n in the case of a pseudo loop ($\Delta E_{mn} = 0$ for a natural loop) where m is connected to pipe $j=1$ and n is connected to pipe $j=NPL$

H_{min_i} = the minimum allowable pressure head at node i

Q_{ex_i} = demand at node i

NPL = the number of pipes j that make up natural loop or pseudo loop l

NPR = the number of pipes j in the path from a fixed-grade node to junction node i

NPJ = the number of pipes j connected to node i

NP = the number of pipes

NJ = the number of junction nodes

NL = the number of natural loops

NF = the number of fixed grade nodes

$(NF-1)$ = the number of pseudo loops

3 Optimisation of water distribution system design

The objective cost function f_1 in Eq. 3.1 is comprised of pipe cost as a function of pipe diameter and pump installation and operating costs as a function of pump power which is, in turn, a function of pump flow and pump head (pump head is determined from the pump characteristic curve given pump flow and pump shut-off head). The pipe cost function and pump installation and operating cost functions are defined with suitable parameters of pipe cost per unit length, capital cost of pumps per unit power, the cost of energy, the annual pumpage time, appropriate values of interest rate and the expected life of the pump. The objective function is subject to a set of linear and nonlinear constraints.

The set of nonlinear constraints in Eq. 3.2 are the loop energy conservation constraints (see Eq. 2.22). The equations state the net energy gain around a loop is zero. The nonlinear constraints in Eq. 3.3 are the minimum node pressure constraints. The constraints state the head losses in pipes between a fixed-grade node and node i should not result in a pressure less than the minimum allowable pressure for node i . The linear constraints given by Eq. 3.4 are the node flow continuity constraints (see Eq. 2.20). The equations state the volumetric flow rate into a junction node equals the volumetric flow rate out of the node. The constraints in Eqs. 3.5 to 3.7 are the sets of bounds on the decision variables. The functions f_2 and f_3 may consist of a combination of the pipe friction head loss, minor losses and pump lift expressions.

The MINOS package is designed to solve large-scale nonlinear programming systems involving sparse constraints. The nonlinear functions in the objective function and the constraints must be continuous and smooth. El-Bahrawy and Smith (1987) described how pipe network components such as pumps, reservoirs, pipe fittings, check valves and pressure reducing valves (PRVs) may be included in their model. A pre-processor was developed to automatically generate the MINOS input data file from the simple network data and a post-processor was used to interpret the output data provided by MINOS.

El-Bahrawy and Smith (1987) proposed an analysis to determine the optimal discrete diameter solution from the continuous diameter solution. They incorporated techniques which determine network layout by identifying the primary tree networks and then selecting loop forming redundant links by assessing network reliability.

Lansley and Mays (1989a) developed a methodology which uses a NLP procedure based on the generalised reduced gradient method (GRG2, Lasdon and Waren, 1983). The pipe network optimisation problem was stated in terms of a vector of nodal pressure heads H and the design decision variables D . The decision variables D define pipe diameters, pump sizes, valve settings and tank volumes or elevations

3 Optimisation of water distribution system design

The nonlinear objective cost function $f(\mathbf{H}, \mathbf{D})$ includes the cost of components such as pipes, pumps, valves and tanks. The objective function is to be minimised subject to the following constraints:

- Hydraulic constraints $\mathbf{G}(\mathbf{H}, \mathbf{D})$ based on the physical laws of node continuity of flow and loop conservation of energy. These constraints define the pressure distribution for one or more demand patterns.
- Pressure head bounds specify the minimum and maximum allowable pressure heads for the nodes in the system.
- Design constraints set by physical limitations or availability of components.
- General constraints such as minimum and maximum pipe velocities.

Existing NLP packages are usually limited in the number of constraints that can be handled. The number of hydraulic constraints \mathbf{G} becomes very large as pipe network size increases or if multiple demand patterns are to be considered. The NLP procedure of Lansey and Mays is coupled with a hydraulic simulation model (KYPIPE, Wood, 1974) as shown in Figure 3.1.

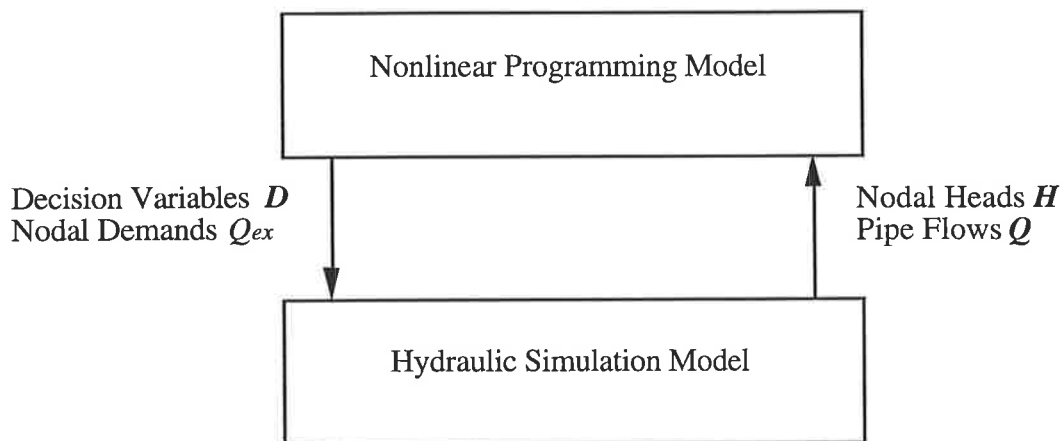


Figure 3.1 Coupled NLP model and simulation model

The simulation model determines the node pressure heads \mathbf{H} given the set of decision variables \mathbf{D} and therefore the hydraulic constraints \mathbf{G} may be removed from the optimisation problem.

The gradient (or reduced gradient) of the new objective cost function $f(\mathbf{H}(\mathbf{D}), \mathbf{D})$ with respect to \mathbf{D} determines how to alter the decision variables \mathbf{D} . The reduced gradients are computed by solving a system of linear equations for each demand pattern since they cannot be calculated directly. Since the pressure heads \mathbf{H} are implicit functions of the decision variables \mathbf{D} , the pressure head bounds are not considered when step size is determined and may be violated. Therefore, the pressure head bounds are incorporated into the objective cost function by using

3 Optimisation of water distribution system design

an augmented Lagrangian penalty function method. The penalty weights and Lagrangian multipliers in the penalty term are also optimised in the overall optimisation problem.

Duan et al. (1990) extended the work of Lansey and Mays (1989a). They developed a general optimisation model that can include pumps and tanks (and the locations of these) as well as multiple loading conditions. The model operates on a hierarchical basis as follows:

- At the master problem level, the numbers and locations of pumps and tanks are identified by implicit enumeration.
- At the subproblem level, the GRG technique is used to find the optimum pipe sizes for the pump and tank layout specified at the master problem level.
- An inner loop within the subproblem uses KYPIPE to ensure that the continuity and head loss constraints are satisfied, and a separate model (RAPS) is used to compute various measures of system reliability.

The Western Australian Water Authority (WAWA) has developed an in-house capability for pipe network optimisation using nonlinear programming (Waters, 1989; Vigus, 1989).

3.3.2 Linear programming (LP)

Schaake and Lai (1969) formulated the nonlinear pipe network optimisation problem as a linear programming (LP) problem. The node pressure head pattern is selected and then the LP model determines a set of optimum pipe diameters. The nonlinear expressions in terms of the pipe diameters D_j are linearised by introducing a new variable X_j using the following variable transformation:

$$X_j = D_j^p \quad \text{for pipes } j = 1, \dots, NP \quad (3.8)$$

The constant p equals 2.63 for the rearranged Hazen-Williams head loss formula (see Eqs. 2.4 and 2.5):

$$Q_j = \alpha C_j L_j^{-0.54} h_{f_j}^{0.54} D_j^{2.63} = \alpha C_j L_j^{-0.54} h_{f_j}^{0.54} X_j \quad (3.9)$$

in which α is a constant which depends on the units used.

3 Optimisation of water distribution system design

The linear programming model was formulated as shown:

$$\text{Minimise cost} \quad Z = \sum_{j=1}^{NP} \beta_j L_j X_j \quad (3.10)$$

subject to

$$\sum_{j=1}^{NPJ} \alpha C_j L_j^{-0.54} h_{fj}^{0.54} X_j = Q_{exi} \quad \text{for junction nodes } i=1, \dots, NJ \quad (3.11)$$

$$X_j \geq 0.0 \quad \text{for pipes } j=1, \dots, NP \quad (3.12)$$

Linear constraints of the type given in Eq. 3.11 for the LP model are based on the mass continuity equations for the junction nodes. The NJ binding linear constraints are obtained by substituting Eq. 3.9 into the node continuity relationship given in Eq. 2.20. The constraints in Eq. 3.11 assume flow in pipe j is towards node i . The pipe lengths L_j , and roughnesses C_j and node demands Q_{exi} are assumed to be known. The node pressure heads are assumed and the friction head loss h_{fj} in pipe j is the difference in hydraulic grade between the start and end nodes of the pipe. The constraints in Eq. 3.11 are linear in the X_j variables for an assumed pressure pattern. The NP inequalities in Eq. 3.12 state the diameters must be non-negative.

The objective function in Eq. 3.10 is some linear function of the X_j variables. A linear pipe cost function is approximated using available pipe cost data in which β_j is the cost per unit length of pipe j (of pipe size D_j) per unit of X_j . It may be necessary to approximate the cost function by a set of linear segments. Schaake and Lai (1969) also considered multiple demand patterns and the inclusion of pumping costs in the objective function (not shown here).

The single step LP model developed by Schaake and Lai (1969) is limited in that it does not compute the optimum operating pressure heads and the quality of solution is dependent on the initially assumed pressure distribution. Schaake and Lai recommended using the LP model for different pressure distributions. The pressure distribution is thought to be near-optimal if small changes in the pressure distribution have little effect on the cost.

Quindry et al. (1981) used the LP model developed by Schaake and Lai (1969) as an iterative step in a gradient search technique. The dual variables λ_i generated by the LP model associated with each constraint in Eq. 3.11 are given by:

$$\lambda_i = \frac{d(\text{cost})}{d(Q_{exi})} \quad \text{for nodes } i=1, \dots, NJ \quad (3.13)$$

3 Optimisation of water distribution system design

Gradient terms which adjust the pressure distribution are given by Eq. 3.14 and are calculated for each node in the pipe network given the current values of nodal pressure heads, pipe flows and the dual variables.

$$\frac{\partial(\text{cost})}{\partial H_i} \quad \text{for nodes } i=1, \dots, NJ \quad (3.14)$$

The adjusted pressure head distribution becomes the starting assumption for the next iteration. Iterations continue until the rate of improvement in cost is small.

Alperovits and Shamir (1977) presented an alternative LP gradient approach. A list of candidate discrete diameters are chosen for each proposed new pipe in the network. The new pipe is assumed to consist of a set of segments of constant diameter. The unknowns in the LP formulation are the partial lengths X_{jm} of pipe j of the m th diameter².

Alperovits and Shamir (1977) assumed an initial flow pattern in the looped pipe network which satisfied continuity at the nodes. If the flows in the pipes are known, the friction head loss hf_{jm} in pipe segment m of pipe j is a linear function of the unknown partial length X_{jm} . The LP model is formulated as shown:

$$\text{Minimise cost } Z = \sum_{j=1}^{NP} \sum_{m=1}^M c_{jm} X_{jm} \quad (3.15)$$

subject to

$$H_{\min_i} \leq H_s \pm \sum_{j=1}^{NPR} \sum_{m=1}^M S_{jm} X_{jm} \leq H_{\max_i} \quad \text{for junction nodes } i=1, \dots, NJ \quad (3.16)$$

$$\sum_{m=1}^M X_{jm} = L_j \quad \text{for pipes } j=1, \dots, NP \quad (3.17)$$

$$\sum_{j=1}^{NPL} \sum_{m=1}^M S_{jm} X_{jm} = \Delta E_{ln} \quad \text{for loops } l=1, \dots, NL+(NF-1) \quad (3.18)$$

$$X_{jm} \geq 0.0 \quad \text{for pipes } j=1, \dots, NP \quad (3.19)$$

² Calhoun (1971) developed a similar LP formulation for tree pipe networks which assumed the total length of new pipe consists of partial lengths of constant diameter. The flow pattern in a tree network is known since it is governed only by continuity.

3 Optimisation of water distribution system design

in which

X_{jm} = the length of pipe segment of the m th diameter in pipe j

c_{jm} = the material cost per unit length of pipe of the m th diameter in pipe j

M = number of candidate discrete pipe sizes for pipe j (number of segments m in pipe j)

H_{min_i} = minimum allowable pressure head at node i

H_{max_i} = maximum allowable pressure head at node i

H_s = pressure head at the source node s

S_{jm} = the hydraulic gradient of segment of m th diameter of pipe j

L_j = the total length of pipe j

The objective function in Eq. 3.15 assumes a linear relationship between the cost of the pipe and its length. The objective function is subject to the set of linear constraints in Eqs. 3.16 to 3.19. The linear constraints in Eq. 3.16 check that the pressures at the nodes are within the allowable bounds. The hydraulic gradient is determined from the Hazen-Williams head loss formula. The hydraulic gradient S_{jm} is linear in X_{jm} if a flow Q_j in pipe j is known or assumed:

$$S_{jm} = \frac{h_{f_{jm}}}{L_{jm}} = R_{jm} Q_j^{1.852} \quad (3.20)$$

The constraints in Eq. 3.17 ensure the total lengths of the segments m equals the length of pipe j . The constraints in Eq. 3.18 are based on the law of conservation of energy around a natural loop or pseudo loop l and they ensure the segments are chosen such that the network is hydraulically balanced. Finally, non-negativity constraints in Eq. 3.19 are required for the variables X_{jm} .

The linear programming model determines the optimal set of decision variables for an assumed flow pattern. The dual variables λ_l generated by the LP model associated with each constraint in Eq. 3.18 are given by:

$$\lambda_l = \frac{d(cost)}{d(\Delta E_l)} \quad \text{for loops } l=1, \dots, NL+(NF-1) \quad (3.21)$$

The dual variables in Eq. 3.21 are used to find a gradient move in the steepest direction in the solution space. The assumed flow pattern is adjusted in this direction. The gradient terms are given by:

$$\frac{\partial(cost)}{\partial \Delta Q_l} \quad \text{for loops } l=1, \dots, NL+(NF-1) \quad (3.22)$$

3 *Optimisation of water distribution system design*

The changes in flows ΔQ_l in the loop l are made proportional to the corresponding components of the gradient terms. The step size of the move is based on the magnitude of the change in flow but may be varied depending on the success of the previous steps. The linear program optimises the design and hydraulically balances the network for each adjusted flow pattern. The adjusted flow distribution becomes the starting assumption for the next iteration. Iterations continue until no further improvement in cost can be achieved.

The list of candidate discrete diameters are selected for each new pipe from commercially available pipe sizes. The candidate diameters for a pipe should cover a range of diameters such that the limits of the list are not a constraint in the optimal design. Minimum diameters may be specified for some pipes for system reliability.

In the optimal design, it may be necessary to round split pipe sizes to a single pipe size for the entire length of pipe. Small pipe length segments should be removed. The final rounded design should be checked to ensure it satisfies the minimum and maximum pressure head constraints. The design may no longer be optimal following the rounding procedure.

Alperovits and Shamir (1977) provided an extension of their model to account for multiple demand patterns, pumps, valves and reservoirs and operational considerations. The model can design a new system or additions to an existing system.

Morgan and Goulter (1985) used a LP model coupled with a simulation model (based on the Hardy Cross method) and heuristics to determine the optimal layout and design of looped water distribution systems. The decision variables for the LP model are partial lengths of a pipe of a given diameter to be replaced by different diameters. The LP model modifies the pipe sizes. The simulation model determines flow and pressure distributions for the modified system layout and design. A heuristic weighting procedure is used to remove uneconomical pipe locations. The LP model again modifies the pipe sizes given the flow distribution and the weights. Iterations of this procedure continue until the pipe sizes are not changed by the LP model.

Schaake and Lai (1969), Quindry et al. (1981) and Morgan and Goulter (1985) applied their LP models to the planned expansions of the primary water tunnels of New York City and the results are shown in Chapter 8. Some LP packages include LINDO (Schrage, 1981), LINGO and GAMS (Brooke et al., 1988).

3.3.3 Two-phase decomposition methods

The pipe network optimisation problem is generally expressed as a nonlinear programming problem (El-Bahrawy and Smith, 1987). Decomposition methods such as the LP gradient methods presented by Alperovits and Shamir (1977) and Quindry et al. (1981) are those that decompose the general NLP formulation of the pipe network optimisation problem into smaller subproblems and solve them in an iterative fashion.

Pipe network optimisation problems generate solution spaces with feasible regions that are nonconvex and have multiple local optimal solutions (Loganathan et al., 1995). The LP gradient methods are limited in that a local search is performed in the vicinity of the assumed starting values of node pressures or pipe flows. The NLP method is also a local search which alone, may only attain a local minimum in the vicinity of the assumed starting point. It is recommended the LP and NLP problems be solved several times starting with a number of initial solutions. The two-phase decomposition methods and other methods which combine global strategy with local refinement attempt to generate better local optimal solutions.

Fujiwara and Khang (1990) proposed a two-phase decomposition method that combined the methods of Alperovits and Shamir (1977), Quindry et al. (1981), and Mahjoub (1983). In the first phase, a NLP gradient method (an extension of the LP gradient method of Alperovits and Shamir, 1977) finds the optimum pipe head losses (and hence the pipe diameters) for an assumed flow distribution and pumping heads. A correction is then applied to the assumed flows and pumping heads using the Lagrange multipliers associated with the previous solution. This iterative process converges to a local optimum. In the second phase, the head losses obtained at the end of the first phase are fixed. A second NLP model finds the optimum pipe flows and pumping heads for these head losses. The solution of this nonlinear concave minimisation problem leads to an improved local optimum that may be used to restart the first phase. Iterations continue between the two phases in such a way as to obtain improving local optimal solutions.

A global optimum solution cannot be guaranteed (there are no techniques which guarantee a global optimal solution for looped systems except for an exhaustive enumeration), however Fujiwara and Khang (1990) believe the two-phase decomposition method allows for movement to better local optimal solutions which may be far apart in the solution space. Fujiwara and Khang showed extensions of the model for multiple sources, booster pumping, multiple loadings and for the expansions of existing systems.

Kessler (1988) and Kessler and Shamir (1991) developed a similar two-phase decomposition method. Results of the optimisation techniques of Kessler (1988) and Fujiwara and Khang (1990) as applied to the classic New York tunnels problem are shown in Chapter 8.

3.3.4 Global search and local optimisation

Randomness is a widely used tool in optimisation algorithms. Random search algorithms are popular because they can easily be implemented and combined with heuristic ideas to suit a particular search and they are often well suited to irregular solution spaces. Random search methods are usually a combination of random sampling and local optimisation. Torn and Zilinskas (1989) provided a comprehensive review of global search strategies including three types of random search methods: crude sampling, single-start and multi-start methods. Crude sampling is random exploration of regions of the solution space. Single-start is local optimisation starting from a randomly chosen point. Multi-start is local optimisation starting from multiple randomly chosen points.

Jacoby (1968) presented a procedure based on NLP and heuristics which was a variation of a multi-start random search of the pipe network design solution space. Jacoby considered the pipe network design problem for NP pipes as a nonlinear constrained optimisation problem in $2NP$ decision variables: the diameters D_j and the flows Q_j in the pipes j . The objective cost function which incorporates pipe capital costs and system operating costs is nonlinear in the decision variables D_j and Q_j . It is assumed that pipe cost can be approximated by some function of the pipe diameter. For example, a linear or polynomial function obtained by fitting a curve to the cost versus discrete diameter data. The objective cost function is subject to sets of linear and nonlinear constraints arising from the physical laws of continuity and energy applied to the network. Minimum and maximum diameter and pressure head bounds may be included as constraints.

Jacoby (1968) combined the objective function with the constraints in a 'merit function', F .

$$F(Q_1, \dots, Q_{NP}, D_1, \dots, D_{NP}) \quad (3.23)$$

The merit function is the cost function which is penalised with additional costs if a constraint is violated. Any kind or combination of constraints can be managed in this way. The penalties should be chosen such that the constraints are satisfied at the global minimum.

3 Optimisation of water distribution system design

Jacoby (1968) considered an iterative procedure that moved in the solution space from point to point. The step size of the movement varies with the success in reducing the merit function, however, it is initially set at a value which depends on the scale of the numbers. The direction of the movement is chosen in one of three ways:

1. negative gradient direction
2. random direction
3. experience direction

The directions are summarised by a direction vector, S

$$S=(s_1, \dots, s_{2NP}) \quad (3.24)$$

The components of the direction vector s_j or s_{2j} are the directions of movement with respect to the decision variable Q_j or D_j respectively. The negative gradient direction is the direction of the local optimum. The components of the negative gradient vector of the merit function can be approximated by finite movements from point to point in the solution in space moving with respect to the corresponding decision variable. The second direction of movement is a randomly chosen direction. The experience direction is a weighted average of the most successful directions of the most recent steps.

A jump step to a random new starting point, occurring at some low frequency is an attempt to avoid local optima. If the value of F cannot be reduced by any of the movements, the last point is the solution. The continuous diameters need to be rounded to commercially available discrete pipe sizes. Jacoby (1968) demonstrated substantial reductions in network cost for a simple 7-pipe system designed by this method.

Loganathan et al. (1995) showed the nonconvex, multimodal features of the pipe network optimisation solution space. They presented a pipe network optimisation procedure with an outer search scheme to choose alternative, feasible flow distributions to initialise an inner LP formulation which determined pipe diameters (and nodal heads) of a local optimal solution. The outer global search strategies of multi-start local search and simulated annealing are adopted to move between local optimal solutions and help locate isolated areas of the feasible region. The multi-start local search chooses a number of random initial flow distributions from the feasible region. Simulated annealing is an iterative improvement algorithm which compares the current solution with the new solution. The new solution is used if it is a better solution and otherwise it is discarded. To avoid being stuck on a local optimum, a worse point is accepted with some specified probability (the 'metropolis step').

Loganathan et al. (1995) recommended a gradient-based optimisation be carried out from the best point or other near-optimal solutions obtained by the outer search - inner optimisation method to locally refine the optimum. The method can consider multiple loadings, pumps and storage tanks. Loganathan et al. applied the model to the optimisation of the New York tunnel network expansion problem (results are shown in Chapter 8).

3.3.5 Dynamic programming

Dynamic programming is applicable to general, separable, serial optimisation problems. The problem is separated into subproblems called stages which are connected serially. Transition functions pass information between each stage. State variables or states record the current condition of the system at each stage. Kally (1972) used dynamic programming to determine the set of optimal pipe sizes for branched pipe networks.

Martin (1980) described a dynamic programming procedure which could be used to establish the minimum cost layout and design of a water distribution system. The procedure described by Martin is best explained by applying it to a simple example pipe network such as the one shown in Figure 3.2.

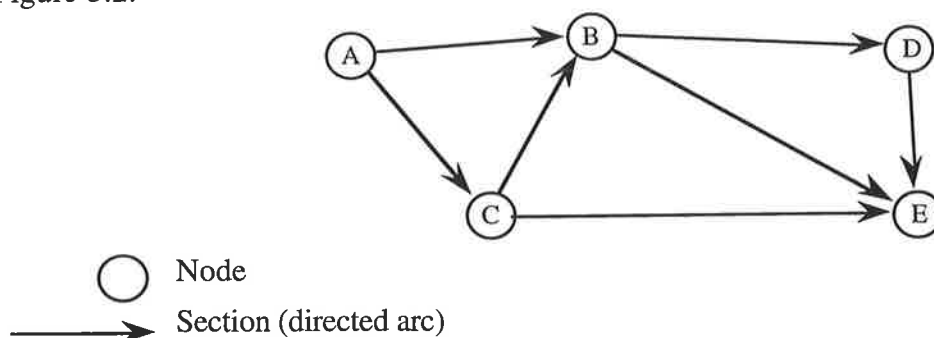


Figure 3.2 Example pipe network with assumed flow directions

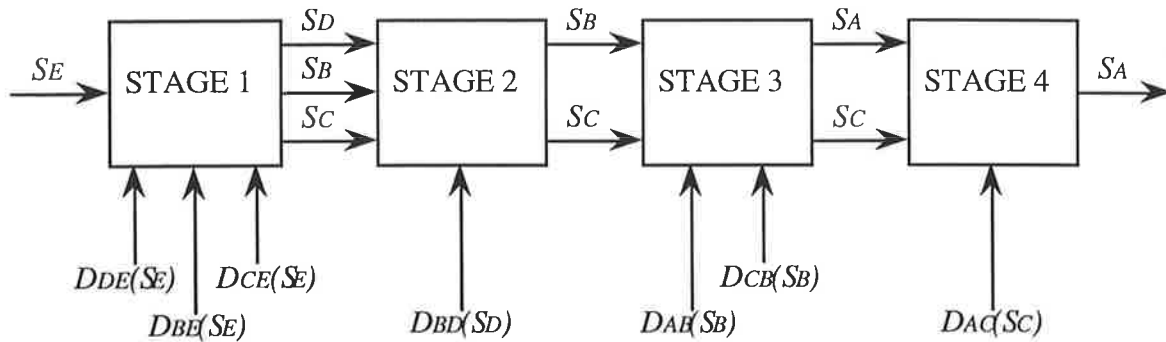
A specified flow is to be delivered from a source node A to a designated delivery node E along a single conveyance route. A set of possible routes make up a directed, non-looping network. A directed network is a set of nodes connected by a set of directed arcs. A non-looping network is one where there is no path along any set of arcs which leads from a node back to itself. The direction of the arcs indicates the direction of flow and this must be specified in advance.

The dynamic programming procedure works progressively backwards from the delivery node in stages. The stages represent sets of sections (for example, sections may be pipes or pumps) which are an equivalent number of sections back from the delivery node, considering the route with the maximum number of sections between the delivery node and the section under consideration. Stage N represents sections that are a maximum of $N-1$ pipes from the delivery

3 Optimisation of water distribution system design

node. The example pipe network is therefore decomposed into four stages as shown in Figure 3.3. Stage 1 represents sections BE, CE and DE. Stage 2 represents only section BD.

The input states for each stage are the HGL elevations of the downstream nodes of the sections under consideration or the output states of the previous stage. The output states for each stage are the HGL elevations of the upstream nodes of the sections under consideration or the input states which pass through the stage unchanged.



S_X Set of feasible HGL elevations and associated costs at node X

$D_{YX}(S_X)$ Set of decisions along the section YX dependent on state S_X

Figure 3.3 Decomposition of the dynamic programming procedure

Consider stage 1 for the example pipe network. A set of design decisions D_{BE} , D_{CE} and D_{DE} (eg., pipe diameters) are considered for sections BE, CE and DE for each feasible downstream HGL elevation S_E . The combination of design decision D_{BE} and downstream HGL elevation S_E generate an upstream HGL elevation S_B . The cost of providing water to the delivery node including the construction and operating costs of the design along a conveyance route is recorded at each stage for feasible upstream HGLs. The dynamic programming procedure moves progressively upstream in this way until it encounters the water source(s). In the example network, water source A is reached and the output states of stage 4 are the HGL elevations at node A. The feasible HGL elevation at node A associated with the minimum cost defines the optimal design and route.

Martin (1980) considered open channel canals and other canal structures and facilities, embankment dams, pipeline systems and pumping stations as the possible design decisions along each section. The dynamic programming procedure was implemented in a computer program called CANAL-I. The procedure was applied to proposed water conveyance systems in Texas.

3.3.6 Enumeration algorithms

Loubser and Gessler (1990) presented a partial enumeration technique. The technique which developed from the enumeration algorithm described by Gessler (1982, 1985) was proposed to overcome many of the practical considerations that make the optimisation of water distribution systems difficult. For the pipe sizing optimisation problem, a designer is usually limited to selecting discrete pipe sizes from a list of available pipe sizes. An enumeration technique makes use of this restriction by generating and evaluating possible combinations of discrete pipe sizes for the proposed pipe network layout. The combination of pipe sizes with the minimum cost and which meets the specified pressure head constraints is selected as the optimum.

A complete (exhaustive) enumeration of every possible combination of the decision variables is the only known technique that can guarantee the determination of the global optimum for looped networks. A limitation of complete enumeration is the computational effort required to investigate every possible combination of pipe sizes. For example, for the proposed expansions to the New York City water supply tunnels (Chapter 8) with 21 tunnels to be sized and with 16 available tunnel sizes to choose from, there are $16^{21} = 1.93 \times 10^{25}$ possible network designs. It is a practical impossibility to evaluate every possible solution for this relatively small network, even using the fastest computers.

Some modifications to the solution space are suggested by Loubser and Gessler (1990) to limit the search and reduce the computer time required. The modifications (1) and (2) reduce the extent of the solution space before the partial enumeration starts.

(1) An understanding of the overall flow pattern in the network from the supply to the demands may allow the designer to place certain pipes in pipe groups which are restricted to the same pipe size. Gessler (1982, 1985) argued that placing selected pipes in pipe groups simplifies the construction process and may help to provide redundancy.

(2) The original list of available pipe sizes may be reduced for the pipe groups.

The modifications (1) and (2) for reducing the search of the solution space require some experience with the pipe network under consideration. Regardless of experience and intuition, the pruning decisions to reduce the search space size will often exclude the optimum solution and therefore potential cost savings are lost.

Other modifications (3) and (4) to the search are applied to reduce the extent of the solution space as the partial enumeration proceeds.

(3) 'Cost Filter': There is no need to test a combination of pipe sizes for hydraulic feasibility if the cost of the solution is greater than some cut-off cost. The cut-off cost corresponds to the cheapest known solution that satisfies the pressure constraints.

(4) 'Size Filter': There is no need to test a combination of pipe sizes for hydraulic feasibility if all the selected pipe sizes are equal to or less than the pipe sizes of some other cut-off infeasible solution.

In some situations, reducing a pipe size in the system has been found to result in a flow pattern redistribution which increases the pressure head at critical nodes³. These circumstances bring the use of the size filter into question.

Some enumerative search schemes are devised so that all the pipe size combinations are visited systematically using say, a Breadth First Search (BFS) algorithm. All the pipe sizes start at the maximum pipe size and the search progresses systematically down the list of available pipe sizes. Gessler (1982) suggested the preliminary use of a Depth First Search (DFS) algorithm to quickly determine low cost feasible combinations of pipe sizes as well as some marginally infeasible combinations of pipe sizes for the effective use of the cost filter and size filter.

The partial enumeration of Loubser and Gessler (1990) identified a number of good solutions from the evaluated solutions. A disadvantage of the partial enumeration is the likely exclusion of the best solutions as a result of the severe pruning of the solution space. The technique also requires some knowledge of the optimal flow patterns for the system and candidate pipe sizes must be specified for the flow patterns. These are not obvious decisions, especially as network size increases and the system is subject to multiple demand patterns.

3.3.7 Heuristic techniques

Some practical optimisation problems especially in engineering may be very complex and are only rigorously defined by mathematics. It is reasonable then, that heuristic methods should be developed. Heuristics are procedures and rules particular to the search. They may be learned from experience with the problem and perhaps are not proven mathematically. Heuristics can generally assist in problem solving and generally reduce the effort of the search. Most of the optimisation algorithms reviewed in this chapter have used some heuristics.

³ This was demonstrated by the genetic algorithm optimisation of a relatively small (5 loops and 27 mains including 13 new and 14 existing mains) water distribution network for a new residential subdivision at Seaford in South Australia (Murphy et al., 1993a).

3 Optimisation of water distribution system design

Featherstone and El-Jumaily (1983) based a heuristic optimisation technique on the concept that a unique hypothetical linear gradient exists which corresponds to an optimal design. An extensive cost function was adopted which accounted for pipe, pump and tank costs, labour, maintenance and water treatment costs. The cost function is expressed in terms of the hydraulic gradient term, S_j for pipe j where:

$$S_j = \frac{h_{fj}}{L_j} \quad (3.25)$$

The pipes in the network are assigned arbitrary (usually identical) diameters as a starting assumption. A hydraulic analysis of the network is performed to determine balanced pipe flows. The coefficients of the cost function are evaluated since they depend on the pipe flows.

The optimal design is found by equating hydraulic gradients in all of the pipe sections S_j to a hypothetical value S_o or dummy optimal gradient. The cost function is a minimum where:

$$\frac{d(Cost)}{dS_o} = 0 \quad (3.26)$$

The value of S_o is computed by differentiating the cost function, setting the first derivative to zero and solving for S_o . The method assumes a cost function for which the first derivative exists. A set of new diameters for the pipes are given by substituting $S_j=S_o$ into the Hazen-Williams friction head loss formula for all the pipes j . The continuous pipe sizes may be rounded to commercially available discrete pipe sizes. The new pipe network corresponding to S_o is balanced using a steady-state hydraulic simulation model. The coefficients in the cost function are revised using the balanced pipe flows. An adjusted value of S_o is found and the pipe network solution corresponding to this value of S_o is determined. The procedure is repeated until the least cost design is reached.

Monbaliu et al. (1990) reported a pipe network optimisation technique which could be described as a type of enumeration search or gradient search incorporating heuristics. The technique is implemented in a computer program called PIPENET. The optimisation algorithm is coupled with a Hardy Cross hydraulic simulation model in PIPENET. The user supplies the algorithm with the network data and a list of candidate pipe sizes. The pipes to be sized are initially allocated the smallest pipe diameter and the proposed network is checked for hydraulic feasibility. If the pressure heads are inadequate, the diameter of the pipe with the largest head loss per metre is increased to the next available pipe size. The procedure is repeated until a feasible solution is obtained. The optimisation algorithm was compared to linear and nonlinear programming and produced similar designs for two simple example pipe networks.

3.3.8 Equivalent pipe methods

Equivalent diameter methods and equivalent length methods are specific types of heuristic search methods for pipe network optimisation. Deb and Sarkar (1971) developed a method of pipe network optimisation based on an equivalent diameter method which can be used to obtain the minimum cost combination of pipes for a given set of nodal pressure heads. The method assumed the pipe network consisted of pipes of equal lengths L_e and roughness coefficients C_e and equivalent diameters D_{ej} such that:

$$D_{ej} = \frac{K_1 Q_j^{0.381}}{h_{fj}^{0.206}} \quad (3.27)$$

The cost function is assumed to be of the form:

$$Y_j = K_2 D_{ej}^m \quad (3.28)$$

in which

Y_j = total cost of pipeline of equivalent diameter D_{ej}

K_1, K_2 = constants

m = an exponent

Deb and Sarkar (1971) substituted Eq. 3.27 into Eq. 3.28 to eliminate D_{ej} and formed the relationship:

$$Y_j = \frac{K_3 Q_j^{0.381m}}{h_{fj}^{0.206m}} \quad (3.29)$$

The friction head loss h_{fj} in pipe j can be determined from the difference in node pressure heads which are specified in advance. In addition, the pressure heads fix the direction of flow in the pipes. Deb and Sarkar (1971) applied Eq. 3.29 to the pipes in a loop and concluded that for the minimum total cost of pipe in a loop the following condition must be satisfied:

$$\sum \frac{D_e^m}{Q} = A' \quad (3.30)$$

where A' is a constant. The optimum value of A' is the maximum value that does not disturb the assumed flow directions. The problem needs to be solved for increasing trial values of A' .

The method also requires a set of initially assumed flows Q_j . The corresponding equivalent diameters D_{ej} can be determined from Eq. 3.27. A known value of A' can be found by applying Eq. 3.30 to the pipes in a loop. Deb and Sarkar (1971) derived a relationship for correction factors ΔQ for flow to be added algebraically to the initially assumed flows in a loop. The procedure is repeated and convergence to the solution in each loop is observed. The equivalent diameters D_{ej} can be converted to the actual diameters D_j with actual length L_j and roughness C_j by the relationship:

$$\left(\frac{D_{ej}}{D_j}\right)^{4.8704} = \left(\frac{L_e}{L_j}\right) \left(\frac{C_j}{C_e}\right)^{1.852} \quad (3.31)$$

The method requires a set of assumed fixed pressure heads for the nodes to be specified. Various solutions may be obtained for various sets of assumed pressure heads. Deb and Sarkar attempted to determine the optimum set of pressure heads by assuming the pressure head profile between the source node and the critical node (usually an extreme downstream point in the network) can be represented by a quadratic function. The validity of the equivalent pipe methods has been questioned. Swamee and Khanna (1974) showed that the equivalent pipe methods essentially fix the hydraulic gradient.

Raman and Raman (1966) considered an equivalent length method to determine optimal pipe sizes from known node pressure heads. Initially, all the pipes j in the network are replaced by pipes of equivalent length L_{ej} of a constant diameter. The technique was based on the concept $\Sigma L_e/Q=0$ must hold for pipes around a loop if the total sum of equivalent lengths ΣL_e of pipes in the loop is a minimum. A correction factor for pipe flows was derived to be applied to a set of initially assumed flows. The actual lengths of pipe are used to determine the actual diameters of the equivalent lengths of pipe.

3.3.9 Evolutionary strategy

The evolutionary strategy was developed by Rechenberg (1973) in Germany, independently of, but in parallel with the development of genetic algorithms by Holland (1975) in the United States. Rechenberg believed that the imitation of the processes of biological evolution should be a promising experimental method for optimising engineering design. Rechenberg also claimed the rules of biological evolution were themselves evolving, such that evolution is an optimal mode of operation.

Rechenberg (1973) described biological evolution most simply in terms of mutation which is the occasional random alterations of members and selection which is the survival of members of high resilience. A two-membered competitive scheme is a simple concept in which a variation

is made to a single member and a decision is made to keep or take back the variation depending on the capacity for survival of the child member compared to the parent member. The two-membered competitive situation is extended to the multi-membered competitive situation. The mechanisms of mutation and selection are applied to a collection of members. A process of recombination may be introduced which is the continuous mixing of parent members.

Cembrowicz and Krauter (1977) utilised an optimisation technique which combines graph theory, linear programming and the evolutionary strategy to optimise a pipe network. A tree is a branched network that contains no loops such that every node is connected. Tree configurations were considered since they correspond to local cost minima. The evolutionary strategy was composed of five major operators based on the principles of mutation, selection and recombination. The evolutionary strategy was used to generate trees by 'permutating the chords in any loop', since a tree network is a loop network minus one chord in each loop. Linear programming was used to optimise the generated tree networks. Since the flows in the tree network are known, the objective cost function is subject to a set of linear constraints.

3.4 Complexities of Pipe Network Optimisation

The complexities of the pipe network design problem has restricted the development of practical optimisation techniques. The following outlines some issues that should be addressed.

Pipe network optimisation is a highly dimensional problem which makes it difficult to solve. The dimensionality of the solution space is determined by the number of decision variables. Dimensionality (and computational time and storage requirements) increase as the pipe network size increases. Pipe network optimisation techniques should be capable of efficiently operating with large pipe networks. Partial enumeration algorithms may experience difficulty as the pipe network size increases, as the near-optimal flow patterns become more difficult to predict and the computational time needed to perform a hydraulic simulation of trial pipe network designs increases significantly.

Water distribution systems are designed to fulfil the diverse water needs imposed by a complex combination of users for domestic, industrial, commercial and emergency (eg. fire-fighting) purposes. In addition, water needs fluctuate with the seasons and during the day. Multiple demand patterns should be analysed to ensure redundancy and a reliable design (Lansey and Mays, 1989b). The critical demand pattern is often difficult to predict. Linear programming and nonlinear programming optimisation methods (that are not coupled to a hydraulic simulation model to evaluate feasibility) generate new sets of hydraulic and system performance constraints for each new demand pattern and the formulation can become cumbersome for multiple demand patterns.



Pipe network optimisation is a highly nonlinear problem, since the cost equations and hydraulic equations for flow in the system components are usually nonlinear expressions. Methods such as linear programming attempt to simplify the problem by linearising the objective function and the constraints by assuming some network conditions. The assumptions immediately place restrictions on the regions on the solution space that may be searched.

Pipe network design is a global optimisation problem. The global minimum is one of many local minima in the solution space and the number and distribution of local minima is difficult to predict. Although some constraints may be extended to their bounds for the local minimum, the cost of the local minimum may still be considerably higher than the cost of the global minimum. For these reasons, local search techniques without some global strategy are not well suited to pipe network optimisation.

Some optimisation methods derive a smooth and continuous pipe cost versus diameter relation by approximating the pipe cost by some function such as a polynomial in the diameters. Approximation of the pipe cost functions can contribute to errors in computation of the objective cost function and hence mislead the optimisation technique.

Pipe diameters are usually a discrete set of commercially available pipe sizes. Some linear programming and nonlinear programming methods operate with continuous pipe diameter variables. Continuous solutions require the rounding of continuous diameters to discrete diameters to be of practical value. The rounding of continuous solutions is itself a secondary optimisation problem. In addition, the rounding of continuous solutions does not ensure an optimal or feasible design. In split pipe designs, the pipe sections of continuous diameter are reduced to lengths of the two adjacent discrete pipe diameters. The partial lengths are chosen so as to create a pipe section with equivalent hydraulic properties.

Other design parameters such as cleaning, lining, duplication or deletion of existing pipes are often difficult to incorporate into the optimisation model. In practice, alternative pipe materials with different pipe roughnesses and internal diameters for a given nominal diameter may be available.

The pipe network may be a new system, but often expansions to an existing system are required to meet increased water needs. The optimisation technique should be capable of handling both situations. Techniques which are only applicable to optimising branched networks have limited practical value, since most urban pipe networks are looped. The optimisation procedure should be capable of handling the simulation and optimisation of the design and operation of other system components such as pumps, tanks, pressure reducing valves and other flow and pressure regulating devices.

4 Overview of Genetic Algorithms

Genetic algorithms (GAs) are a class of powerful search procedures which simulate an evolution strategy (Holland, 1975; Goldberg, 1989). Evolution is the process of gradual and continuous growth by which species of living things develop from earlier forms. Charles Darwin proposed new species arose by a process called natural selection acting on individual variations in a population. The study of genetics explains the individual variations by genetic mechanisms such as spontaneous mutations.

The genetic algorithm (GA) search is a simplified computer simulation of Darwin's rules of natural selection and mechanisms of population genetics. In nature's established optimisation process, fitter populations of individuals evolve and adapt to the current environment over many generations. In the artificial GA evolution, simplified versions of natural selection rules and genetic mechanisms are applied to an evolving population of trial solutions; to identify global optimal (or near-optimal) solutions in a solution space defined by the optimisation or search problem.

Genetic algorithms have been theoretically and empirically proven to effectively search complex solution spaces. DeJong (1975) demonstrated the far-reaching possibilities of the GAs when he subjected the simple GA and five variations to a set of search spaces with diverse characteristics. Since their introduction by Holland in 1975, the GA search, sometimes with modifications to the traditional GA formulation, has performed efficiently in a number of scientific, engineering, economic and even artistic applications (Goldberg, 1989). This demonstrates the robustness of the GA search and the flexibility of the GA formulation. In this thesis, the GA search is applied to the global search of the vast, complex solution space for a water distribution system (looped, pumped) optimisation problem. In this chapter, the traditional genetic algorithm method is introduced and some applications relating to the genetic algorithm formulation developed in this thesis are reviewed.

4.1 Chromosomes

The genetic algorithm (GA) represents individual solutions by some unique encoded structure such as a string or block of binary bits. Coded strings may represent individual solutions to the optimisation problem such as pipe network designs, just as a *chromosome* of genetic code characterises an individual of a species in biological systems.

Chromosomes are arrangements of genes which are the inherited biological factors that define the characteristics of an individual. In natural systems, the total genetic make-up of an

4 Overview of genetic algorithms

organism is contained in a package of one or more chromosomes called a *genotype* (Goldberg, 1989). In the genetic algorithm, the coded string is the artificial genotype or collection of chromosomes which represents the decision variables for the optimisation problem. In natural systems, the *phenotype* is the organism formed by the interaction of the genetic make-up with its environment (Goldberg, 1989).

The first task of the GA user in the implementation of a GA model for a new application is to develop a coding scheme to represent solutions to the optimisation problem as coded structures. Traditionally, substrings of binary code arranged in a coded string decode to the decision variable choices of the optimisation. Alternatively, substrings of Gray code (Bethke, 1981; Caruana and Schaffer, 1988), integer values, letters (Davis and Coombs, 1987), or real numbers (Goldberg, 1990) may be mapped to decision variable choices to represent a trial solution to the optimisation problem. Consider a coded string representing a trial pipe network design for the hypothetical pipe network design problem shown in Figure 4.1. The five decision variables for the optimisation are the diameters of the proposed new pipes [1] to [5].

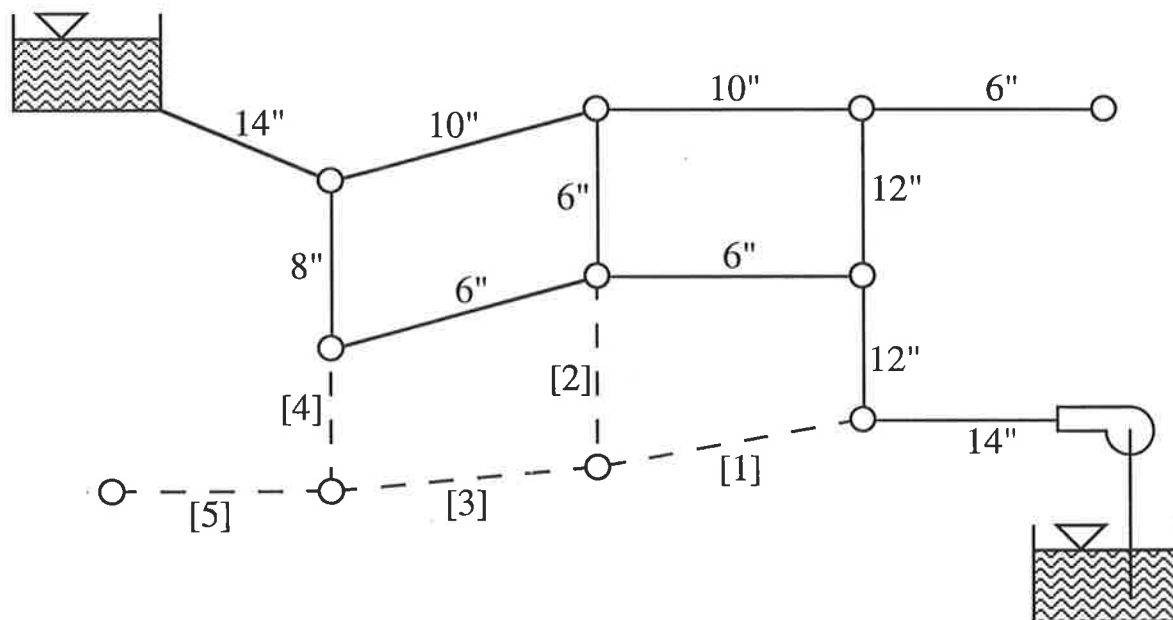


Figure 4.1 Hypothetical pipe network design problem requiring the selection of five new pipe sizes

The coded string is constructed of five coded substrings each corresponding to one of the five decision variables of the optimisation problem. In this case, the substring positions correspond to the new pipes to be sized:

Coded substring position:	1 - 2 - 3 - 4 - 5
Corresponding pipe number:	[1] - [2] - [3] - [4] - [5]

4 Overview of genetic algorithms

The unique substring code at a substring position decodes to the proposed decision variable choice (eg., pipe diameter) in the trial pipe network design by observing a specified mapping between the substring code and the decision variable choices. For example, if the available new pipe sizes for the hypothetical pipe network design were 4", 6", 8" and 10" diameters, one of the mappings in Table 4.1 may be used to represent trial network solutions.

Table 4.1 The representation of design parameters by pieces of code

PHENOTYPE	GENOTYPE		
Available new pipe diameters	Unique substrings of binary code	Substrings of Gray code	Integer code
4"	00	00	1
6"	01	01	2
8"	10	11	3
10"	11	10	4

If for example, the optimal solution to the hypothetical pipe network design problem was:

$$D_{[1]}=10" - D_{[2]}=6" - D_{[3]}=8" - D_{[4]}=8" - D_{[5]}=4"$$

Then the coded strings representing the optimal pipe network design solution are as follows:

- Substrings of binary codes: 11 - 01 - 10 - 10 - 00
- Substrings of Gray codes: 10 - 01 - 11 - 11 - 00
- Substrings of integer code: 4 - 2 - 3 - 3 - 1

4.2 Fitness of Coded Structures

In natural systems, fitness may reflect a living thing's compatibility with its surrounding conditions and ultimately regulates its survival. Similarly, the fitness value of a coded string (an artificial chromosome) is a measure of the quality of the solution and determines its chances of survival in the genetic algorithm search. Each coded string is assigned a fitness and the GA searches for the string with the highest fitness.

The fitness function is a relationship between the fitness value of a string and the objective function value of the trial solution, and may incorporate a penalty for infeasible solutions. A penalty method is one approach to a constrained optimisation problem. Infeasible solutions are unacceptable solutions which do not satisfy one or more of the constraints of the optimisation.

The genetic algorithm penalises infeasible solutions by reducing their fitness. The penalty applied to a solution is usually function of the distance from feasibility (Richardson et al., 1989) such that penalties increase as the quality of solution decreases. The penalty can be a function of the number of violated constraints or a function of the degree by which the solution violates the constraints.

The penalty is the cost of achieving feasibility. Infeasible solutions are not discarded from the GA search since valuable genetic information may be contained in a coded string representing an infeasible solution which just fails to satisfy the constraints. Some infeasible strings will be composed of pieces of the optimal string since the global optimal solution lies close to the boundary between feasible and infeasible solutions and is only a step away from feasible and infeasible solutions. The penalty function should be selected so that the GA search approaches the optimum solution from both the feasible and infeasible regions of the solution space.

4.3 The Solution Space

A coded string is a trial solution and the set of all coded strings describes the solution space to be explored by the genetic algorithm search. The number of decision variables and the number of possible choices for the decision variables determines the size of the solution space. The fitness values of the coded solutions reflects the topography of the solution space. The solution space for the constrained pipe network optimisation problem is made up of a feasible region (the set of solutions which satisfies the constraints) and an infeasible region.

4.4 Populations of Coded Structures

In nature, organisms live within competitive populations. Similarly, the genetic algorithm considers a collection or population of coded strings at any given time. The population represents some distribution of trial solutions in the solution space. The size of the population can be anywhere from 10 to 1,000 or more coded structures. The population size is kept constant throughout the traditional GA evolution, although the population size may vary through the generations for some GA formulations.

The GA artificial evolution process is initiated with a starting population. The starting population of coded strings is usually generated randomly although some GAs may bias the starting population in some way to attempt to help direct the search. A randomly generated starting population of coded strings represents a random distribution of trial solutions in the solution space, giving the GA a picture of the global topography.

The GA successively evaluates and regenerates new populations of coded strings. The GA search generates new populations of coded solutions from old populations using simple operators. The GA effectively searches and improves from a population of strings and their corresponding fitness values.

4.5 Genetic Algorithm Operators

The chromosomes of living things are combined and manipulated by genetic mechanisms from generation to generation. The genetic algorithm performs simple operations to the current population of coded structures to create a new population. The operations imitate nature's rules of survival and mechanisms of population genetics. The simple, yet powerful traditional genetic algorithm employs selection, crossover and mutation operators.

Consider the schematic of an evolution strategy in Figure 4.2. The shading of the artificial chromosomes represents fitness, such that the darker chromosomes are fitter. In this simple representation, the fitter chromosomes are more likely to survive and reproduce and their progeny make up the new generation of chromosomes. The new chromosomes are formed from parts of their parent chromosomes and occasionally a gene mutation occurs in a child chromosome. As the generations proceed, the population gradually becomes fitter and more competitive.

4.6 Selection

Selection (or reproduction) is based on Darwin's survival-of-the-fittest philosophy of natural selection. The fittest coded solutions from a population of solutions are more likely to be selected to advance to the next generation of the GA evolution by the selection operator.

In natural systems, the survival of a living thing depends to some degree on its strength (or fitness) and to some degree on good fortune. In a similar fashion, the selection of a coded string from within a competing population of strings depends on the fitness of the string relative to fellow strings and chance factors.

The traditional selection operator is called *proportionate selection* or *stochastic sampling with replacement* (Goldberg, 1989). Goldberg likened proportionate selection to a weighted roulette wheel with segments corresponding to the strings in the population and sized according to the string's fitness. The roulette wheel segments are sized such that strings with higher fitness have a higher probability of selection.

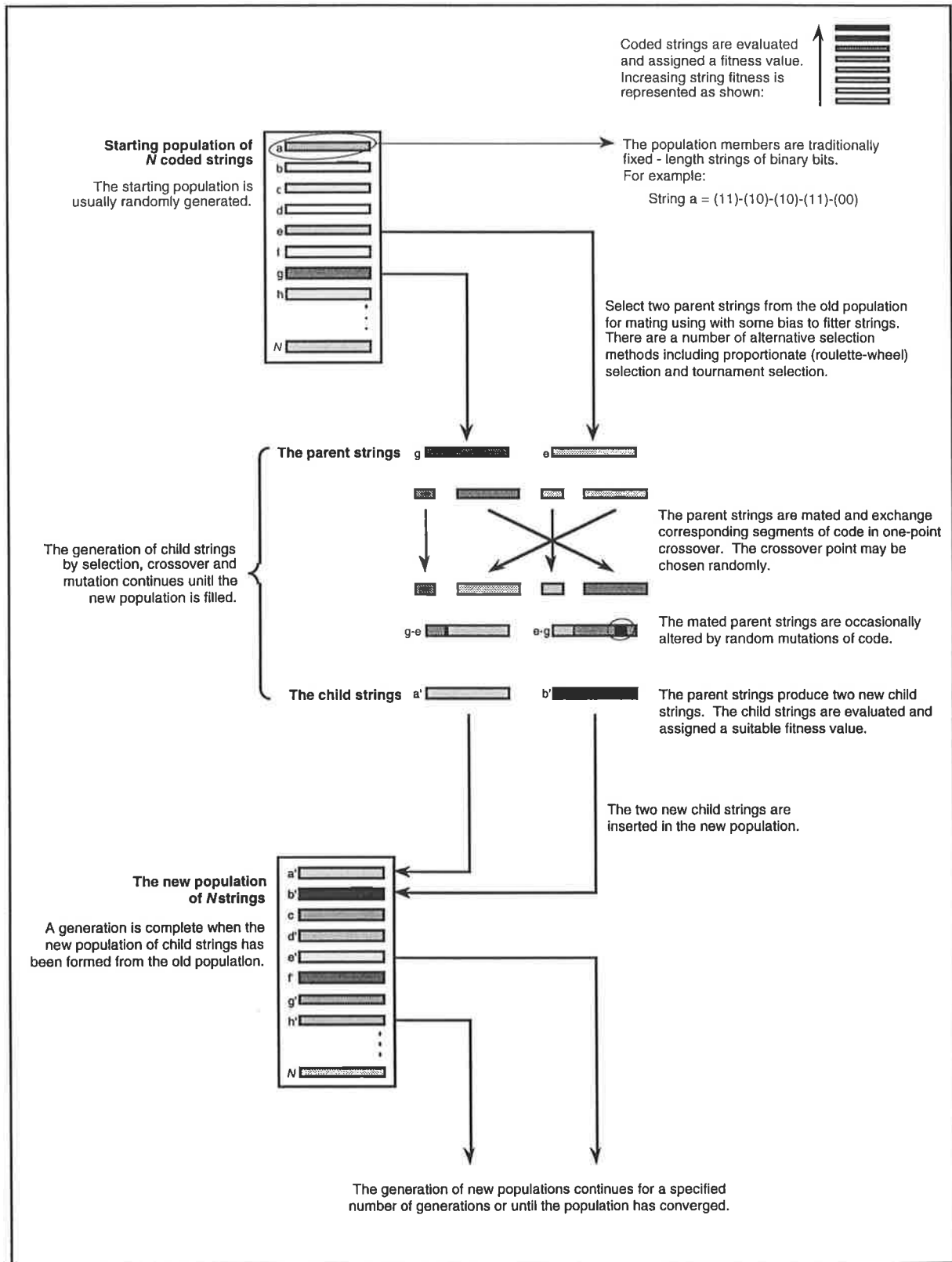


Figure 4.2 A model of a generation of a simple evolution strategy

4 Overview of genetic algorithms

The fitness assigned to a string must differentiate it from stronger and weaker strings to allow the string an appropriate chance of selection in the reproduction process.

Some alternative parent selection methods include tournament selection and ranking selection (Goldberg and Deb, 1991). Tournament selection selects two (or more) parent strings from the old population by stochastic sampling and the parent string with the highest fitness is declared the winner and proceeds to produce offspring strings in the new population.

Hollstien (1971) investigated a number of parent selection methods and mating methods based on natural breeding practices and controlled breeding practices applied by man to plants and animals (Goldberg, 1989).

The traditional selection scheme of proportionate selection or roulette-wheel selection has been used successfully combined with the use of fitness scaling techniques (Goldberg, 1989), although tournament selection is often the preferred selection operator since selection and fitness scaling are combined into one step (Goldberg and Deb, 1991).

4.7 Crossover

Crossover mimics the mixing of chromosomes and swapping of genes that occurs when two living things mate. The genetic algorithm crossover operator breaks two parent strings selected from the old population and exchanges segments of code to produce two offspring strings in the new population.

The traditional GA crossover operator is simple one-point crossover. A position in the coded string is selected randomly and crossover proceeds by breaking the coded strings and exchanging corresponding segments of code between the strings after the crossover site.

Some alternative crossover operators include multiple-point crossover, uniform crossover and shuffle crossover (Eshelman et al., 1989). Multiple-point crossover is the random selection of two or more crossover sites and the exchange of corresponding alternate segments of code. In uniform crossover, corresponding bits of code are exchanged rather than segments of code. The bits may be exchanged between the parents with some probability. In adaptive crossover, the crossover sites are not chosen randomly, but instead are chosen with some probability that may be specified or determined as the GA run progresses.

4.8 Mutation

Mutation is the occasional, random alteration of code in the offspring strings. The genetic algorithm mechanism of mutation spontaneously alters pieces of code to prevent the loss of a potentially useful genetic trait.

The traditional bit-wise complement mutations operate on randomly selected offspring strings of binary bits. A bit position is randomly selected and the bit value is inverted.

Decision-variable-wise mutation performs a mutation of decision variable substring code chosen randomly from the coded string. Goldberg (1990) suggested the use of traditional bit-wise mutations and decision-variable-wise or phenotypic mutations in binary-coded GAs to overcome such problems as Hamming Cliffs.

A marked mutation in any environment may not easily survive, however, more subtle mutations may survive. Davis and Coombs (1987) introduced an operator called *creep* in their study of the design of packet switching communication networks. The coded strings were lists of communication link speeds which corresponded to links in the network and creep altered selected link speeds upward or downward one or more steps in the list of allowable link speeds.

4.9 String Similarities (Schemata)

Holland (1975) developed the Schema Theorem for genetic algorithms. A schema describes a family of coded strings related by coding similarities at given positions on the string. A schema is a similarity template (Goldberg, 1989) describing a subset of strings with similarities at given positions on the string. Schemata, a set of schema, are a way of representing similarities between strings of the same string length and constructed with the same alphabet.

Consider the set of strings with a string length of 15 bits coded using the binary alphabet $\{1,0\}$. There are 2^{15} unique coded strings of 15 binary bits. Schemata are constructed using the ternary alphabet $\{1,0,*\}$ in which the $*$ is sometimes called the 'don't care' symbol. There are 3^{15} unique schemata which may be defined for a coded string of 15 binary bits. In general, there are $(k+1)^l$ schemata, for a string of length l constructed with an alphabet of cardinality k (Goldberg, 1989). The examples of schemata in Table 4.2 are accompanied by coded strings which belong to the family of coded strings designated by the schemata. Schema H_l describes a subset of 2^{11} coded strings with an identical 0-1 structure in the bit positions 3 to 6 and either

4 Overview of genetic algorithms

a 0 or a 1 in the bit positions occupied by a *. Three members of this family are shown in Table 4.2.

The order of a schema is the number of fixed positions in the schema. The low order of schema H_1 is 4 and the relatively high order of schema H_2 is 11. The order of schema H_3 is 5. The defining length of a schema is the distance between the outermost specific positions of a schema. The schema H_1 has a short defining length of 3, the schema H_2 has a defining length of 11 and the schema H_3 has a relatively long defining length of 13.

The GA interprets and exploits the relationship between string similarities and corresponding fitness values in a population of strings. As such, the GA identifies building blocks which are schemata of low order and short defining length and which are associated with high fitness (Goldberg, 1989). These building blocks are an important element in the action of genetic algorithms. Should schema H_1 in Table 4.2 be found in strings of high fitness, it could be an effective building block with its low order and short defining length. Goldberg considered the effects of selection, crossover and mutation on the expected numbers of schemata in subsequent populations and concluded the power of the GA is in the propagation and the combination of building blocks.

Table 4.2 Example schemata and subset members

EXAMPLE SCHEMATA AND FAMILY MEMBERS FOR CODED STRINGS OF 15 BINARY BITS		
Schema H_1	**1-110-***-***-***	111-110-000-010-110 001-110-111-110-110 011-110-010-111-000
Schema H_2	*11-*10-000-010-1**	111-110-000-010-110 011-010-000-010-110 111-110-000-010-101
Schema H_3	1**-11*-***-***-11*	111-110-000-010-110 101-111-101-001-111 110-110-010-000-110

GAs exploit string similarities by allowing more copies of the fitter strings to continue to the mating pool during selection. The GA selects strings with high fitness more often in reproduction to contribute further to the GA evolution. Hence highly fit string similarities or

highly fit schemata survive and accumulate in successive generations. The offspring strings inherit the combined highly fit schemata of their parents in crossover. Schemata of short defining length and low order are less likely to be disrupted by the effects of crossover.

4.10 The Power of the Genetic Algorithm

Selection rules and genetic code modifications from generation to generation of coded strings attempt to reproduce the efficiency of the development of living things towards adaptation. In nature, weak genetic characteristics of living things are discarded in time while the strong genetic characteristics of living things become prominent. In the GA, building blocks are regenerated in reproduction and are combined with other building blocks by crossover to construct superior new strings from the best parts of the best old strings. Subsequent populations of coded strings converge on 'fitter' regions of the solution space.

The GA searches from a population of coded strings and their corresponding fitness values. All the information required for the search is contained in these two elements. The GA does not rely on auxiliary information such as continuity or the existence of derivatives. The power of the GA comes from the exploitation of the special relationship between the fitness of strings and the string similarities within a population.

The GA works with an evolving population of trial solutions where conventional methods move from a single point in the solution space to the next. The starting population is usually randomly distributed throughout the solution space. The GA climbs many peaks in parallel which is particularly important in many-peaked and discontinuous solution spaces. The GA only searches a fraction of the total number of solutions before near-optimal solutions are determined.

The GA produces a number of acceptable solutions that are near-optimal and yet perhaps quite different. The penalty function can be modified so that consideration can be given to solutions that are just infeasible. The GA technique does not exclude infeasible solutions from the search. A well chosen penalty function considers the degree by which the constraints are violated and calculates an appropriate penalty cost. Solutions that are just infeasible are recognised as useful solutions since the optimal solution lies on the boundary between feasible and infeasible solutions.

4.11 GAs applied to Pipe Optimisation Problems

In recent years, researchers have proposed the genetic algorithm approach for aspects of the design of pipeline systems. Goldberg and Kuo (1987) applied the traditional GA to the

4 Overview of genetic algorithms

optimisation of the operation of a steady-state serial gas pipeline consisting of 10 pipes and 10 compressor stations each containing 4 pumps in series. The objective was to minimise power, while supplying a specified flow and maintaining allowable pressures. The simple three-operator GA found near-optimal pump operation alternatives after evaluating a fraction of the total possible number of solutions (about 3500 from 1.10×10^{12} possible combinations).

Davidson and Goulter (1992a, 1992b) used GAs to optimise the layout of a branched rectilinear network such as a rural natural gas or water distribution system. The optimal layout was assumed to be the one of least length. The layout solutions were represented by blocks of binary code and new GA operators of recombination and perturbation were introduced to reduce the numbers of infeasible solutions created by the traditional GA operators of crossover and mutation.

Walters and Lohbeck (1993) found the GA effectively converges to near-optimal branched network layouts selected from a directed base graph which defines a set of possible layouts. The nodal connectivity of the trial branch network solutions was represented by a string of code. Alternative GA coding schemes were investigated including a binary representation and an integer representation. Walters and Cembrowicz (1993) extended these ideas using linear programming for the optimal selection of pipe sizes for branched pipe networks generated by a GA. The combination of GAs, graph theory and linear programming formed an effective search for near-optimal branched pipe network designs.

In the next chapter, a methodology is proposed for the application of the traditional genetic algorithm to the pipe network optimisation problem.

5 Application of the Traditional Genetic Algorithm to Pipe Network Optimisation

The traditional three-operator genetic algorithm (GA) introduced in the previous chapter is applied to a relatively simple pipe network design optimisation problem in this chapter. The traditional GA applies proportionate selection, one-point crossover and occasional random bit-wise mutations to an evolving population of fixed-length binary strings. The fixed-length binary strings characterise the form of representation space to be searched. The GA optimisation of the multimodal, high-dimensional objective function for the pipe network optimisation problem presented in this chapter is based on the traditional form of the GA, to take advantage of the powerful properties identified by Holland (1975). The GA search, sometimes with modifications to the traditional GA formulation, has been shown to perform efficiently in a number of different applications. This efficiency indicates the robustness of the search method that underlies the GA approach and the flexibility of the formulation itself (Goldberg, 1989).

5.1 A Genetic Algorithm Approach

The genetic algorithm search can be formulated in many different ways, such is the flexibility of the technique. The GA formulation presented in this chapter outlines a traditional GA approach to pipe network optimisation (Simpson, Dandy and Murphy, 1994).

5.1.1 Coded strings

In a simple form, the pipe network optimisation problem is the selection of a combination of pipe sizes in a gravity fed water distribution system, such that pipe material and laying costs are minimised (Chapter 3). The pipe network design is subject to system performance constraints, hydraulic constraints and other design constraints. The decision variable choices for this standard pipe network optimisation problem are the new pipe sizes. The GA represents the set of decision variable choices describing a trial solution by a unique coded string of finite length (Goldberg and Kuo, 1987). The coded string is similar to the structure of a chromosome of genetic code. The {0,1} binary code (the minimum alphabet) is chosen as the representation mapping in the traditional GA application. Consider a coded string of ten binary bits, constructed of five coded substrings each of two binary bits:

11 - 01 - 10 - 10 - 00

This artificial genetic code could represent a trial pipe network design for a pipe network optimisation problem. The five decision variable substrings each correspond to one of five new

pipes. The five decision variables for the optimisation are the five pipe diameters to be selected for the new pipes. The 2-bit substring symbols {00,01,10,11} map the decision variable to one of four possible choices of discrete pipe diameter.

5.1.2 Fitness of coded strings

The coded string representing a trial pipe network design is assigned a fitness. The fitness is a value which measures the quality of the solution. The fitness of a coded string is analogous to the fitness of the genetic make-up of an individual member of a species. The fitness value to accompany the coded string is determined by the pipe costs and the hydraulic performance of the pipe network design. The total cost of a pipe network design is taken as the sum of 1) the present value of material, construction, maintenance and operation costs and 2) penalty costs (where the system performance constraints such as minimum pressure requirements are violated).

The simple GA search operates with strings of binary code and the associated fitness values. All the information required for the search is contained in these two elements. By comparison, some traditional optimisation methods rely on the existence and continuity of derivatives or other auxiliary information (Goldberg and Kuo, 1987).

5.1.3 Implementation of a simple genetic algorithm

The genetic algorithm (GA) procedure involves the following steps:

Generation of an Initial Population of Coded Strings

The GA operates directly with a population of coded strings at any given time. The simple GA randomly generates an initial population of coded strings. The bit positions in the string take on a value of either 0 or 1 produced by a random number generator. A recurrence called a linear congruential generator is used in this study to propagate sequences of pseudo-random numbers (Barnard and Skillicorn, 1988).

The population size, N (typically $N = 50$ to 1000) is a GA parameter which is usually selected based on the length of the coded string. The population of N coded strings represent N different pipe network configurations. The randomly generated starting population of strings represents a random distribution of trial pipe network solutions in the solution space.

Decoding and Evaluation of Coded Strings

The GA decodes and evaluates each of the N coded strings in the current population in turn. The strings of binary code are decoded to pipe network designs by observing a specified mapping between binary substrings (genotype) and decision variable choices such as pipe diameters (phenotypes). The evaluation of the pipe network design to determine a suitable measure of string fitness consists of an estimation of network expansion costs and a hydraulic analysis of the network design. If crossover does not occur for a selected pair of parent strings in the generation of new populations, the previously computed fitness values are used.

Computation of Pipe Network Costs

The GA estimates the pipe network costs for each decoded pipe network design in the current population. The pipe material and laying costs, and system operation and maintenance costs may be included in the estimation.

Hydraulic Analyses of Trial Pipe Networks

A hydraulic simulation model is used to assess the hydraulic feasibility of the proposed pipe network designs. The node pressures and pipe flows are computed for the predicted demand patterns for each of the trial pipe networks in the current population. For the initial population, N pipe networks are analysed. The actual node pressures are compared with the minimum allowable pressures and pressure deficits are noted.

The hydraulic simulation model is linked to the evaluation scheme of the GA optimisation model. The method of hydraulic analysis adopted in this research uses the Newton-Raphson technique applied to the set of simultaneous nonlinear algebraic equations in terms of the unknown flow corrections around the loops (Chapter 2). The hydraulic analyses are computationally intensive and special techniques such as sparse matrix routines are implemented to ensure the efficiency of the hydraulic simulation model.

Computation of Penalty Costs

The pipe network designs are subject to system performance requirements such as minimum node pressures. The constraints divide the solution space into a *feasible* region containing the solutions which satisfy the constraints and an *infeasible* region containing solutions which do not satisfy the constraints. The constraints are included in the GA formulation by way of a penalty function method, which effectively formulates the constrained optimisation problem as an unconstrained problem. The penalty cost applied to infeasible pipe network solutions is usually some function of the degree by which the design violates the system performance

constraints. Penalty costs are computed if the pipe network design does not maintain minimum allowable pressures while supplying the specified demand flows.

The penalty function used in this study is the product of the pressure violation(s) (the sum of all node pressure violations or the maximum node pressure violation) for each demand pattern and a chosen penalty factor, k . The penalty cost is thought of as the cost of achieving feasibility and the penalty factor is interpreted as a measure of the cost of a deficit of a unit of pressure head (e.g., $k = \$100,000 / \text{metre of pressure head deficit}$). The GA computes the penalty costs for each infeasible trial pipe network design in the current population. For strings duplicated in subsequent generations (not disrupted by crossover or mutation), the previously computed penalty costs are used.

The optimum solution lies close to the boundary between feasible and infeasible solutions. The penalty factor should be carefully selected such that near-optimal infeasible solutions are highly fit (but not optimum), so that the search approaches the optimum solution from both the feasible and infeasible regions of the solution space. Some trial and error may be necessary to select a suitable penalty factor for the given pipe network optimisation problem.

Computation of Total Costs

The total cost of each trial pipe network design in the current population is taken as the sum of the pipe network cost and the penalty cost.

Computation of the Fitness Values of Coded Strings

The fitness of a coded string representing a proposed pipe network design is a function of the total objective function cost. The pipe network optimisation problem is the search for the minimum cost pipe network configuration. Thus the objective function must be minimised, while the simple GA searches for the coded string with the highest fitness. The inverse of the total pipe network cost (the sum of pipe costs and penalty costs) is an example of a form of the fitness function selected to ensure highly fit (low cost) coded strings survive. It has the following form:

$$fitness_i = \frac{1}{cost_i} \quad \text{for coded strings } i = 1, \dots, N \quad (5.1)$$

Many other forms of fitness function may be selected, however the form of Eq. 5.1 was found to be effective in the GA search. The GA computes the fitness of each coded string in the current population.

Generation of a New Population of Coded Strings

The artificial GA evolution imitates rules of survival in nature and mechanisms of natural population genetics in an effort to reproduce the efficiency of the natural evolution processes. The traditional GA most commonly uses the three simple operators of proportionate selection (or reproduction), one-point crossover and bit-wise mutation. The GA operators apply selection rules and code modifications to the current (old) population of coded strings to generate a new population. The GA operators use probabilistic (randomised) transition rules, as opposed to deterministic rules, to move from population to population in the solution space.

The Selection Operator

Selection (or reproduction) is based on Darwin's survival-of-the-fittest philosophy of natural selection. The selection of a coded string from within a competing population to advance to the next generation of strings will depend on the fitness of the string compared to fellow strings and chance factors. The traditional selection operator is *proportionate selection* (Goldberg, 1989). The strings are selected randomly (with replacement) from the current population. Strings with higher fitnesses have a higher probability of being selected. The probability of selection, p_i of coded string i to advance to the new population of N strings using the proportionate selection method is given by Eq. 5.2.

$$p_i = \frac{\text{fitness}_i}{\sum_{j=1}^N \text{fitness}_j} \quad \text{for coded strings } i = 1, \dots, N \quad (5.2)$$

The Crossover Operator

Crossover is the partial exchange of binary bits between two parent strings to form two offspring strings. Parent strings selected from the old population (by the selection operator) mate in the traditional GA operator of one-point crossover with some specified probability of crossover, p_c . To perform one-point crossover, a bit position in the string (crossover site) is selected randomly and crossover occurs by breaking the coded strings and exchanging corresponding segments of code at bit positions after the crossover site.

The Mutation Operator

The traditional GA operator of bit-wise complement mutation is the occasional random alteration of binary bits to prevent the loss of a potentially useful genetic trait. The bit value of a randomly selected bit position of a randomly selected string (which has undergone crossover) is inverted (from 1 to 0 or from 0 to 1) with specified probability of mutation, p_m .

Production of Successive Generations

Goldberg (1989) refers to the application of the GA operators of selection, crossover and mutation as a standard genetic algorithm. The above procedure produces a new generation and the GA repeats the process to generate successive generations. The fittest strings (say the best 20) are stored and updated as fitter strings are generated. The size of the pipe network in terms of the number of decision variables determines the maximum number of generations which should be evaluated (typically a GA will evaluate 100 to 1000 generations). GAs do not necessarily guarantee that the global optimum solution will be reached, although experience indicates that they will give near-optimal solutions after a reasonable number of evaluations.

5.2 Case Study: The Two-Reservoir *Gessler* Pipe Network

In this chapter, a case study in pipe network optimisation is considered. A pipe network design problem described by Gessler (1985) is the chosen case study to investigate the application of genetic algorithms to the optimisation of water distribution pipe networks. The *Gessler* network has some interesting features including:

- the selection of the diameters of five new pipes
- three existing pipes may be cleaned, duplicated or left alone
- three demand patterns must be satisfied
- two supply sources are available

5.2.1 Description of the *Gessler* problem

The two-reservoir Gessler pipe network is a looped, gravity-fed water distribution system. The case study design problem involves an expansion to an existing pipe system. The layout of the existing pipe network and the proposed pipe network expansions are shown in Figure 5.1. The network consists of fourteen pipes, including five new pipes and nine existing pipes. The new pipes are to be sized and three of the existing pipes (the main supply lines) may be cleaned or duplicated with new parallel pipes. The pipe network connectivity and pipe hydraulic attributes are presented in Table 5.1. The Hazen-Williams formula is used to express pipe friction head loss in terms of pipe flow. US customary units used in the original study were preserved for this study.

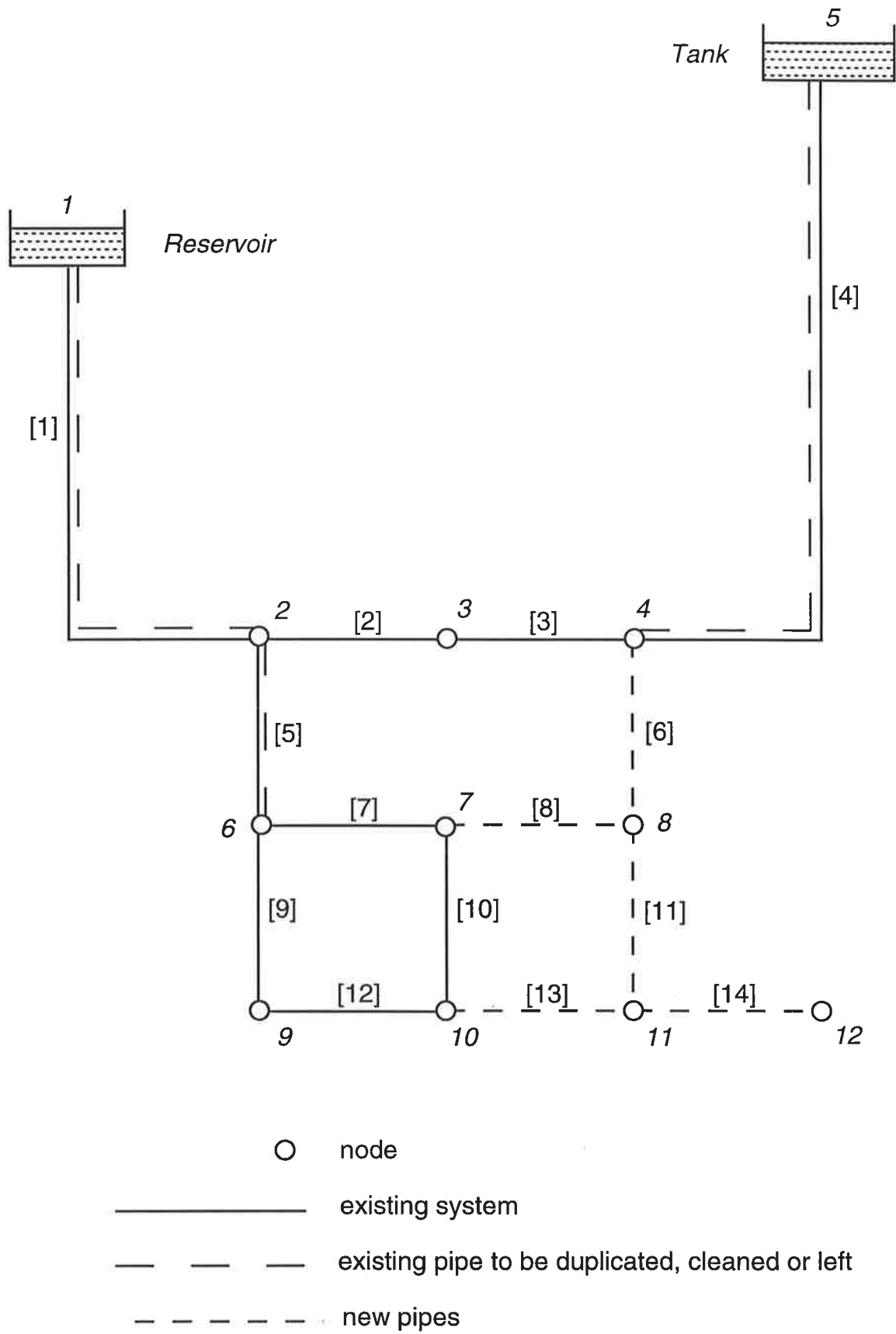


Figure 5.1 Layout of the Gessler network

Table 5.1 Pipe connectivity, lengths, diameters and roughness coefficients

Pipe	Start node	End node	Diameter (in)	Length (ft)	Hazen-Williams roughness, C
[1] [†]	1	2	14	15840	75 [#]
[2]	2	3	10	5280	80
[3]	3	4	10	5280	80
[4] [†]	4	5	10	21120	80 [#]
[5] [†]	2	6	10	5280	80 [#]
[6]	4	8	New	5280	120 [*]
[7]	6	7	8	5280	100
[8]	7	8	New	5280	120 [*]
[9]	6	9	10	5280	80
[10]	7	10	4	5280	100
[11]	8	11	New	5280	120 [*]
[12]	9	10	8	5280	100
[13]	10	11	New	5280	120 [*]
[14]	11	12	New	5280	120 [*]

[†] These existing pipes may be cleaned or duplicated

[#] A cleaned pipe will have a $C=120$

^{*} New pipes are assumed to have a $C=120$

The Gessler pipe network consists of a reservoir (node 1) and a tank (node 5) connected to ten demand nodes. Water distribution systems are designed to supply anticipated water demands to the nodes with satisfactory pressures. The Gessler pipe network is required to satisfy three instantaneous demand patterns. The three demand patterns designated GE1, GE2 and GE3, and corresponding minimum allowable pressure heads at the nodes and the node elevations are presented in Table 5.2. The demand pattern GE1 may represent a peak loading condition with relatively generous minimum pressures. The demand patterns GE2 and GE3 may represent emergency loading conditions such as fire fighting flows at nodes 7 and 12 respectively.

The set of available pipe sizes and the corresponding unit pipe costs and the unit costs for pipe cleaning are given in Table 5.3. The design problem is to determine the least cost pipe network expansions, such that the new system design performs adequately subject to the three expected loading conditions.

Table 5.2 Node elevations, demands and associated minimum pressures

Node	Elevation (ft)	Demand pattern GE1		Demand pattern GE2		Demand pattern GE3	
		Demand (gpm)	Minimum allowable pressure (psi)	Demand (gpm)	Minimum allowable pressure (psi)	Demand (gpm)	Minimum allowable pressure (psi)
1	1200	Reservoir					
2	1050	200	40	200	20	200	20
3	1070	200	25	200	20	200	20
4	1090	0	25	0	20	0	20
5	1220	Tank					
6	980	300	50	300	20	300	20
7	970	300	50	1300	15	300	20
8	960	300	50	300	20	300	20
9	950	200	50	200	20	200	20
10	950	300	50	300	20	300	20
11	960	300	50	300	20	300	20
12	950	200	50	200	20	800	15

Table 5.3 Available pipe sizes and associated costs

Diameter (in)	Cost of new pipe (\$/ft)	Cost of cleaning existing pipe (\$/ft)
6	15.1	14.5
8	19.3	15.7
10	28.9	16.8
12	40.5	17.7
14	52.1	18.5
16	59.4	19.2

5.2.2 Exhaustive enumeration of the Gessler problem

It was feasible to perform an exhaustive enumeration of every possible pipe network design for the Gessler problem. An exhaustive enumeration checks every possible combination of pipe sizes and identifies the global optimum design as the lowest cost design which satisfies the pressure constraints. There are six possible pipe sizes for the five new pipes and eight alternatives for the three existing pipes (including six possible duplicate pipe sizes, cleaning the pipe or 'do nothing'). Therefore, there are 3,981,312 (6^5 by 8^3) potential combinations of pipe sizes. Gessler (1985) used partial enumeration of a pruned search space of approximately 900 combinations to approximate the optimum design for this problem (Section 5.2.3).

5 Application of the traditional genetic algorithm to pipe network optimisation

The solution space was considered small enough such that an exhaustive enumeration could be performed to find the true minimum. The hydraulic network solver systematically executed the 3.981 million evaluations of the solution space (in 82 CPU hours on a SUN 4/280 computer for 11.944 million hydraulic simulations). An evaluation included a computation of the pipe network costs and a hydraulic analysis of the pipe network design for the three demand patterns to test for hydraulic feasibility.

The best 50 solutions to the Gessler problem are ranked in Table 5.4. The 50 pipe network designs in Table 5.4 satisfy the pressure constraints at the nodes for the three demand patterns. The lowest cost feasible designs or global optimum solutions at \$1.7503 million are solutions 1 and 2. Although 50 solutions represents only a fraction of the solution space, the cost of solution 50 is \$101,900 (or 5.82%) more than the cost of solution 1. The solutions reached by the genetic algorithm optimisation can be compared with this set of 50 solutions for some indication of the effectiveness of the search. In Table 5.4, “dup 14” denotes duplication of the existing pipe with a 14 inch diameter pipe, “clean” denotes cleaning the existing pipe and “leave” denotes leaving the existing pipe as it is (i.e., ‘do nothing’).

Some solution space statistics were determined in the process of the exhaustive enumeration. Of the 3.981 million possible solutions, 3.294 million are infeasible leaving only 687,500 feasible pipe network designs. Figure 5.2 shows the number of solutions with a given number of node pressure constraint violations. Notably, there are 233,500 solutions with the maximum number (30) of pressure violations. The demand pattern for which the maximum number of pressure violations occurred and the node with the most critical pressure for the infeasible solutions are shown in Figure 5.3. The demand pattern GE3 (the emergency loading at the extreme downstream node 12) is the most difficult to satisfy and the most critical nodes are nodes 4, 11 and 12.

Table 5.4 The best 50 solutions to the Gessler problem

No.	Cost (\$m)	Pipe Selections (inch diameter)							
		Existing pipes			New pipes				
		[1]	[4]	[5]	[6]	[8]	[11]	[13]	[14]
1	1.7503	leave	dup 14	leave	12	8	8	6	10
2	1.7503	leave	dup 14	leave	12	8	10	6	8
3	1.7725	leave	dup 14	leave	12	8	8	8	10
4	1.7725	leave	dup 14	leave	12	8	10	8	8
5	1.7910	leave	dup 14	dup 8	10	8	8	6	10
6	1.7999	leave	dup 14	clean	10	8	8	8	10
7	1.8010	leave	dup 14	leave	12	8	10	6	10
8	1.8010	leave	dup 14	leave	12	10	8	6	10
9	1.8010	leave	dup 14	leave	12	10	10	6	8
10	1.8115	leave	dup 14	leave	14	8	8	6	10
11	1.8115	leave	dup 14	leave	14	8	10	6	8
12	1.8115	leave	dup 14	leave	12	8	12	6	8
13	1.8115	leave	dup 14	leave	12	8	8	6	12
14	1.8131	leave	dup 14	dup 8	10	8	8	8	10
15	1.8232	leave	dup 14	leave	12	8	8	10	10
16	1.8232	leave	dup 14	leave	12	8	10	8	10
17	1.8232	leave	dup 14	leave	12	10	8	8	10
18	1.8232	leave	dup 14	leave	12	8	10	10	8
19	1.8232	leave	dup 14	leave	12	10	10	8	8
20	1.8285	leave	dup 14	clean	10	8	10	6	10
21	1.8300	leave	dup 14	dup 6	12	8	8	6	10
22	1.8300	leave	dup 14	dup 6	12	8	10	6	8
23*	1.8337	leave	dup 14	leave	12	8	12	8	8
24	1.8337	leave	dup 14	leave	14	8	10	8	8
25	1.8337	leave	dup 14	leave	14	8	8	8	10
26	1.8337	leave	dup 14	leave	12	8	8	8	12
27	1.8385	clean	dup 12	leave	10	10	10	6	10
28	1.8390	clean	dup 12	dup 8	10	8	8	6	10
29	1.8390	leave	dup 14	clean	12	8	8	6	10
30	1.8390	leave	dup 14	clean	12	8	10	6	8
31	1.8390	leave	dup 14	clean	10	8	12	6	8
32	1.8390	clean	dup 12	dup 6	10	8	8	8	10
33	1.8390	clean	dup 12	dup 8	10	8	10	6	8
34	1.8417	leave	dup 14	dup 8	10	8	10	6	10
35	1.8417	leave	dup 14	dup 10	10	8	8	6	10
36	1.8417	leave	dup 14	dup 10	10	8	10	6	8
37	1.8417	leave	dup 14	dup 8	10	10	8	6	10
38	1.8417	leave	dup 14	dup 6	10	10	8	8	10
39	1.8480	clean	dup 12	clean	10	8	8	8	10
40	1.8501	leave	dup 14	leave	16	8	8	6	10
41	1.8501	leave	dup 14	leave	16	8	10	6	8
42	1.8506	leave	dup 14	clean	10	8	8	10	10
43	1.8506	leave	dup 14	clean	10	8	10	8	10
44	1.8506	leave	dup 14	clean	10	10	8	8	10
45	1.8517	leave	dup 14	leave	12	10	10	6	10
46	1.8517	leave	dup 14	leave	10	12	10	6	10
47	1.8517	leave	dup 14	leave	10	10	12	6	10
48	1.8517	leave	dup 14	leave	10	10	10	6	12
49	1.8522	leave	dup 14	dup 8	12	8	8	6	10
50	1.8522	leave	dup 14	dup 6	12	8	8	8	10

* Gessler's best solution

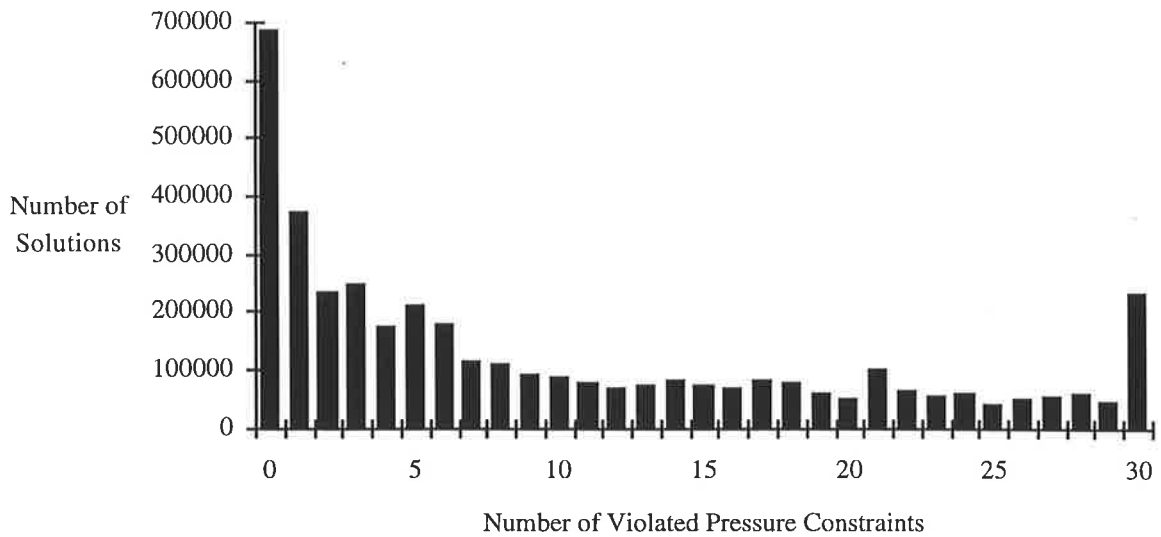


Figure 5.2 Number of solutions with a particular number of nodes at which pressure constraints are violated

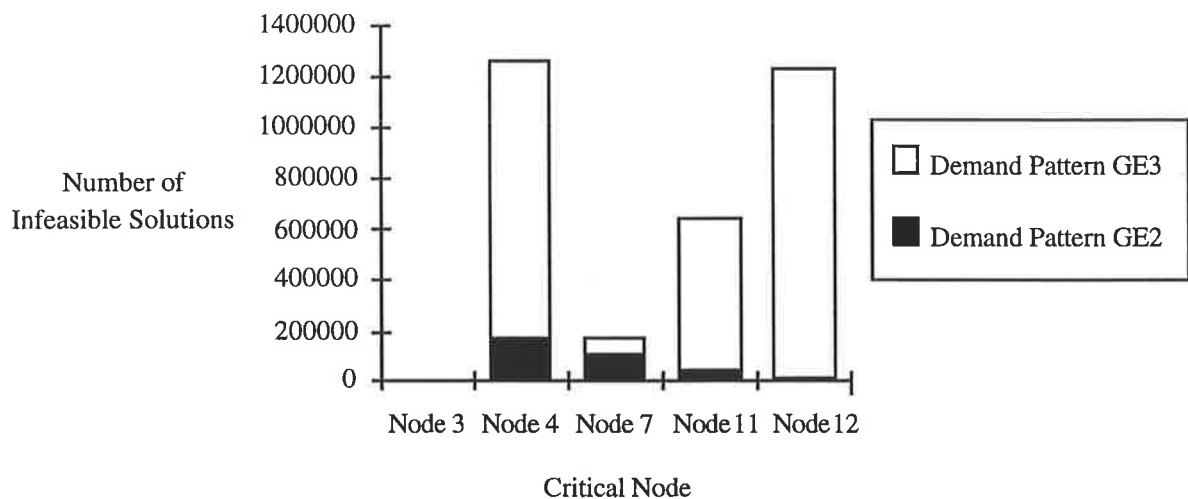


Figure 5.3 Critical demand patterns and critical nodes

The feasibility of solutions with respect to the alternative decisions for the existing pipe [1] and the new pipe [6] is shown in Figures 5.4 and 5.5 respectively. There is clearly a dominance of infeasible solutions. There are no feasible solutions which select a 6 inch diameter pipe for the new pipe [6]. Similarly, there are no feasible solutions which select a 6 inch diameter pipe for the new pipe [14] and no feasible solutions which choose to ‘do nothing’ with the existing pipe [4].

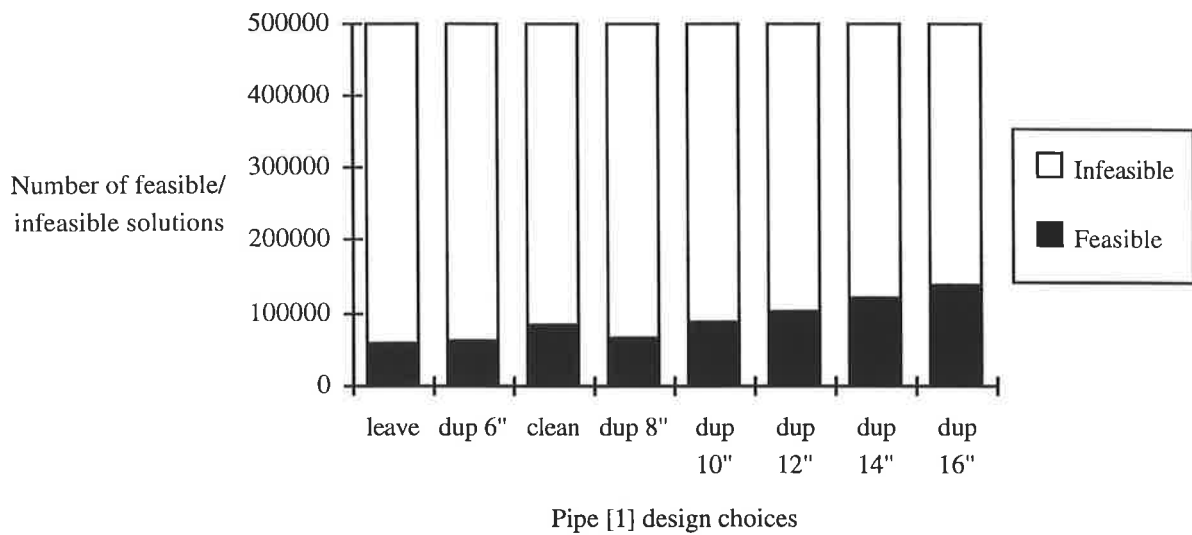


Figure 5.4 Feasible solutions when implementing pipe [1] rehabilitation decisions

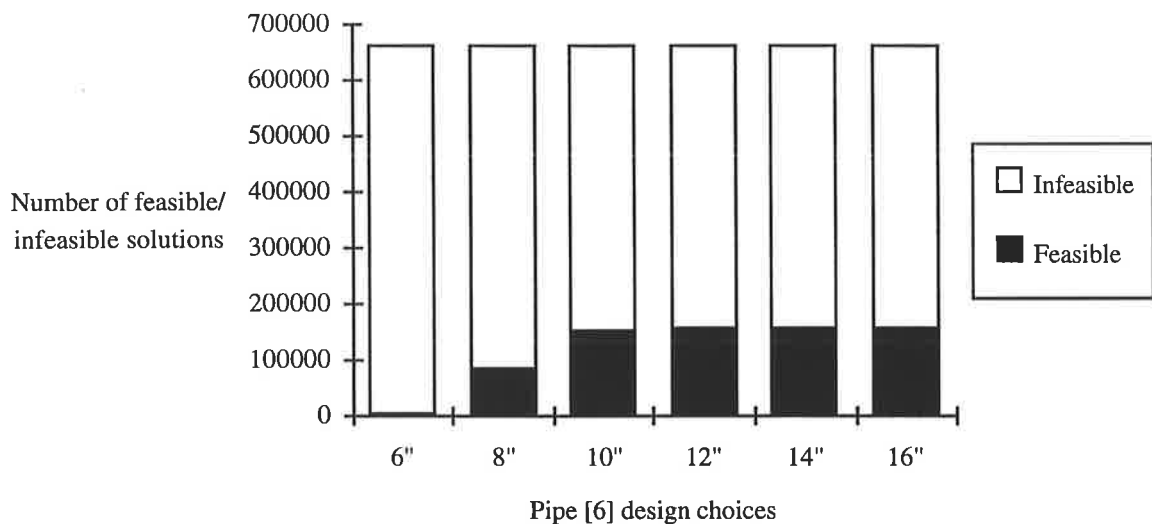


Figure 5.5 Feasible solutions when implementing decisions for new pipe [6]

The feasibility of solutions with respect to pipe network expansion costs is shown in Figure 5.6. There are no feasible solutions with pipe network costs less than \$1.75 million (the two global optimal solutions cost \$1.7503 million). There are only 1,096 feasible solutions with pipe network costs less than \$2.0 million. The minimum pipe network cost is \$0.3986 million (infeasible), the maximum pipe network cost is \$4.0772 million and the average pipe network cost across the entire solution space is \$2.1762 million.

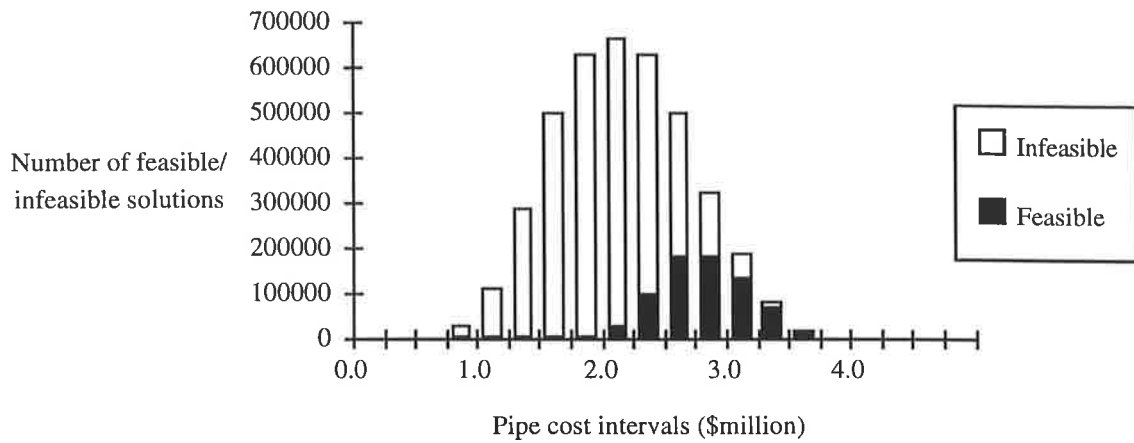


Figure 5.6 The feasibility of solutions with given pipe network costs

The pipe flow and node pressure head distributions for the optimum solution 1 (in Table 5.4) for the three demand patterns are supplied in Tables 5.5 and 5.6 respectively. The direction of the flows in the pipes in Table 5.6 is determined from the node connectivity data in Table 5.1.

Table 5.5 The balanced pipe flows for global optimal solution 1

Pipe	Demand pattern GE1		Demand pattern GE2		Demand pattern GE3	
	Head Loss (ft)	Flow (cfs)	Head Loss (ft)	Flow (cfs)	Head Loss (ft)	Flow (cfs)
[1]	30.974	1.676	68.143	2.565	50.000	2.170
[2]	-0.932	-0.201	-1.538	-0.264	-0.457	-0.137
[3]	-8.093	-0.647	-9.601	-0.709	-6.667	-0.583
[4]	-41.949	-3.448	-77.004	-4.786	-62.876	-4.290
[5]	35.238	1.432	90.579	2.384	57.323	1.862
[6]	23.719	2.801	47.531	4.077	39.865	3.707
[7]	-0.434	-0.093	29.456	0.903	-0.485	-0.098
[8]	-20.110	-0.882	-83.643	-1.904	-24.096	-0.973
[9]	13.593	0.856	12.325	0.812	29.132	1.292
[10]	20.861	0.121	-11.594	-0.088	55.721	0.206
[11]	38.385	1.251	54.024	1.504	97.306	2.067
[12]	6.833	0.410	5.536	0.366	26.104	0.846
[13]	-2.586	-0.137	-18.025	-0.390	17.489	0.384
[14]	1.914	0.446	1.914	0.446	24.949	1.782

Table 5.6 Junction node pressure heads for global optimal solution 1

Node	Demand pattern GE1		Demand pattern GE2		Demand Pattern GE3	
	Minimum allowable pressure (psi)	Actual pressure (psi)	Minimum allowable pressure (psi)	Actual pressure (psi)	Minimum allowable pressure (psi)	Actual pressure (psi)
2	40	51.50	20	35.42	20	43.27
3	25	43.25	20	27.43	20	34.81
4	25	38.10	20	22.93	20	29.04
6	50	66.54	20	26.51	20	48.76
7	50	71.06	15	18.10	20	53.29
8	50	84.09	20	58.62	20	68.05
9	50	73.64	20	34.16	20	49.13
10	50	70.69	20	31.77	20	37.84
11	50	67.48	20	35.24	20	25.94
12	50	70.98	20	38.74	15	19.47

5.2.3 Partial enumeration applied by Gessler (1985)

Gessler (1985) solved the optimisation problem by a partial enumeration of a limited solution space of 900 combinations of the design parameters. Gessler pruned the solution space to 900 by placing the pipes into five pipe groups as shown in Table 5.7. The pipes in a group were restricted to selecting the same design parameter from the candidate design parameters in Table 5.7.

Table 5.7 Pipe groups (as used by Gessler, 1985)

Pipe group	Pipes in group	Candidate design parameters (inches)
1	[1] and [5]	leave, clean, dup 12, dup 14, dup 16
2	[4]	leave, clean, dup 12, dup 14, dup 16
3	[6] and [11]	8, 10, 12
4	[8] and [13]	6, 8, 10
5	[14]	6, 8, 10, 12

A further 495 combinations were eliminated since the 6" pipe for pipe group 5 and the leave and clean alternatives for pipe group 2 were not considered. According to the exhaustive enumeration carried out in the research presented in this thesis:

- there are no feasible solutions with a new 6" pipe [14]
- there are no feasible solutions for leaving the existing pipe [4]
- and there are only 2,421 feasible solutions for cleaning the existing pipe [4].

The pruning of the solution space for a partial enumeration must be based on experience. Unfortunately, the global optimum may be eliminated in this process. Gessler excluded the global optimum by restricting pipes [6] and [11] in pipe group 3 and pipes [8] and [13] in pipe group 4 to the same pipe size.

Gessler enumerated the remaining 405 combinations. A hydraulic analysis to determine the pressure head pattern was required for only 6% of the 405 combinations. The other 94% were excluded from the search during the ‘cost filter’ and ‘size filter’ described in Section 3.3.6. Gessler achieved a best solution with a cost of \$1.8337 million represented by solution 23 in Table 5.4.

5.2.4 Nonlinear programming optimisation of the Gessler problem

The Gessler case study problem was solved by Simpson et al. (1994) using the nonlinear optimisation package GINO assuming that continuous sizes were available for all pipes. The GINO analysis was carried out independently of the exhaustive enumeration.

A cost function for new pipes was fitted to the data in Table 5.3. The cost per unit length (c_A) was approximately linear with pipe diameter for new pipes. The line fitted to the new pipe cost data is only approximate, especially for the smallest pipe size. The process of fitting a cost function to the data is a weakness of the application of nonlinear optimisation to pipe networks. Table 5.8 shows the fitted cost functions for new and duplicated pipes.

The ‘equivalent diameter’ approach was used to model the effects of cleaning or duplicating existing pipes [1], [4] and [5] (i.e., each pipe was assumed to be replaced by a new pipe with the same Hazen-Williams coefficient as the old pipe but a new equivalent diameter). Thus the equivalent diameter must be greater than or equal to the existing diameter. The cost per unit length (c_B for existing pipe [1] and c_C for existing pipes [4] and [5]) was approximately linear with the logarithm of equivalent diameter D_e for these pipes (Table 5.8).

Table 5.8 Fitted cost functions for the nonlinear optimisation model

Pipes	Cost function (\$/ft length)
New	$c_A = 4.730 D - 16.124$
Pipe [1]	$c_B = 128.05 \log(D_e) - 67.735$ (for $D_e \geq 14''$)
Pipes [4] & [5]	$c_C = 84.826 \log(D_e) - 17.963$ (for $D_e \geq 10''$)

Note: D = diameter of new pipe (inches)

D_e = equivalent diameter of pipe (inches)

5 Application of the traditional genetic algorithm to pipe network optimisation

The variables in the GINO model are the diameters (D_{new}) of the five new pipes, the equivalent diameters (D_{new}) of the three existing pipes [1], [4] and [5], the 14 flows (Q_k) in each pipe for each loading case and the 10 total heads (H_i) at each junction node for each loading case. This gives a total of 80 variables.

The constraints for each of the three loading cases and additional minimum diameter constraints are as follows:

- (i) Continuity of flow at each node (10 linear equations).

$$\sum_{k=1}^{NPJ} Q_k + Q_{ex_i} = 0 \quad \text{for all junction nodes } i = 1, \dots, NJ \quad (5.3)$$

where Q_k are the flows in each of the NPJ pipes connected to node i if a convention is adopted such that a flow away from the node is taken as positive. The demand at the node is Q_{ex_i} .

- (ii) The Hazen-Williams head loss equation for each pipe k connecting nodes i and j (14 nonlinear equations).

$$H_i - H_j = \frac{8.528 \times 10^5 L_k Q_k |Q_k|^{0.852}}{C_k^{1.852} D_k^{4.8704}} \quad \text{for all pipes } k = 1, \dots, NP \quad (5.4)$$

where L_k is the length of pipe k , and C_k is the Hazen-Williams coefficient for pipe k .

- (iii) Minimum pressure head constraints at each node as given in Table 5.2 (10 linear inequalities).

$$H_i \geq \overline{H}_i \quad \text{for all junction nodes } i = 1, \dots, NJ \quad (5.5)$$

The items (i) to (iii) give 34 constraints per loading case.

- (iv) In addition there are 8 lower bounds for the pipe diameters.

$$D_{new} \geq \overline{D}_{new} \quad \text{for all new pipes } new=1, \dots, NEW_{total} \quad (5.6)$$

The minimum diameter for the five new pipes in the case study problem is 6" (Table 5.3). The minimum equivalent diameters for pipes which could be duplicated (pipes [1], [4], [5]) are the existing diameters (Table 5.1). Upper bounds on pipe diameters were not required for this problem.

5 Application of the traditional genetic algorithm to pipe network optimisation

There were a total of 110 constraints for the problem (i.e. $3 \times 34 + 8$) as a result of the three loading cases being considered simultaneously. The objective function (to be minimised) is the total cost of the network, that is, the cost per unit length for each pipe (Table 5.8) multiplied by its length and summed over all pipes in the network. The following objective function was optimised in GINO:

$$\text{Minimise cost} \quad c_B L_{[1]} + \sum_{k \in [4],[5]} (c_C L_k) + \sum_{k \in [6],[8],[11],[13],[14]} (c_A L_k) \quad (5.7)$$

where c_A , c_B , and c_C are the pipe cost functions in Table 5.8.

GINO uses a generalised reduced gradient approach for solving nonlinear optimisation problems. Liebman et al. (1986) discusses the details of the method. Table 5.9 shows the optimum solution obtained with an estimated cost of \$1.76 million. The continuous diameters need to be rounded up or down to the nearest discrete pipe size. This may involve considerable judgement, particularly for a large network as it is not clear that the solution after rounding will automatically satisfy the minimum pressure constraints at all nodes. In this case, the rounding was carried out by trial and error with each rounded solution being checked to see if it satisfied the minimum pressure constraints for the three loading cases.

Table 5.9 Solution from GINO nonlinear optimisation for the Gessler network

Pipe	Existing diameter (inches)	Diameter from nonlinear optimisation (inches)	Rounded solution (Equivalent diameter, inches and/or whether rounded up or down)
[1]	14	14 (existing)	Don't Duplicate
[4]	10	17.52	Duplicate with 14" (17.91", round up)
[5]	10	10 (existing)	Don't Duplicate
[6]	-	11.69	12" (round up)
[8]	-	8.07	8" (round down)
[11]	-	8.78	10" (round up)
[13]	-	6 (minimum)	6"
[14]	-	8.66	10" (round up)
COST=		\$1.76 million	\$1.8010 million

The rounded solution determined for this problem is shown in Table 5.9. The duplication of pipe [4] with a 14" diameter pipe gives an equivalent diameter of 17.91" that is slightly larger than the optimum continuous size of 17.52". Four pipes have been rounded up in size, one has been rounded down and three do not require rounding as they are at the respective lower bounds. The cost of this rounded solution is \$1.8010 million. This solution corresponds to solution 7 in Table 5.4 and is about 3% more expensive than the optimum.

5 Application of the traditional genetic algorithm to pipe network optimisation

The nonlinear programming model was modified to calculate the pressure heads throughout the network for the particular set of pipe sizes. This was accomplished by replacing the lower bounds on pipe diameters by equations which define the diameters and deleting the minimum head constraints. The rounded solution was verified by a commercial hydraulic computer simulation program (WATER, Fowler, 1990).

The most significant rounding in this solution was for pipes [11] and [14] whose sizes fell midway between the commercial sizes of 8" and 10". In the above solution both were rounded up in size. A further three alternative rounding solutions were tested:

- (1) round both pipes [11] and [14] down to 8" diameter
- (2) round pipe [11] down to 8" and pipe [14] up to 10"
- (3) round pipe [11] up to 10" and pipe [14] down to 8"

Solutions (2) and (3) above both satisfied the minimum pressure constraints but solution (1) violated the constraints. The costs of solutions (2) and (3) are identical and both equal to \$1.7503 million. These solutions were identified as the likely optimum solutions to the problem. Comparison with the solutions obtained by exhaustive enumeration (Table 5.4) shows that the two solutions obtained by nonlinear optimisation are indeed the minimum cost solutions to the problem.

Each optimisation run took approximately 6.8 CPU minutes on a SUN 4/280 for this problem. Identification of the global optima after rounding in this case could have been fortuitous as there is no guarantee that rounding from a continuous solution will give the optimum discrete sizes.

5.3 A Small Scale Simulation of the Simple Genetic Algorithm applied to the Gessler Problem

A simple three-operator genetic algorithm (GA) using a population of ten coded string solutions is applied to the Gessler pipe network optimisation problem. The GA formulation, GA operations and string decoding and evaluation procedures are presented in detail to illustrate the application.

5.3.1 Coded strings representing Gessler network designs

The set of decision variables describing a trial solution to the pipe network optimisation is represented by a unique coded string of binary bits. A trial solution to the Gessler problem includes the sizing of five new pipes ([6], [8], [11], [13] and [14]) and the choice to clean, duplicate or 'do nothing' with three existing pipes ([1], [4] and [5]). Thus, there are eight decision variables and the coded string is made up of eight binary-coded substrings.

A list of the discrete choices for the decision variables is presented in Table 5.10. The 18" and 20" pipe sizes for new pipes were added to the list to make up eight alternatives for each decision variable for the genetic algorithm formulation. The order of the list of decision variable choices for the existing pipes in Table 5.10 is chosen in terms of increasing unit pipe cost (\$/ft) and increasing equivalent pipe diameter (in).

Table 5.10 Decision variable choices and corresponding binary substrings

Decision variable choices		Binary coded substring
If pipe exists ([1], [4] and [5])	If pipe is new ([6], [8], [11], [13] and [14])	
leave as exists	6"	000
duplicate with 6"	8"	001
clean existing pipe	10"	010
duplicate with 8"	12"	011
duplicate with 10"	14"	100
duplicate with 12"	16"	101
duplicate with 14"	18"	110
duplicate with 16"	20"	111

A trial Gessler network solution is represented by a binary string according to the mapping between discrete choices for the decision variables and binary-coded substrings in Table 5.10. Since there are eight decision variable choices, a 3-bit binary substring is appropriate. A 24-bit string which represents a Gessler network solution can be constructed from eight concatenated 3-bit decision-variable substrings as shown in Figure 5.7.

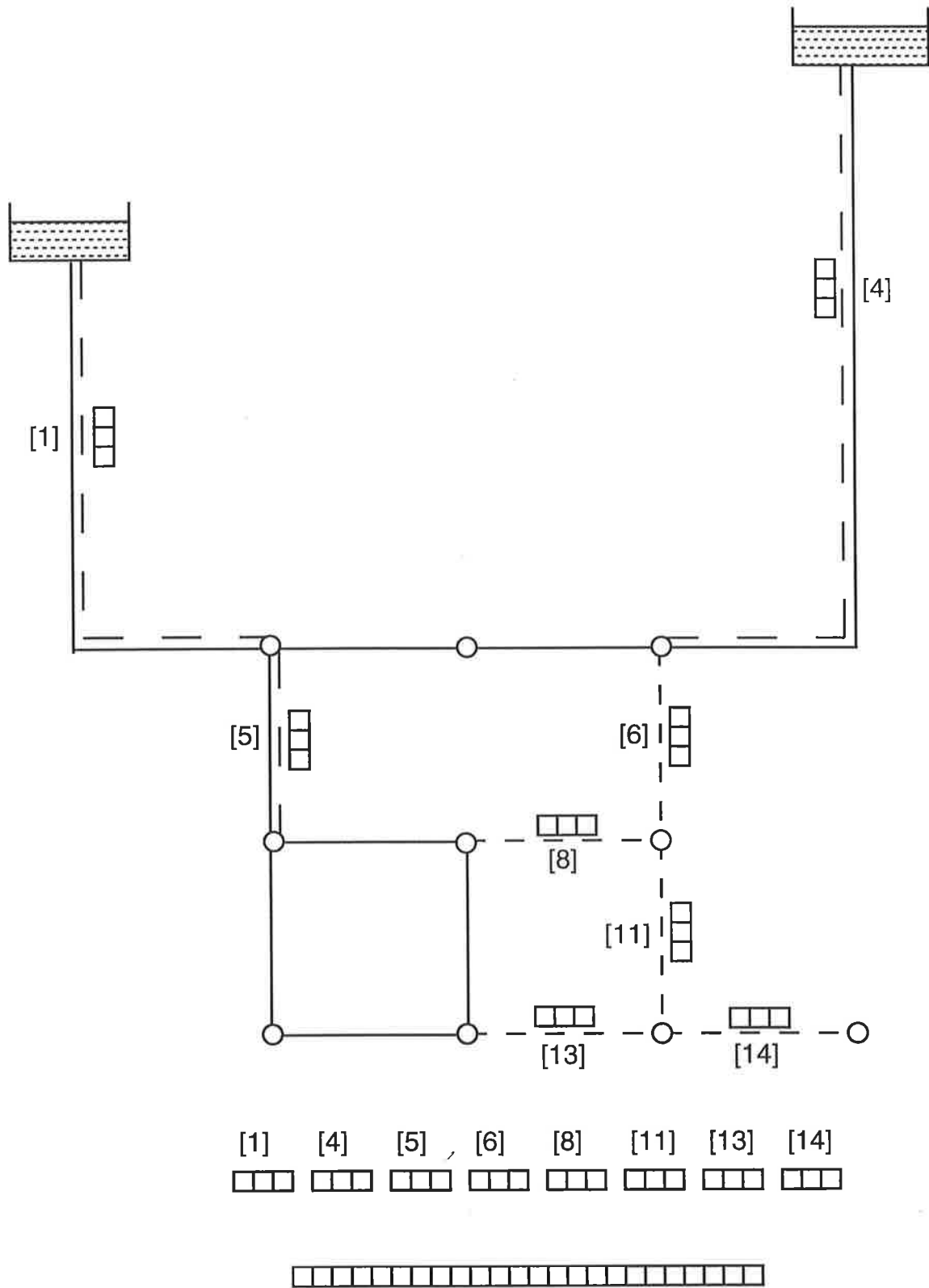


Figure 5.7 The formation of a 24-bit coded string

The 3-bit binary substrings may decode to a zero pipe size for existing pipes such that proposed duplicate pipes may be eliminated from the design. The genetic algorithm can thus be used to consider the parts of the existing pipe system which require reinforcement with duplicate pipes. The GA could be used in a similar way to determine the optimum layout of new pipes. Often new pipes cannot be eliminated and the new pipes are constrained to a minimum diameter (this is the case for the Gessler pipe network design).

5.3.2 Coded string populations

GAs search the solution space using a population of points. A point in the GA solution space is a unique 24-bit coded string. GAs move in the solution space by successively evaluating and regenerating populations of coded strings. The GA search begins by randomly generating a starting population of strings. Random unbiased generation of 1's and 0's to form the bits of the strings produced the starting population of ten coded strings shown in Table 5.11. The strings in the starting population are randomly distributed in the solution space. The solution space searched by the GA for this problem contains 16,777,216 points since there are 2^{24} unique 24-bit binary strings.

Table 5.11 The starting population

No.	Coded string
1	100-101-101-011-111-111-011-000
2	101-100-000-100-110-111-110-101
3	001-010-111-111-111-100-010-011
4	010-001-001-011-111-100-101-010
5	101-101-001-111-110-010-101-010
6	110-001-000-110-111-111-010-000
7	000-100-001-010-111-101-011-001
8	100-001-010-110-110-111-110-110
9	010-011-110-001-100-110-010-010
10	010-010-010-110-101-100-010-101

5.3.3 Evaluation of the coded strings

In turn, the binary strings in the current population are decoded to trial pipe network solutions and each solution is evaluated. Consider the evaluation of String 9 of the starting population. Each 3-bit decision-variable substring corresponds to a pipe in the network to be sized or reinforced as shown below:

Binary substrings of String 9: 010 011 110 001 100 110 010 010
 Pipe corresponding to substring position: [1] [4] [5] [6] [8] [11] [13] [14]

Strings are decoded by observing the mapping between substrings and decision variable choices in Table 5.10. String 9 specifies the pipe network design in Table 5.12. The implications of the decisions on existing diameters and Hazen-Williams roughness coefficients of the pipes in the network are summarised in Table 5.13.

Table 5.12 Decoding String 9

Substring	Decision
010	clean existing pipe [1]
011	duplicate existing pipe [4] with an 8" pipe
110	duplicate existing pipe [5] with a 14" pipe
001	new pipe [6] is an 8" pipe
100	new pipe [8] is a 14" pipe
110	new pipe [11] is an 18" pipe
010	new pipe [13] is a 10" pipe
010	new pipe [14] is a 10" pipe

Table 5.13 Equivalent diameters and Hazen-Williams (H-W) coefficients for the network solution represented by String 9

Pipe	Existing diameter (in)	Duplicated or new diameter (in)	Equivalent diameter (in)	Existing H-W coefficient C	New H-W coefficient C
[1]	14	0	14	75	120
[2]	10	0	10	80	80
[3]	10	0	10	80	80
[4]	10	8	10.78	80	120
[5]	10	14	15.33	80	120
[6]	0	8	8	0	120
[7]	8	0	8	100	100
[8]	0	14	14	0	120
[9]	10	0	10	80	80
[10]	4	0	4	100	100
[11]	0	18	18	0	120
[12]	8	0	8	100	100
[13]	0	10	10	0	120
[14]	0	10	10	0	120

The existing pipes [2], [3], [7], [9], [10] and [12] are unchanged. The existing pipe [1] is cleaned and hence the Hazen-Williams roughness coefficient of $C=75$ is increased to $C=120$. The existing pipe [4] is duplicated with an 8" diameter pipe and the two parallel pipes are equivalent to a 10.78" diameter pipe with a $C=120$ of the same length. Existing pipe [5] is duplicated with a 14" diameter pipe. New pipes [6], [8], [11], [13] and [14] are sized. The new pipes are assumed to have a $C=120$.

5 Application of the traditional genetic algorithm to pipe network optimisation

The pipe network design proposed by String 9 is now established and the fitness of the solution can be evaluated. The decisions for a proposed network solution are associated with unit pipe material costs or unit pipe cleaning costs given in Table 5.3. The pipe cost for a decision (\$) is the product of the unit pipe cost of the decision (\$/ft) and the length of the pipe (ft). The pipe costs for String 9 are summarised in Table 5.14. The lengths of pipes [1] and [4] make the pipe costs expensive and the significant positions of pipes [1] and [4] in the network (main supply lines) make the decisions very influential on network hydraulic performance. The total pipe cost for String 9 is \$2.0328 million.

Table 5.14 Determining the pipe cost of String 9

Pipe	Decision	Unit pipe cost (\$/ft)	Length of pipe (ft)	Pipe cost (\$ million)
[1]	clean	18.5	15840	0.293
[4]	duplicate 8"	19.3	21120	0.408
[5]	duplicate 14"	52.1	5280	0.275
[6]	8"	19.3	5280	0.102
[8]	14"	52.1	5280	0.275
[11]	18"	70.5	5280	0.375
[13]	10"	28.9	5280	0.153
[14]	10"	28.9	5280	0.153
Total pipe cost=				2.0328

The capacity of the network represented by the coded string to meet the system performance constraints is assessed. Hydraulic simulations of the pipe network are performed to determine balanced pipe flows and the corresponding pressure heads at the junction nodes for each of the demand patterns given in Table 5.2. The Newton-Raphson iterative technique applied to the simultaneous nonlinear loop corrective flow equations was used to determine the balanced pipe flows (Chapter 2). The Hazen-Williams friction head loss formula (Eqs. 2.3-2.5) is used to compute the head loss in the pipe sections. Table 5.15 provides the balanced pipe flows and the corresponding head loss in the pipe sections for the three demand patterns.

The node connectivity in Table 5.1 may be consulted to determine the direction of flow and head loss in the pipe sections in Table 5.15. The head losses in the supply line pipes [1], [4], [6] and [7] are significant for each demand pattern. The losses throughout the network are highest for demand pattern GE2. Demand pattern GE2 simulates an emergency fire fighting demand at node 7.

Table 5.15 The balanced pipe flows

Pipe	Demand pattern GE1		Demand pattern GE2		Demand pattern GE3	
	Head loss (ft)	Flow (cfs)	Head loss (ft)	Flow (cfs)	Head loss (ft)	Flow (cfs)
[1]	47.684	3.389	96.561	4.963	75.174	4.335
[2]	0.352	0.119	2.522	0.344	1.443	0.255
[3]	-2.290	-0.327	-0.262	-0.101	-0.847	-0.191
[4]	-65.794	-1.735	-118.920	-2.389	-95.845	-2.126
[5]	7.293	2.825	15.010	4.173	11.624	3.634
[6]	47.765	1.408	117.246	2.287	86.021	1.935
[7]	38.313	1.041	106.295	1.807	73.945	1.486
[8]	0.240	0.351	-1.777	-1.037	1.072	0.789
[9]	22.189	1.115	48.238	1.697	37.472	1.481
[10]	0.870	0.022	-4.326	-0.052	1.426	0.028
[11]	0.574	1.091	0.180	0.582	1.853	2.055
[12]	16.941	0.670	53.855	1.252	37.895	1.035
[13]	0.008	0.023	2.659	0.532	1.535	0.395
[14]	1.918	0.446	1.918	0.446	24.920	1.782

Since the elevation of the source nodes 1 and 5 are known, the pressure heads at the junction nodes throughout the network can be computed. The actual node pressure heads are compared with the minimum allowable pressure heads for the three demand patterns in Table 5.16. The junction nodes in the network where the pressure head is insufficient are shown in Table 5.16 (shaded). String 9 represents an infeasible pipe network solution, since it does not meet the minimum pressure head requirements at the nodes while supplying demand patterns GE2 and GE3. String 9 fails at node 4 for demand pattern GE3 and at several junction nodes for demand pattern GE2. The minimum allowable pressure heads are satisfied for demand pattern GE1.

Table 5.16 Comparison of actual and allowable node pressure heads

Node	Demand pattern GE1		Demand pattern GE2		Demand Pattern GE3	
	Required pressure (psi)	Actual pressure (psi)	Required pressure (psi)	Actual pressure (psi)	Required pressure (psi)	Actual pressure (psi)
2	40	44.27	20	23.12	20	32.38
3	25	35.47	20	13.34	20	23.10
4	25	27.78	20	4.79	20	14.78
6	50	71.41	20	46.92	20	57.64
7	50	59.15	15	5.22	20	29.97
8	50	63.38	20	10.31	20	33.83
9	50	74.79	20	39.03	20	54.40
10	50	67.45	20	15.72	20	38.01
11	50	63.12	20	10.24	20	33.02
12	50	66.62	20	13.74	15	26.56

5 Application of the traditional genetic algorithm to pipe network optimisation

A penalty cost is added to the pipe costs for infeasible solutions to reduce the fitness of the string. The penalty cost is taken as a function of the degree of violation of the pressure constraints. The penalty function used in this formulation considers the pressure heads at the nodes and compares this to the minimum allowable pressure heads to find the maximum violation of the pressure constraints for each demand pattern. The penalty cost (\$) for each demand pattern is taken as a product of the maximum node pressure deficiency (psi) and a chosen penalty factor, k (\$/psi). A value of $k = \$50,000/\text{psi}$ for the penalty factor was chosen for the study of the Gessler problem. The total penalty cost is the sum of these component penalty costs for each demand loading case. The computation of penalty cost for String 9 is summarised in Table 5.17. The total penalty cost for String 9 is \$1.0215 million.

Table 5.17 Calculation of the penalty cost for String 9

Demand pattern	Critical pressure deficiency (psi)	Critical node	Penalty multiplier, k (\$/psi)	Penalty cost (\$ million)
GE1	None	-	50,000	0
GE2	15.21	4	50,000	0.7605
GE3	5.22	4	50,000	0.2610
Total	20.43			1.0215

The total cost of the string is the sum of the pipe costs and the penalty costs. The total cost of String 9 is \$3.0541 million. The objective of the optimisation is to find the network solution with the lowest total cost. The string is accompanied by a fitness taken as some function of the total cost of the string. The GA searches for the string with the highest fitness. A simple inverse fitness function (Eq. 5.1) may be used to calculate string fitness from total network cost. The fitness of String 9 is 0.3274. The procedure described above is used to decode and evaluate the fitness of each string in the starting population. The total costs and the fitness values of the strings in the starting population are summarised in Table 5.18.

Table 5.18 Cost and fitness of the strings in the starting population

String	Pipe cost (\$ million)	Penalty cost (\$ million)	Total cost (\$ million)	Fitness
1	2.8897	3.4810	6.3708	0.1570
2	3.0180	1.0847	4.1027	0.2437
3	2.4045	5.5427	7.9472	0.1258
4	2.0745	4.3469	6.4214	0.1557
5	2.9980	0.0806	3.0786	0.3248
6	2.6067	7.1571	9.7638	0.1024
7	1.8997	2.5152	4.4149	0.2265
8	2.7926	4.5689	7.3615	0.1358
9	2.0328	1.0213	3.0541	0.3274
10	2.1664	4.2231	6.3895	0.1565

Some population cost statistics for the starting population are presented in Table 5.19. The starting population contains no feasible solutions and penalty costs comprise the major part of the total cost. The best generation cost for the starting population of 10 randomly generated strings is \$3.0541 million for String 9. The best generation cost (highest fitness and lowest cost) for String 9 is quite different from the optimum cost of \$1.7503 million identified by the complete enumeration. The starting population does not contain any feasible solutions, however, the small population size of 10 strings used for this experiment is unreasonably small for a problem of this size.

Table 5.19 Starting population cost statistics

Generation cost statistics	
Lowest cost (String 9)	\$3.0541 million
Highest cost (String 6)	\$9.7638 million
Average generation cost	\$5.8905 million
Average pipe cost	\$2.4883 million
Average penalty cost	\$3.4022 million
Number of infeasible solutions	10

5.3.4 The second generation of coded strings

The second generation is generated from the starting population using a series of GA operators. The traditional three-operator GA applies the GA operators of proportionate selection, one-point crossover and bit-wise mutations to the old population to produce the new population.

5.3.5 Selection

Selection (or reproduction) uses a combination of the fitness information and random chance to select pairs of parent strings to form the next generation. The GA gives strings with higher fitness relative to the other strings a better chance of selection in the reproduction process.

Goldberg (1989) likened proportionate selection to a weighted roulette wheel with slots sized according to fitness. A casino roulette wheel has 37 segments of equal size and the ball may fall into any one of the segments with equal probability. The number of segments of the weighted roulette wheel equals the population size (in this case 10). Each string in the population is assigned a segment and the size of the segment is proportional to the fitness of the string. The ball may fall into any segment but there is greater probability the ball will select the segment of a highly fit string. The probability of selection in reproduction is given by Eq. 5.2.

The probability of selection by proportionate reproduction of the strings in the starting population has been computed and is given in Table 5.20. The strings are assigned intervals between 0.0 and 1.0 proportional to the probabilities of selection by the cumulative distribution in Table 5.20. A string is selected by generating a random number between 0.0 and 1.0 and looking up the interval in which this random number falls. For example, if the number 0.1206 is generated randomly, String 2 is selected.

Table 5.20 The probability of survival

String	Fitness	Probability of selection (%)	Cumulative probability distribution	Copies
1	0.1570	8.03	0.0803	-
2	0.2437	12.46	0.2049	2
3	0.1258	6.43	0.2692	-
4	0.1557	7.96	0.3488	1
5	0.3248	16.61	0.5149	1
6	0.1024	5.24	0.5673	-
7	0.2265	11.58	0.6831	-
8	0.1358	6.94	0.7525	2
9	0.3274	16.74	0.9199	2
10	0.1565	8.00	1.0000	2
Total	1.9556	100.00		10

Also recorded in Table 5.20 is the number of copies of the strings in the starting population actually selected by the reproduction operator for the new generation. Strings 5 and 9 have high fitness values. String 6 has a low fitness and does not survive. The fittest strings have the best chance of survival while some less fit strings may survive by chance (such as String 8 which is selected twice).

5.3.6 One-point crossover

The mating pool is developed from the application of the selection operator to the starting population. Pairs of parent strings are mated using an operator known as crossover. Crossover is analogous to a mechanism of natural genetics which involves the exchange of corresponding segments of a pair of chromosomes. A gene position is selected randomly so that crossover may proceed about that gene position by breakage and reunion. The GA operator of one-point crossover operates on pairs of strings in a similar way. A bit position on the strings is selected randomly and is called the crossover site. The corresponding segments of the strings after the crossover site are exchanged. The two offspring strings replace their parent strings in the new generation. The strings that have been created to form the second generation of strings are shown in Table 5.21. Crossover occurs with some specified probability of crossover. If crossover does not occur, the two selected parent strings pass to the new generation unchanged. GA users have achieved the most success with high

probabilities of crossover such as 0.7 to 0.9 (Goldberg and Kuo, 1987). A probability of crossover of 0.7 (on average, 7 strings in every 10 experience crossover) was used in the small scale GA as shown in Table 5.21. In this case, the five pairs of selected parent strings have all undergone crossover.

Table 5.21 The second generation

Offspring strings	Parent strings	Crossover site	Mutation site
010-001-010-110-110-111-110-110	9, 8	Bit 4	-
100-011-110-001-100-110-010-010	8, 9	Bit 4	-
000-001-010-110-110-111-110-101	8, 2	Bit 22	<i>Bit 1</i>
101-100-000-100-110-111-110-110	2, 8	Bit 22	-
010-101-001-111-110-010-101-010	9, 5	Bit 3	-
101-011-110-001-100-110-010-010	5, 9	Bit 3	-
010-001-001-010-101-100-010-101	4, 10	Bit 10	-
010-010-010-111-111-100-101-010	10, 4	Bit 10	-
101-100-000-100-110-111-110-101	2, 10	Bit 22	-
010-010-010-110-101-100-010-101	10, 2	Bit 22	-

5.3.7 Mutation

Mutation is the occasional random alteration of a bit. Mutation involves randomly selecting bit positions (with low probability) within strings which have experienced crossover and changing a zero at this bit position to one or a one to zero. A low probability of mutation of 0.01 (on average, one bit mutation in every one hundred bits crossed over) has been used in this example. There were 240 bits crossed over in the creation of the second generation and one bit was mutated. This bit was the first bit of String 3 as shown in bold in Table 5.21. Mutation is an insurance against the loss of potentially useful genetic information. If the 1's or 0's at a particular bit position on the string are lost for all the strings in the population they could never be restored without a mutation at this bit position. For example, in the second generation of strings there are no zeros at bit position 13 in any of the strings. A mutation at bit position 13 is the only way a zero may occur again at this bit position. Note that the optimum solution contains a zero at this bit position.

The second generation (strings in Table 5.21) has been created from the starting population. The costs of the strings, including penalty costs derived from hydraulic analyses of the demand loading conditions, in the second generation have been computed and included in Table 5.22. Some generation cost statistics for the second generation have been provided in Table 5.23.

Table 5.22 Cost of the strings in the second generation

String	Pipe cost (\$ million)	Penalty cost (\$ million)	Total cost (\$ million)
1	2.6279	4.7588	7.3867
2	2.1975	0.8804	3.0779
3	2.2736	8.6067	10.8803
4	3.0793	1.0847	4.1639
5	2.6495	0.2665	2.9160
6	2.3813	0.2322	2.6135
7	1.8992	3.7300	5.6292
8	2.3332	4.2758	6.6090
9	3.0180	1.0847	4.1027
10	2.1664	4.2231	6.3895

Table 5.23 Second generation cost statistics

Generation cost statistics		% Improvement over first generation
Lowest cost (String 6)	\$2.6135 million	14.4%
Highest cost (String 3)	\$10.8803 million	-10.3%
Average generation cost	\$5.3769 million	8.7%
Average pipe cost	\$2.4626 million	1.0%
Average penalty cost	\$2.9143 million	14.3%
Number of infeasible solutions	10	-
Average cost of strings selected for reproduction from first generation	\$5.1316 million	-

A better cost solution has been found for the second generation and the average generation cost has been reduced in the second generation. There are no feasible solutions in the new generation.

5.3.8 Subsequent generations and the Schema Theorem

The second generation of strings has been created from parts of the fitter strings of the first generation. The process is repeated for many generations until convergence of the population is observed.

A rigorous theoretical basis for the power exhibited by genetic algorithms was established in Holland's (1975) Schema Theorem. The fundamental concept of a schema or similarity template was discussed in Chapter 4. According to the Schema Theorem, short, low-order, above-average-fitness schemata called *building blocks* (Goldberg, 1989) receive exponentially increasing trials in subsequent generations. The GA selects strings with high fitness more often in reproduction to form a new population. Building blocks receive more samples in the new

5 Application of the traditional genetic algorithm to pipe network optimisation

population and are recombined with other building blocks in crossover to construct superior new strings from the best parts of the best old strings.

As a simple demonstration of the Schema Theorem at work for the small scale GA simulation performed on the Gessler problem, consider Schema H_1 and Schema H_2 below:

Schema H_1 ***_***_***_***_1*0_*1*_***_***

Schema H_2 ***_***_***_***_***_***_***_*10

There does not appear to be (and we do not expect) large numbers of useful building blocks in the random initial population of 10 strings, however Schema H_1 (order=3 and defining length=4) and Schema H_2 (order=2 and defining length=1) represent short, high performing string similarities in the starting population. Schema H_1 is present in the low cost Strings 2, 5 and 9 (and also String 8) and Schema H_2 is present in the low cost Strings 5 and 9 (and also Strings 4 and 8).

The schemata H_1 and H_2 are termed building blocks and they make their presence felt in the new population. The second generation contains seven strings which are representatives of Schema H_1 and six strings which are representatives of Schema H_2 . The new population shows some improvement compared to the starting population, producing two solutions that are superior to any of the randomly generated solutions in the starting population. The three lowest cost strings in the new population of 10 offspring strings each contain the schemata H_1 and H_2 . Schema H_1 and Schema H_2 are likely to be important building blocks in subsequent generations of this small scale GA search.

5.4 Sensitivity Analysis of Genetic Algorithm Parameters

This application of genetic algorithms (GAs) to the Gessler pipe network optimisation uses the traditional three-operator GA (Goldberg, 1989). Several full scale GA model runs were performed to try to determine the effect on GA performance of variations of the following GA parameters:

- population size, N
- probability of crossover, p_c
- probability of mutation, p_m

In addition, various random number generator seeds were used to check the influence of the random number sequence.

5 Application of the traditional genetic algorithm to pipe network optimisation

DeJong (1975) constructed a set of five test functions to be minimised to test the robustness of the standard three-operator GA. DeJong performed extensive experiments with combinations of the GA parameters on one of his five test functions. The function was a smooth, convex, quadratic function of three variables.

$$f(x_i) = \sum_{j=1}^3 x_j^2 \quad (5.8)$$

Two measures of effectiveness were considered. On-line performance is the average of all function evaluations up to the current trial. Off-line performance measures convergence and is an average of the best function evaluations up to a given time. DeJong showed a population size $N=200$ maintained population diversity and eventually showed excellent convergence. A $N=50$ displayed better initial on-line performance. DeJong's parametric study found a probability of crossover of $p_c=0.6$ exhibited good on-line and off-line performance. DeJong achieved good on-line performance with a probability of mutation $p_m=0.001$ to 0.01 and best off-line success with $p_m=0.005$ to 0.02 . A $p_m>0.02$ helped maintain population diversity but at the cost of poor performance. An almost insignificant $p_m=0.001$ displayed excellent on-line performance but did not achieve convergence.

Goldberg and Kuo (1987) suggested good results may be obtained with relatively small population sizes ($N=35$ to 200), high crossover probabilities ($p_c=0.5$ to 1.0) and mutation probabilities inversely proportional to the population size ($p_m=0.1/N$ to $5.0/N$). Goldberg and Kuo emphasise the GAs are not highly sensitive to the chosen GA parameters.

In accordance with these recommendations the GA parameters were initially chosen as:

$$N=100, p_c=0.7 \text{ and } p_m=0.02$$

With a population size of $N=100$, the GA successively generates and evaluates new populations of 100 coded strings. On average, approximately $(p_c)(N)=(0.7)(100)=70$ of the 100 coded strings in the new population will be created by crossing over two strings in the old population using a probability of crossover of $p_c=0.7$. The other 30 or so strings will be copied from the old population to the new population untouched. Approximately 2 bits in every 100 bits crossed over are mutated using a probability of mutation of $p_m=0.02$. Since the string length is 24 bits for the Gessler problem, about 1 bit in every two strings crossed over or about 34 bits in a new population will be altered by a mutation.

5 Application of the traditional genetic algorithm to pipe network optimisation

When a new string is created in the new population by a crossover, an evaluation of the string is required. An evaluation of a coded string is the computation of the total cost of the corresponding network solution, including pipe costs and penalty costs. The calculation of the appropriate penalty costs requires hydraulic analyses of the network solution for each demand pattern. An evaluation of a string untouched by crossover (a copy of a parent string) is not necessary since the cost information is already known. Therefore, there are about 70 new string evaluations for every generation. The GA runs were allowed a maximum of 10,000 string evaluations. The 10,000 evaluations are equivalent to about 143 ($=10,000/70$) generations of populations of coded strings.

If a population size $N=50$ and probability of crossover $p_c=0.5$ are used, on average about 25 ($=0.5 \times 50$) new strings would be formed in the new population and 400 ($=10,000/25$) generations of populations of coded strings would constitute the 10,000 function evaluations. The smaller population sizes and crossover probabilities have the benefit of more generations, but this is at the expense of less mixing of parent strings per generation. The evaluation of 10,000 strings from the 16,777,216 possible pipe network combinations for the Gessler problem represents a search involving less than 0.06% of the solution space. The GA runs of 10,000 evaluations utilised about 6 minutes of CPU computing time on a SUN SPARCstation-1+ (about 9 minutes of CPU time on a SUN 4/280 computer).

5.4.1 Variations of the random number generator seed

The random number generator seed is an arbitrarily chosen integer which initiates a unique sequence of random numbers. The GA search is a randomised search directed by probabilistic operators. A GA run usually requires the use of thousands of random numbers.

The random number generator function used in this thesis (Barnard and Skillicorn, 1988) generates sequences of pseudo-random numbers using a recurrence called a linear congruential generator of the form:

$$x_{i+1} = (a x_i + c) \text{ mod } m \quad (5.9)$$

in which the constants a , c and m are suitably chosen integers. The constant m is usually chosen as the maximum positive integer the machine can represent, typically 2^{31} . If the constants a (chosen to be positive) and c (chosen to be non-negative) are carefully selected, the recurrence will contain all the integers between 0 and $m-1$ exactly once in some random sequence before there is any repetition and there will be no correlation between successive terms.

5 Application of the traditional genetic algorithm to pipe network optimisation

The random number generator seed is the starting point in the sequence of random numbers. Any particular seed produces the same sequence of random numbers. The user is required to input the starting seed for the GA runs. The same seed will generate the same starting population of strings for the genetic algorithm run. This is useful for the comparison of different combinations of GA parameters and also for debugging purposes.

Some simple tests were performed to verify the randomness of the random numbers generated. The random number generator and an arbitrary seed, 12345, were used to generate 1,000,000 random numbers between 0 and 1. The numbers were placed in one of the ten intervals (0→0.1, ..., 0.9→1.0). The resulting distribution in Table 5.24 verifies the uniformity of the distribution of the numbers generated.

Table 5.24 Distribution of 1,000,000 random numbers

Interval	Random Numbers
0.0→0.1	100,023
0.1→0.2	100,105
0.2→0.3	99,920
0.3→0.4	99,733
0.4→0.5	100,417
0.5→0.6	100,245
0.6→0.7	100,088
0.7→0.8	99,693
0.8→0.9	99,422
0.9→1.0	100,354

The mean of the 1,000,000 random numbers generated was 0.49988 and the mean absolute deviation from the mean was 0.24991. The statistics of the sample almost coincide with the expected mean of 0.50000 and the expected mean absolute deviation from the mean of 0.25000.

Random numbers are required by the GA run to generate the starting population of coded strings, select fitter strings in reproduction, decide whether crossover will occur, choose a crossover site and to decide whether mutation will occur for a particular bit position. For the GA run of 10,000 evaluations, approximately 300,000 random numbers were produced by the random number generator. The integer m in the random number generator was chosen as $2^{31}=2,147,483,648$. Therefore, the sequence of random numbers generated is much longer than the number of random numbers required by the GA run. As a further precaution, the actual number of crossovers and mutations are counted and are compared with the expected number of crossovers and mutations for the GA run.

5 Application of the traditional genetic algorithm to pipe network optimisation

Five GA runs were initiated using five different random number seeds ($seed=100, 200, 300, 400$ and 500). The population size, probability of crossover and probability of mutation were held constant. The lowest cost network solutions achieved in the GA runs are given in Table 5.25. All the GA runs with different random number seeds (unique sequences of random numbers generating different starting populations) approach the global optimum solution and all achieve a minimum cost within 5.0% of the global optimum solution cost of \$1.7503 million. The optimum solution is reached using a $seed=200$. The minimum cost network designs identified by the GA runs are slight variations of the global minimum cost network design. The chosen random number seed only appears to affect the performance of the GA slightly.

The solution numbers for the lowest cost network designs identified by each GA run are given in Table 5.25. The feasible near-optimal network solutions to the Gessler problem were identified by the complete enumeration and they are listed in Table 5.4. The infeasible network solutions determined by the GA runs in this analysis are given in Section 5.4.5. Marginally infeasible designs are prominent in the GA runs using a $seed=300$ and a $seed=400$. The network costs in Table 5.25 are the sum of pipe costs and penalty costs.

Table 5.25 Minimum cost network solution with varying random number seed

Random number seed	Minimum cost network (\$ million)	Generation number	Evaluation number	Solution number (Table 5.4)	% difference from optimum
100	1.8285	39	2730	20	4.47
200	1.7503 [†]	122	8540	1	0.0
300	1.8154*	104	7280	2**	3.72
400	1.8048*	118	8260	1**	3.11
500	1.7725	113	7910	4	1.27

$N=100, p_c=0.7, p_m=0.02$

[†] global optimum

* includes penalty costs

** infeasible designs (presented in Table 5.29)

The least cost solution achieved in a generation is plotted against the number of function evaluations up to and including the current generation for a $seed=200$ and a $seed=400$ in Figure 5.8. The GA run using a $seed=200$ is successful in achieving the optimum after 8,540 evaluations. Although the $seed=200$ exhibits superior performance the general shape of the two plots is similar. The average cost of the solutions in a generation is plotted against the number of function evaluations up to and including the current generation for a $seed=200$ and a $seed=400$ in Figure 5.9. The gradual improving trend of both plots towards the optimum solution is evident.

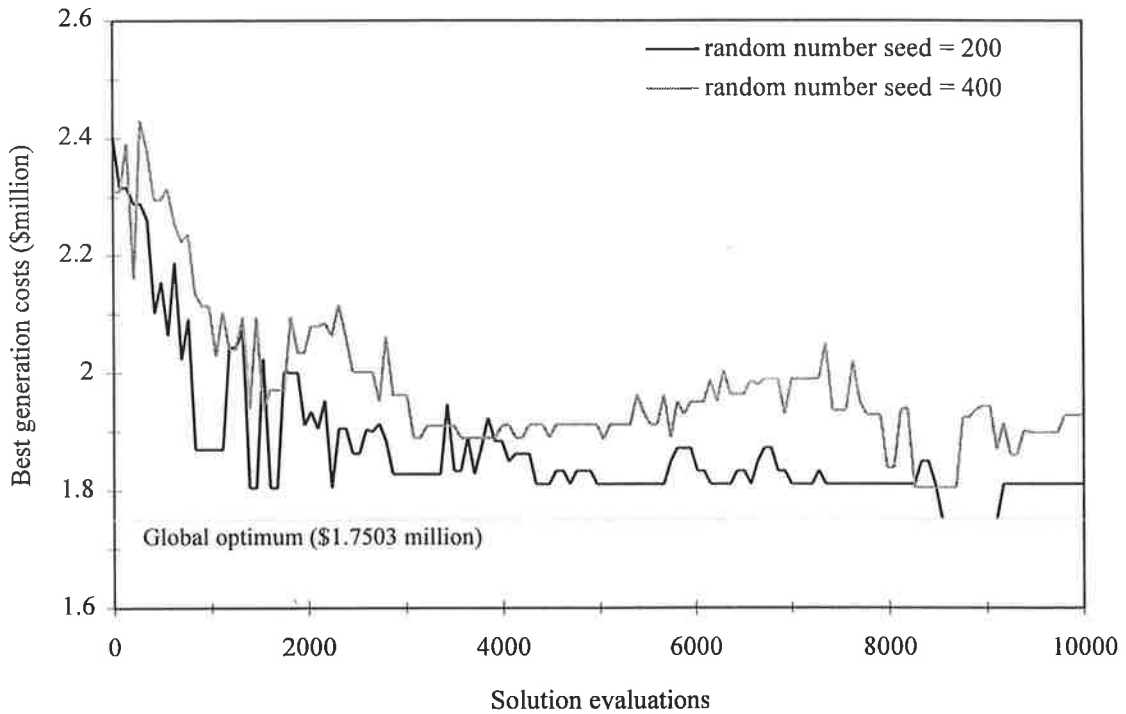


Figure 5.8 Best generation costs for random number seeds, $seed=200$ and $seed=400$ ($N=100$, $p_c=0.7$ and $p_m=0.02$)

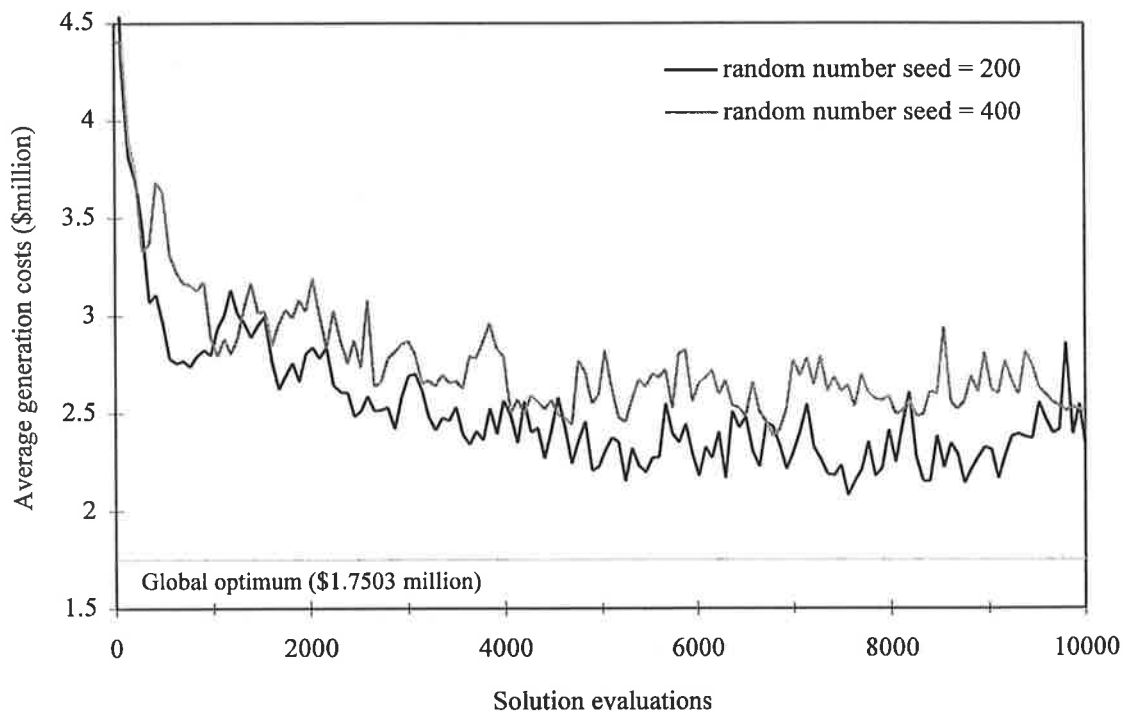


Figure 5.9 Average generation costs for random number seeds, $seed=200$ and $seed=400$ ($N=100$, $p_c=0.7$ and $p_m=0.02$)

The GA parameters of $N=100$, $p_c=0.7$ and $p_m=0.02$ displayed good convergence and attained low cost network solutions. The random number $seed=100$ was chosen for all the subsequent GA runs in the sensitivity study in this chapter.

5.4.2 Variations of the population size, N

A suitable population size is an important consideration for GA users. Populations which are too small are likely to converge too quickly and without sufficient processing of the limited numbers of schemata, and population sizes which are too large are likely to take a long time to converge, since it takes longer to get enough mixing of building blocks (Goldberg, 1989b).

Goldberg (1985) developed a theoretical basis for selecting initial population size for a binary string of fixed length. Probability rules are used to compute the expected total number of unique schemata, n_s in a random population of strings of given population size. A single binary string (population size, $N=1$) of length l contains 2^l schemata since each string position may take its assigned value of 0 or 1 or the 'don't care' symbol *. As population size increases, the number of schemata approaches 3^l , the maximum possible number of schemata for a binary string of length l (Goldberg, 1989b). Goldberg (1985) argues that the schemata in this string are only useful when they are duplicated in other strings or combined with new schemata from other strings. The number of excess schemata in a random population n_{es} is a measure of this schemata processing potential.

$$n_{es} = n_s - 2^l \quad (5.10)$$

Goldberg (1985) maximises the number of excess schemata per population member. An increase in population size is desirable if it increases the number of schemata per population member.

$$\max \left(\frac{n_s - 2^l}{N} \right) \quad (5.11)$$

A Fibonacci search is used to maximise this function. An approximate formula is developed for optimal population size, N^* .

$$N^* = 1.65 \cdot 2^{0.21l} \quad (5.12)$$

Tables of recommended population sizes for string length l are provided by Goldberg (1985). Goldberg's theory suggests a population size of $N^*=51$ for a string length of 24 bits.

5 Application of the traditional genetic algorithm to pipe network optimisation

Goldberg (1989b) extends the computations of the expected numbers of schemata in populations of fixed-length binary strings (Goldberg, 1985), to consider alphabets of higher cardinality and for cases in which more than one copy of each schema is required. Goldberg (1989b) derives an approximation for the real-time rate of schema processing by considering the calculations of the expected number of schemata in a population of given size and the relationship between population size and number of generations to convergence (for two different assumed levels of convergence). The optimal population sizes maximise the real-time rate of schema processing for different string lengths and for processing by serial and parallel machines. Goldberg's theory suggests relatively small population sizes are appropriate for serial processors (such as the one used in this research).

Goldberg et al. (1992) presented an alternative approach to the population sizing question. An equation for sizing populations was derived from statistical decision theory which chooses 'large-enough' population sizes to control errors in building block decision making. The equation is modified to include the noise introduced by selection and other GA operators, and the explicit noise of the objective function.

A series of GA runs were performed on the Gessler problem using population sizes between $N=20$ and $N=500$. The probability of crossover, probability of mutation and random number seed were held constant as:

$$p_c=0.7, p_m=0.02, seed=100$$

The minimum cost network solutions determined by the GA runs are summarised in Table 5.26. The plots of best generation costs and average generation costs for GA model runs using various population sizes are shown in Figures 5.10-5.17.

Population sizes of $N=20$ and $N=150$ reach the global minimum cost network for \$1.7503 million. The best generation cost plots and average generation cost plots for the large population sizes $N=150$ and $N=500$ in Figures 5.16 and 5.17 respectively suggest the GA run needed additional generations to attain the optimum since convergence is slower. The large populations have a greater capacity for coded information, however they require more generations for sufficient processing and mixing of this information.

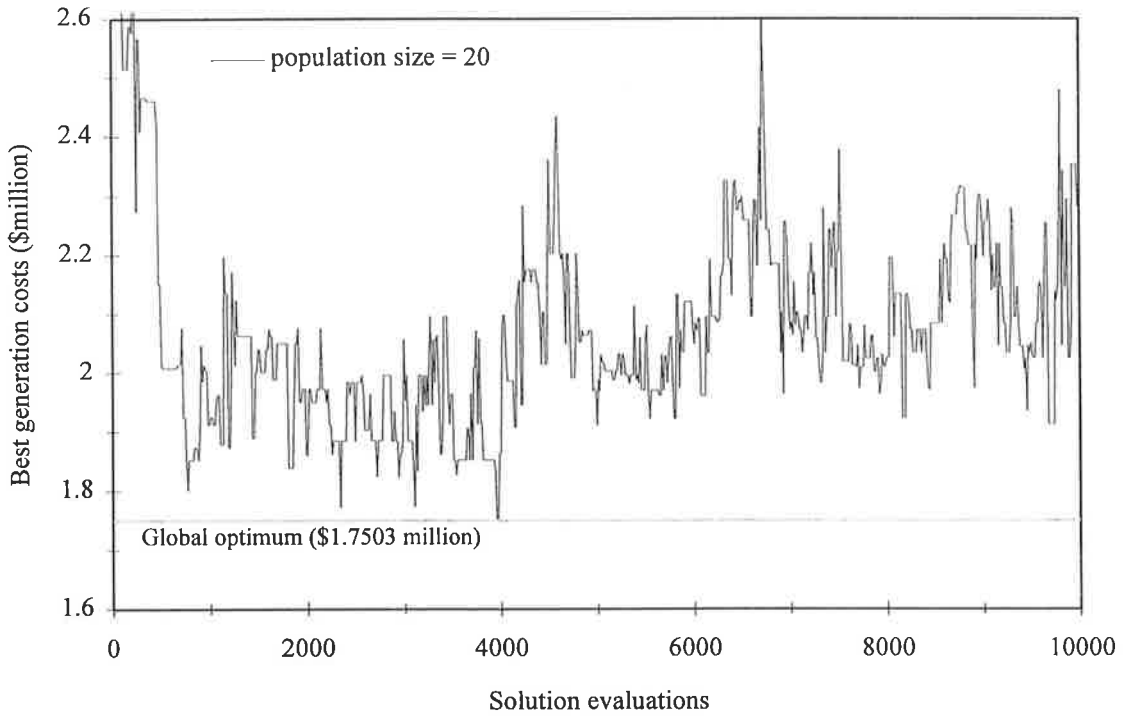


Figure 5.10 Best generation costs for a population size, $N=20$
($p_c=0.7$, $p_m=0.02$ and $seed=100$)

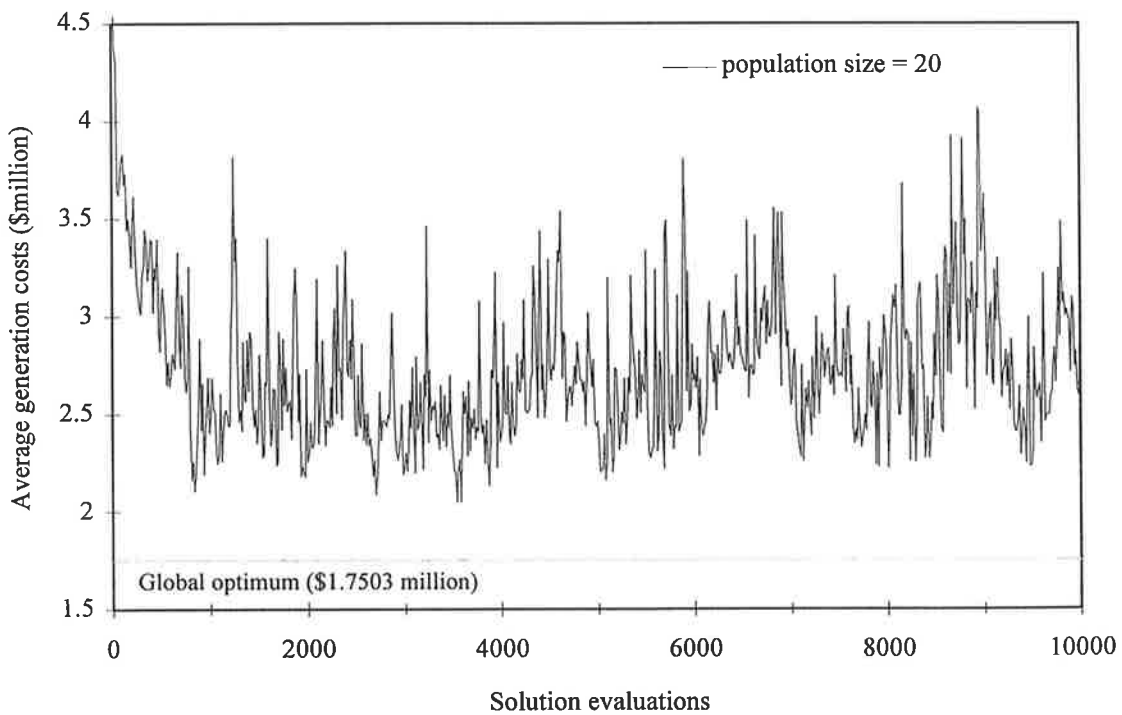


Figure 5.11 Average generation costs for a population size, $N=20$
($p_c=0.7$, $p_m=0.02$ and $seed=100$)

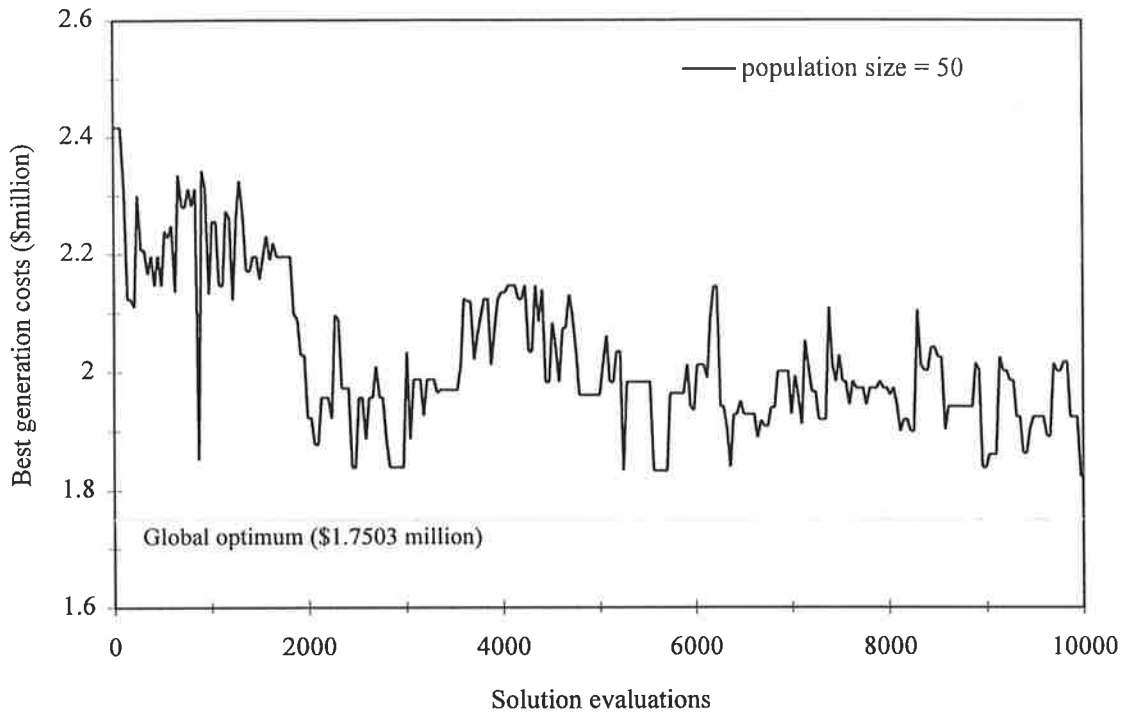


Figure 5.12 Best generation costs for a population size, $N=50$
($p_c=0.7$, $p_m=0.02$ and $seed=100$)

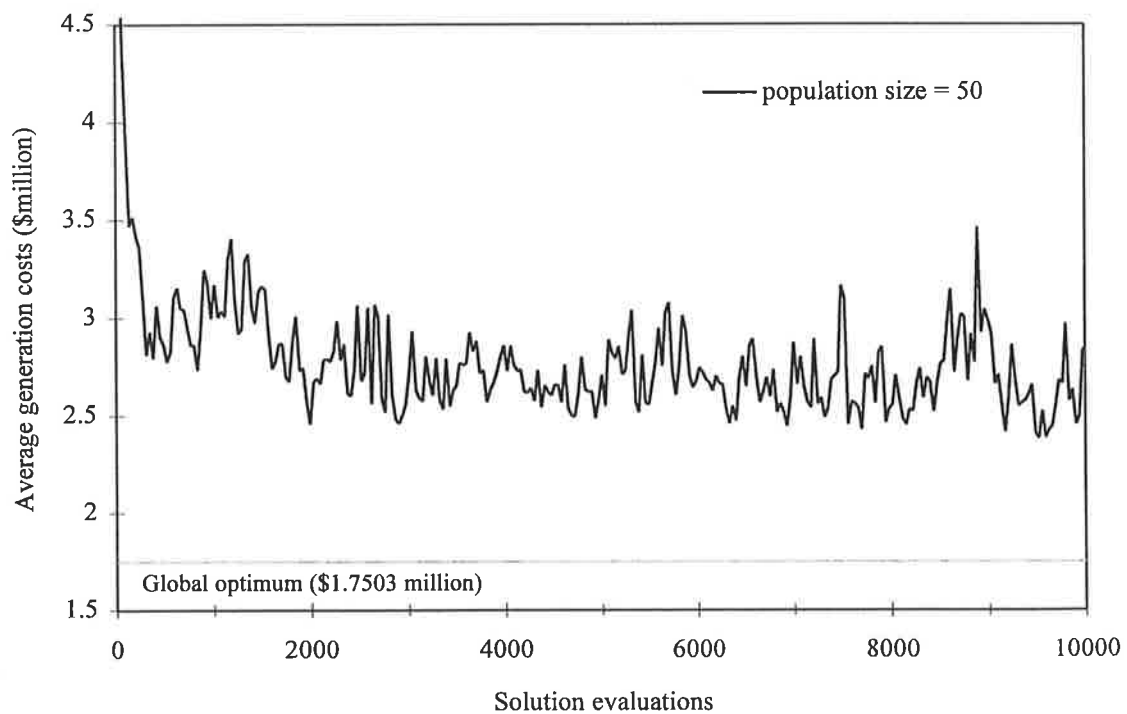


Figure 5.13 Average generation costs for a population size, $N=50$
($p_c=0.7$, $p_m=0.02$ and $seed=100$)

5 Application of the traditional genetic algorithm to pipe network optimisation

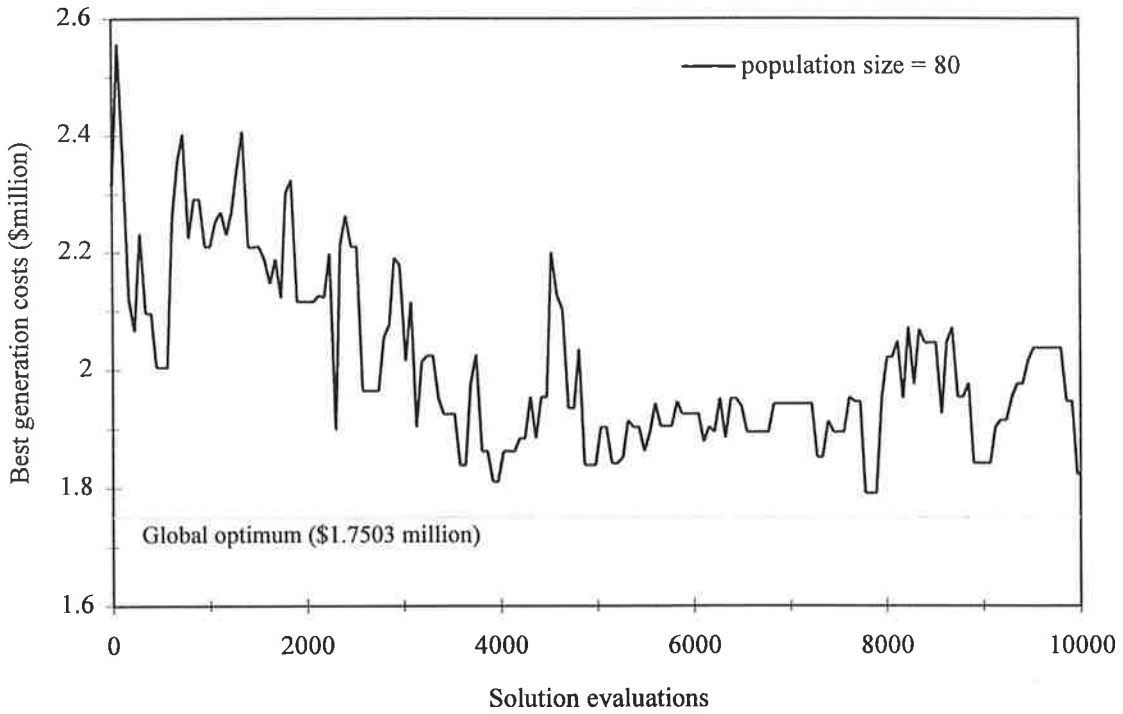


Figure 5.14 Best generation costs for a population size, $N=80$
($p_c=0.7$, $p_m=0.02$ and $seed=100$)

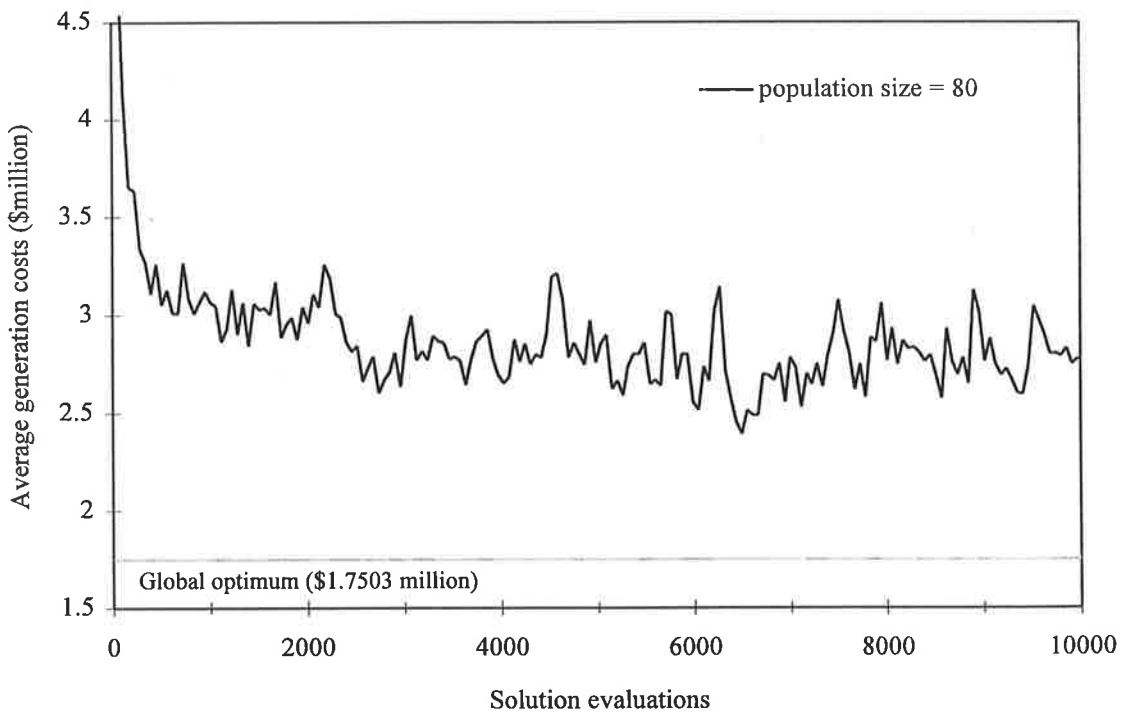


Figure 5.15 Average generation costs for a population size, $N=80$
($p_c=0.7$, $p_m=0.02$ and $seed=100$)

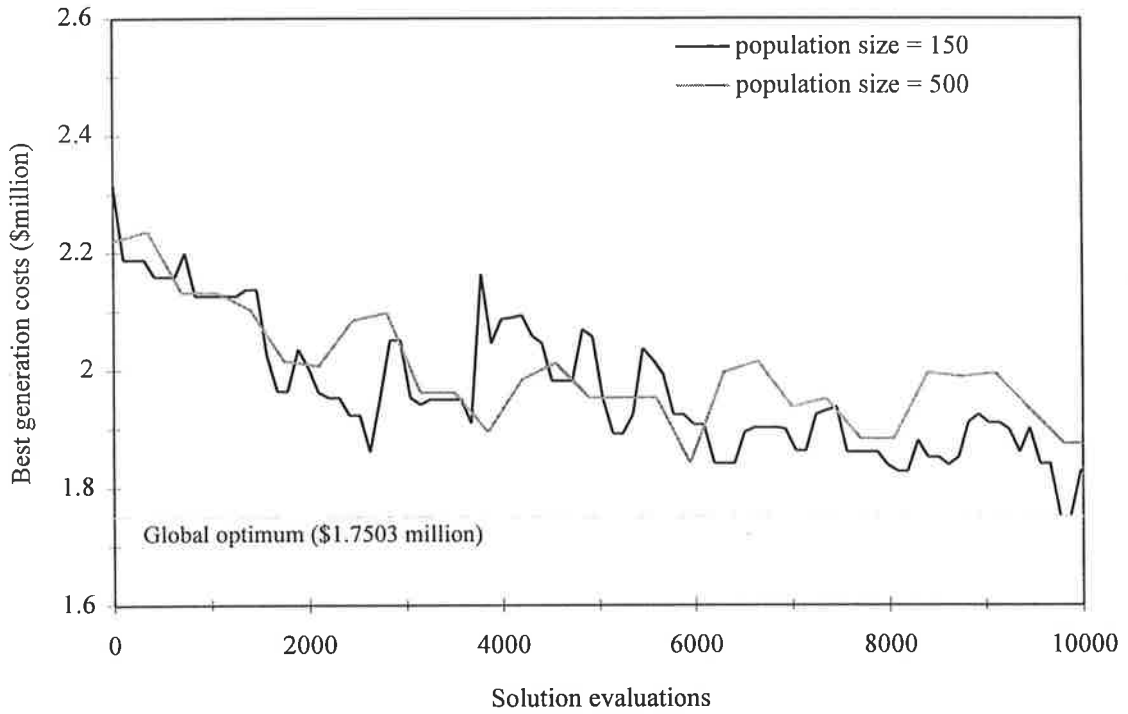


Figure 5.16 Best generation costs for population sizes, $N=150$ and $N=500$ ($p_c=0.7$, $p_m=0.02$ and $seed=100$)

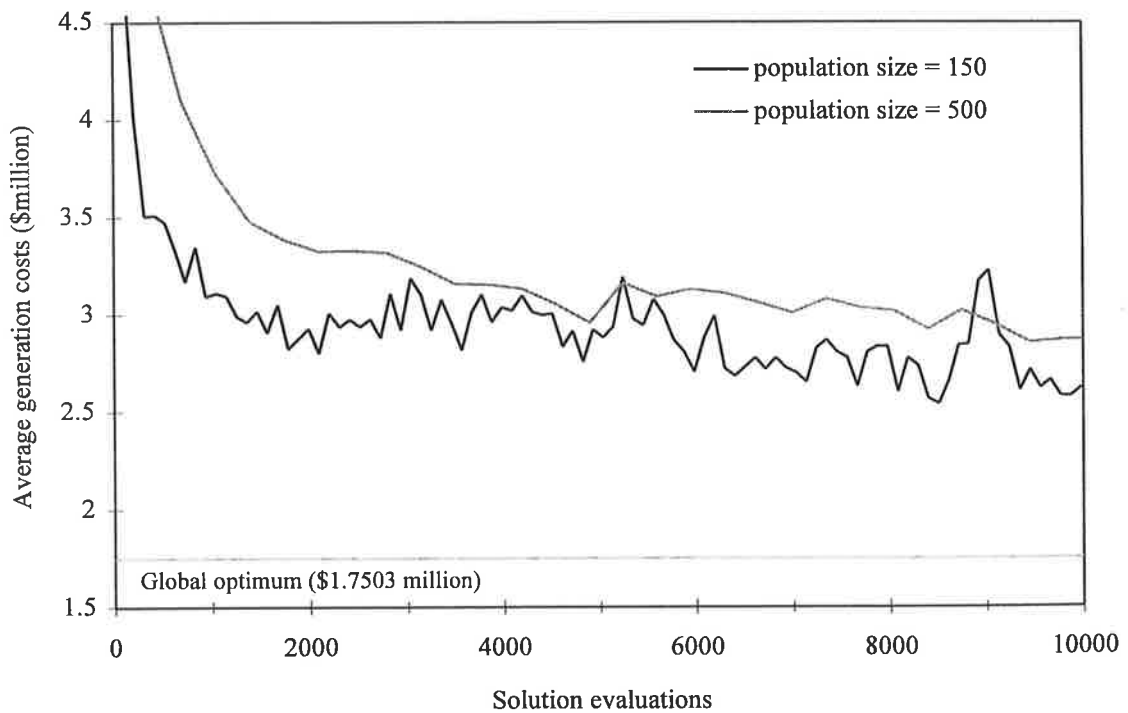


Figure 5.17 Average generation costs for population sizes, $N=150$ and $N=500$ ($p_c=0.7$, $p_m=0.02$ and $seed=100$)

Table 5.26 Minimum cost network solution with varying population size, N

Population size, N	Minimum cost network (\$ million)	Generation number	Evaluation number	Solution number (Table 5.4)	% difference from optimum
20	1.7503 [†]	283	3962	2	0.0
40	1.7725	140	3920	3	1.27
50	1.8232	285	9975	19	4.17
60	1.8300	161	6762	22	4.56
80	1.7910	139	7784	5	2.33
100	1.8285	39	2730	20	4.47
150	1.7503 [†]	93	9765	2	0.0
200	1.8048*	64	8960	1**	3.11
500	1.8417	17	5950	35	5.22

$$p_c=0.7, p_m=0.02, seed=100$$

† global optimum

* includes penalty costs

** infeasible designs (presented in Table 5.29)

The population sizes of $N=80$ and $N=150$ determine near-optimal and optimal cost network solutions respectively. The best generation cost plots for $N=80$ and $N=150$ in Figures 5.14 and 5.16 respectively display consistent improvement of the lowest cost network solution as the GA run progresses. The convergence of the average generation cost plots for $N=80$ and $N=150$ is shown in Figures 5.15 and 5.17. The larger population sizes maintain more diverse populations for longer and convergence of average generation costs is slower.

The small population sizes of $N=20$ and $N=40$ have determined cost effective network designs. The initial convergence of the best generation cost plots for $N=20$ and $N=50$ in Figures 5.10 and 5.12 respectively is efficient. The smallest population size considered, $N=20$ identifies the global optimum solution after only 3,962 solution evaluations, however, this solution is subsequently lost and the GA does not find another network solution less than \$1.90 million for the remaining 6,000 evaluations. The variation of average generation costs for the GA runs with $N=20$ and $N=50$ are shown in Figures 5.11 and 5.13 respectively. A moderate population size of $N=100$ was chosen for all the following GA runs in this analysis.

5.4.3 Variations of the probability of crossover, p_c

Ten GA runs were performed using varying values of crossover probability between $p_c=0.1$ and $p_c=1.0$ while the other GA parameters were held constant. Applied to a population size of $N=100$, a $p_c=0.1$ is expected to create 10 offspring strings by crossover of 5 pairs of selected parent strings. Therefore, it is assumed there are approximately 10 new string evaluations on average per generation. Using a $N=100$, a $p_c=1.0$ will crossover every pair of selected parent

5 Application of the traditional genetic algorithm to pipe network optimisation

strings to create 100 offspring strings in the new population and therefore it is known there are 100 new string evaluations per generation. There are about 1,000 generations required to complete 10,000 evaluations using a $p_c=0.1$ while only 100 generations are required to complete 10,000 evaluations using a $p_c=1.0$.

The minimum cost network solutions identified by the GA model runs with various crossover rates are given in Table 5.27. The global optimum solution is not found by any of the GAs, however the lowest cost network solution determined by the GA run using $p_c=0.9$ for \$1.7725 million is just one step from the global optimum. All the GA runs in Table 5.27 achieve minimum cost solutions within 5.5% of the cost of the global optimum.

Table 5.27 Minimum cost solution with varying probability of crossover, p_c

Probability of crossover, p_c	Minimum cost network (\$ million)	Generation number	Evaluation number	Solution number (Table 5.4)	% difference from optimum
0.1	1.8010	48	480	8	2.90
0.2	1.8010	69	1380	9	2.90
0.3	1.7910	138	4140	5	2.33
0.4	1.8115	201	8040	10	3.50
0.5	1.7999	108	5400	6	2.83
0.6	1.8154*	133	7980	2**	3.72
0.7	1.8285	39	2730	20	4.47
0.8	1.8460*	102	8160	3**	5.47
0.9	1.7725	41	3690	3	1.27
1.0	1.8115	80	8000	12	3.50

$N=100, p_m=0.02, seed=100$

* includes penalty costs

** infeasible designs (presented in Table 5.29)

The best generation cost plots for low probabilities of crossover, $p_c=0.2$ and $p_c=0.4$ in Figure 5.18 exhibit fast initial convergence, however further improvement is not significant beyond 1,500 solution evaluations. The best generation cost plots for higher crossover rates in Figures 5.19 and 5.20 display slower, but more consistent convergence. The average generation cost plots for various crossover rates are shown in Figure 5.21. The average generation costs are higher for higher (more disruptive) crossover rates.

5 Application of the traditional genetic algorithm to pipe network optimisation

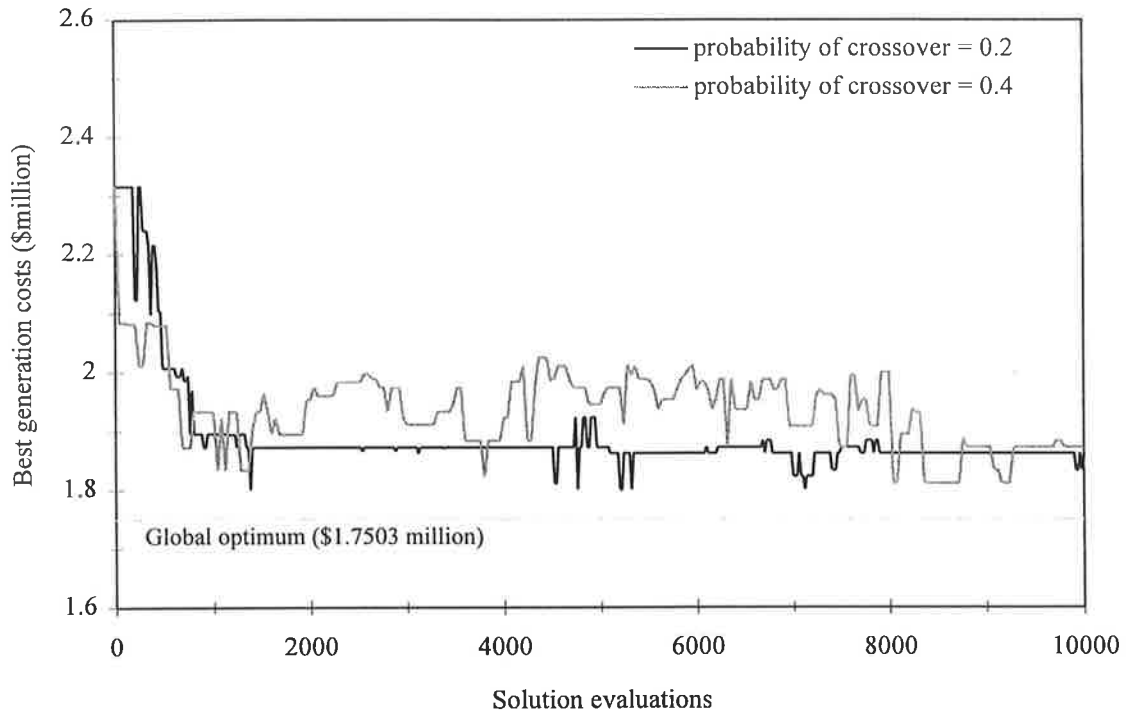


Figure 5.18 Best generation costs for probability of crossover, $p_c=0.2$ and $p_c=0.4$ ($N=100$, $p_m=0.02$ and $seed=100$)

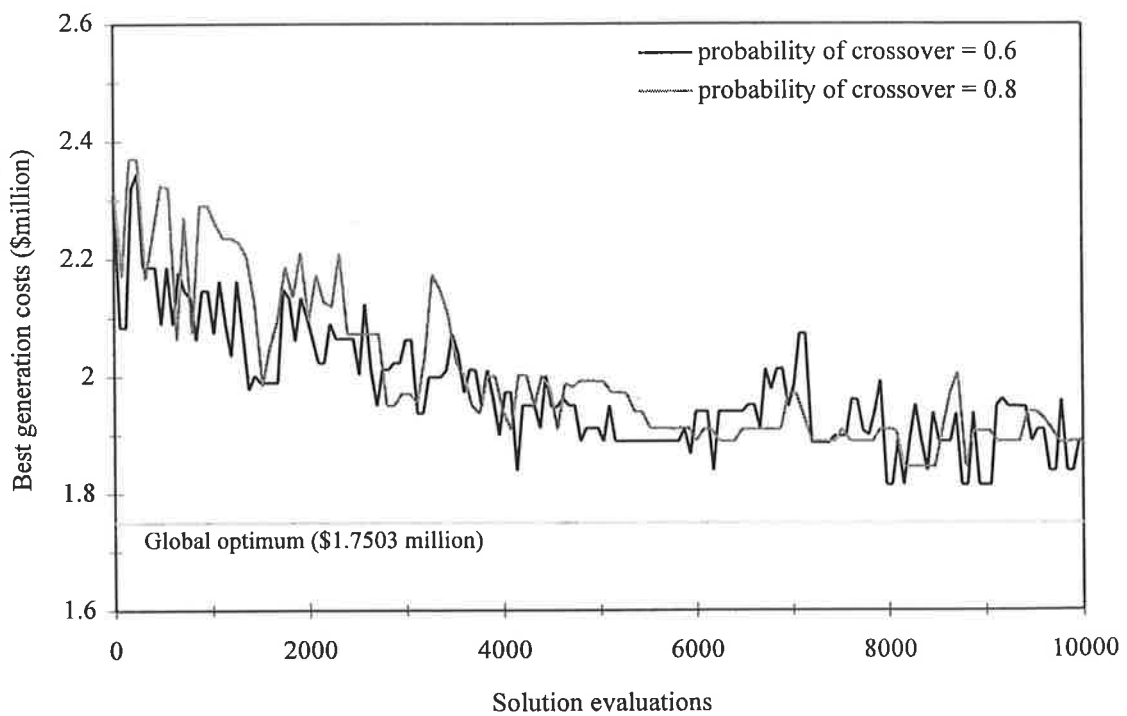


Figure 5.19 Best generation costs for probability of crossover, $p_c=0.6$ and $p_c=0.8$ ($N=100$, $p_m=0.02$ and $seed=100$)

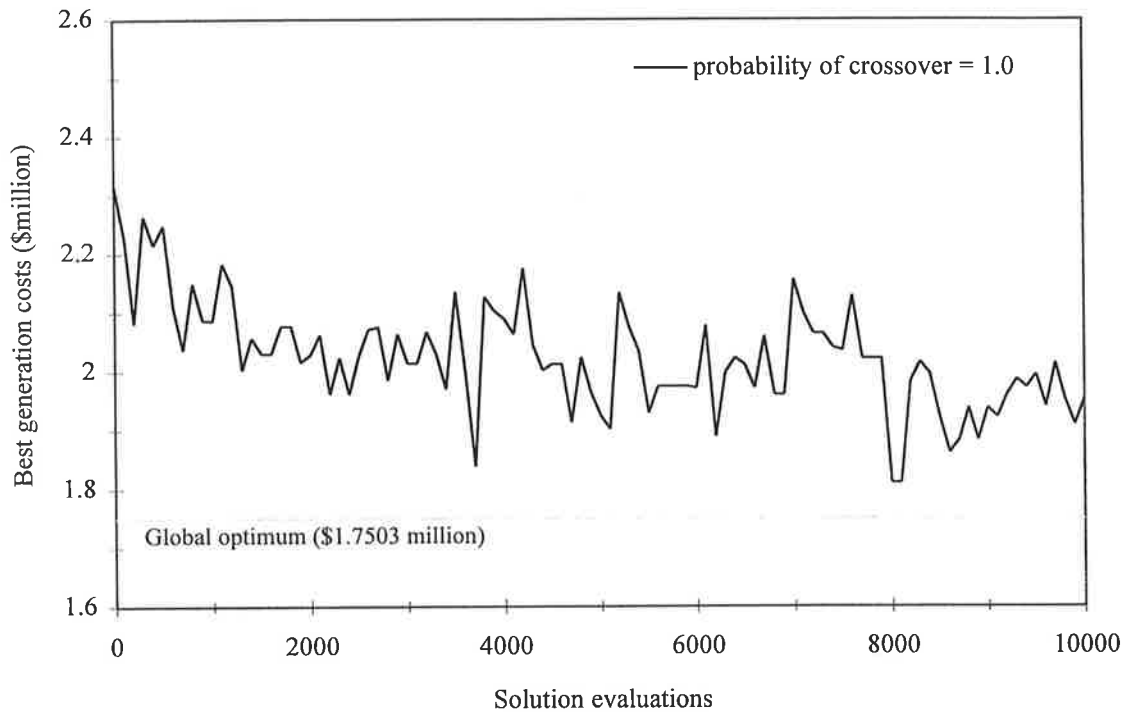


Figure 5.20 Best generation costs for a probability of crossover, $p_c=1.0$
($N=100$, $p_m=0.02$ and $seed=100$)

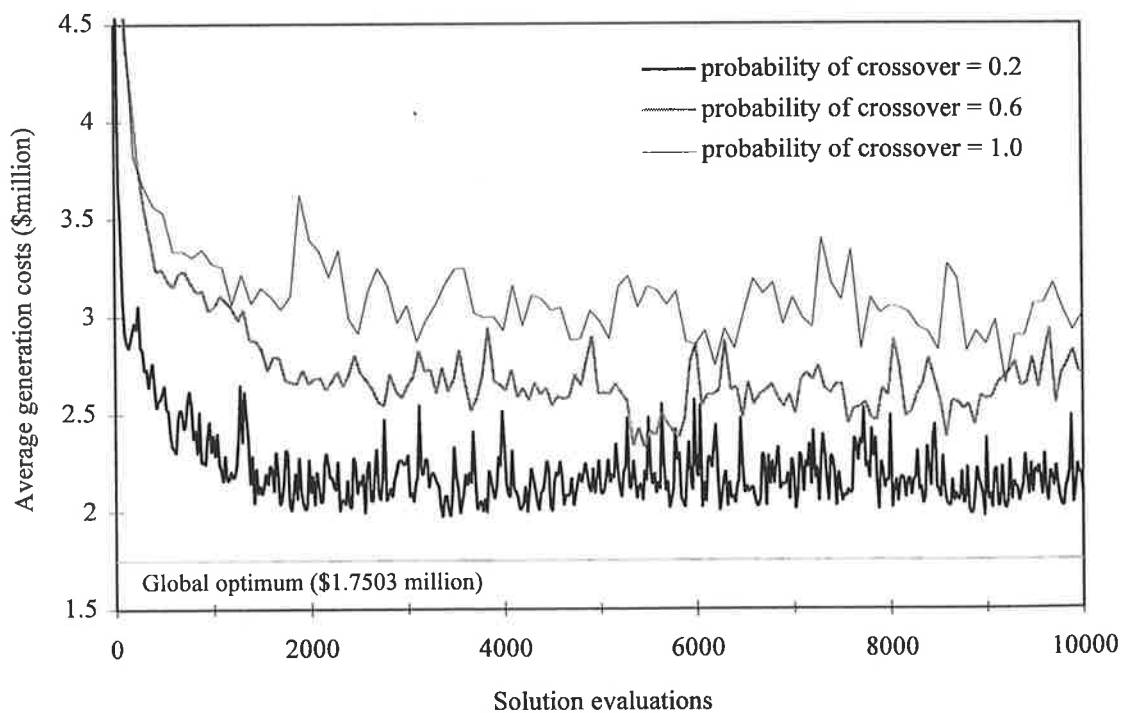


Figure 5.21 Average generation costs for varying probability of crossover, p_c
($N=100$, $p_m=0.02$ and $seed=100$)

5.4.4 Variation of the probability of mutation, p_m

The optimum network solution is determined by two of the nine GA runs with varying mutation probabilities between $p_m=0.0$ and $p_m=0.2$. The probability of mutation is the chance that a bit visited by the mutation operator is altered by mutation. The mutation operator considers each bit of every offspring string which is the result of the crossover of a pair of parent strings.

The minimum cost network solutions determined by the GA runs for various mutation rates are summarised in Table 5.28. The global optimum solution is identified using $p_m=0.0$ and $p_m=0.03$. The GA runs using $p_m>0.03$ are not very successful in identifying low cost network solutions.

Table 5.28 Minimum cost solution with varying probability of mutation, p_m

Probability of mutation, p_m	Minimum cost network (\$ million)	Generation number	Evaluation number	Solution number (Table 5.4)	% difference from optimum
0.0	1.7503†	28	1960	2	0.0
0.001	1.8115	57	3990	11	3.50
0.005	1.7725	31	2170	3	1.27
0.01	1.8232	35	2450	19	4.17
0.02	1.8285	39	2730	20	4.47
0.03	1.7503†	38	2660	2	0.0
0.05	1.8623	35	2450	>50	6.40
0.1	1.8417	5	350	36	5.22
0.2	1.9510	119	8330	>50	11.47

$N=100$, $p_c=0.7$, $seed=100$

† global optimum

The curves of best generation costs for the low mutation rates $p_m=0.0$ and $p_m=0.005$ in Figure 5.22 show less variability compared to the curves of best generation costs for high mutation rates $p_m=0.03$ and $p_m=0.1$ in Figure 5.23. The plots of average generation costs for various mutation rates are shown in Figure 5.24. The average generation costs are lower for the lower (less disruptive) mutation rates. A high value of probability of mutation of $p_m=0.1$ is too disruptive and the plots of best generation costs and average generation costs resemble a random walk in the solution space. The plot of best generation costs using $p_m=0.03$ in Figure 5.23 is more successful identifying the optimum solution on two distinct occasions within the given 10,000 solution evaluations. The plots of best generation cost plots for the relatively low mutation probabilities of $p_m=0.0$ and $p_m=0.005$ in Figure 5.22 approach the optimum solution efficiently. The GA run with no mutations easily achieves the global optimum, however after about 4,000 solution evaluations the optimum is lost and never regained.

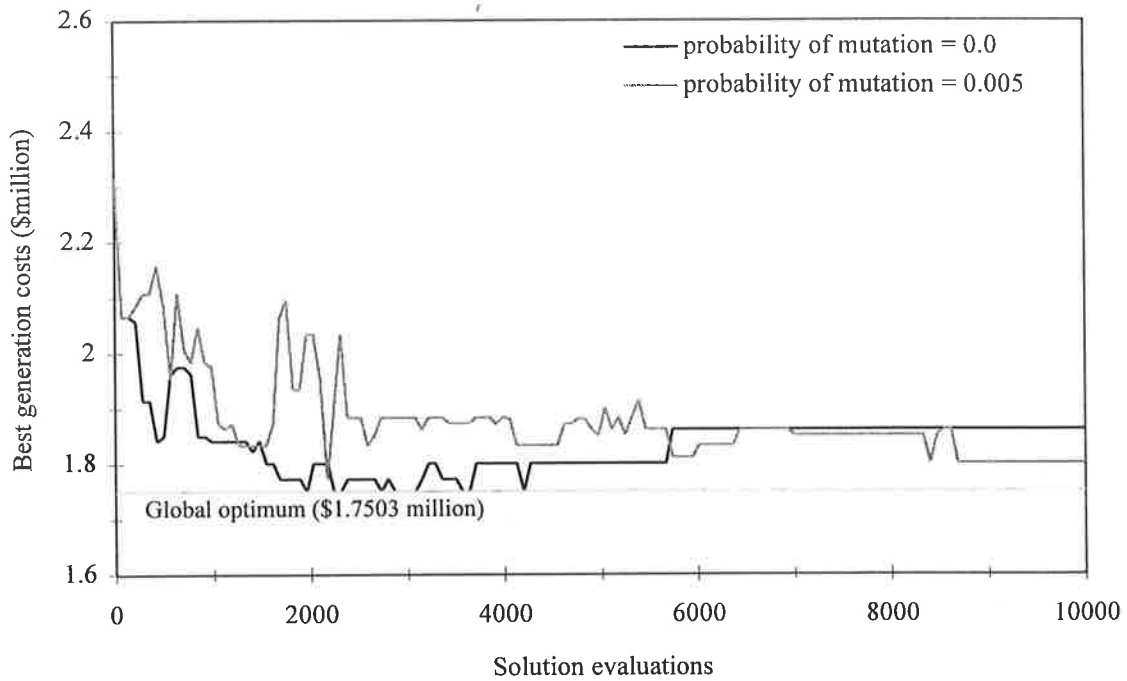


Figure 5.22 Best generation costs for probability of mutation, $p_m=0.0$ and $p_m=0.005$ ($N=100$, $p_c=0.7$ and $seed=100$)

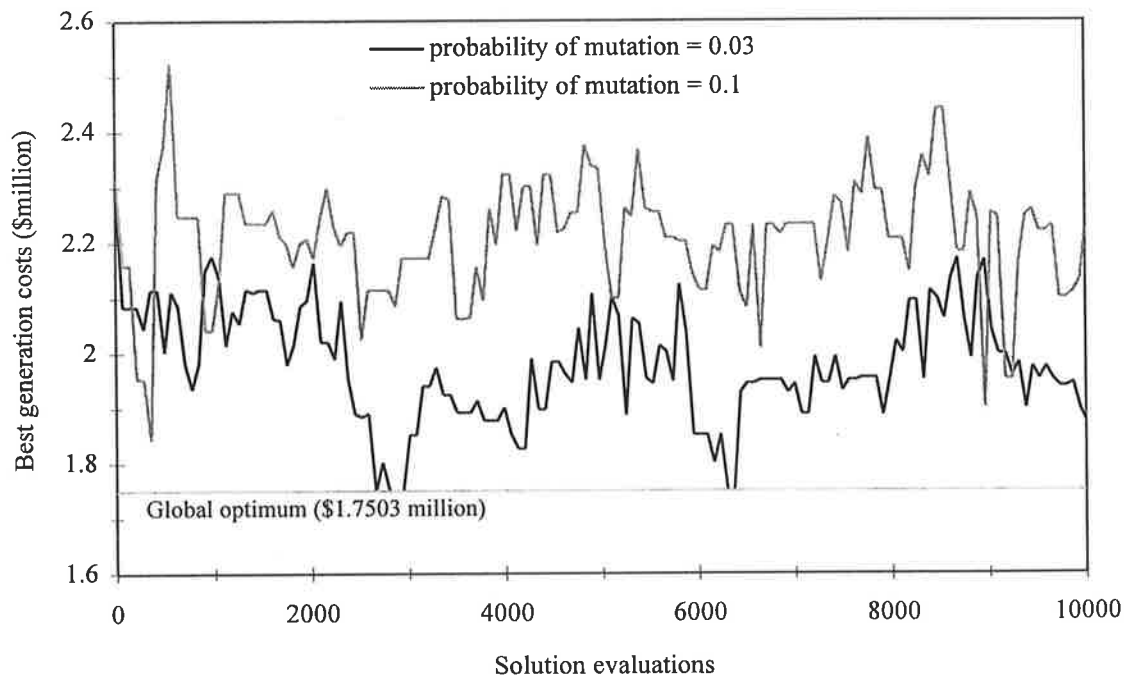


Figure 5.23 Best generation costs for probability of mutation, $p_m=0.03$ and $p_m=0.1$ ($N=100$, $p_c=0.7$ and $seed=100$)

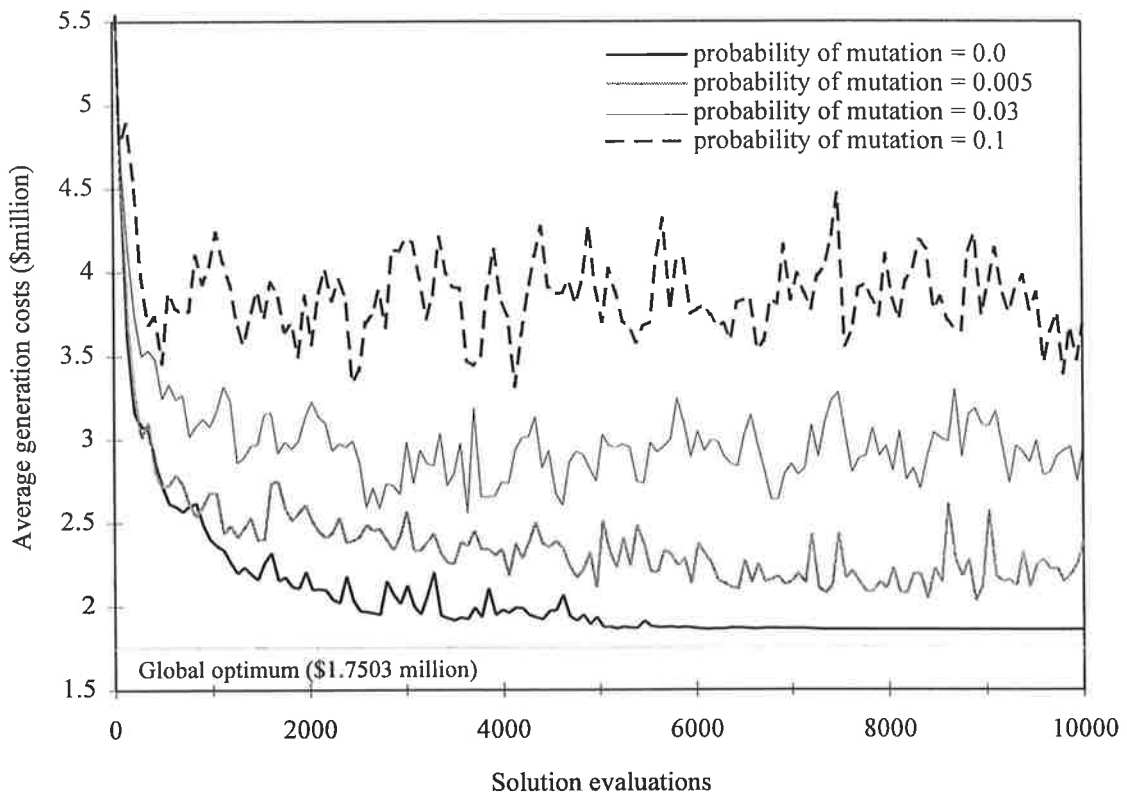


Figure 5.24 Average generation costs for varying probability of mutation, p_m
($N=100$, $p_c=0.7$ and $seed=100$)

The plot of average generation cost using $p_m=0.0$ in Figure 5.24 approaches and eventually equals the best generation cost as the GA run progresses. The selection and crossover operators alone have no further effect on the population after about 6,000 evaluations. The population is dominated by copies of one low cost (but not optimal) solution.

5.4.5 Infeasible network designs

The infeasible network designs determined by the GA model runs in this analysis of the GA parameters are given in Table 5.29. The infeasible network designs may be compared with the lowest cost feasible network designs determined by the exhaustive enumeration and presented in Table 5.4.

Table 5.29 Commonly identified infeasible pipe network configurations

No.	Total cost* (\$m)	Pipe selections (inch diameter)							
		Existing pipes			New pipes				
		[1]	[4]	[5]	[6]	[8]	[11]	[13]	[14]
1	1.8048	leave	dup 14	leave	10	10	10	6	10
2	1.8154	clean	dup 12	leave	12	8	8	6	10
3	1.8460	clean	dup 12	leave	12	8	8	8	10

* including penalty costs

The occurrence of low cost marginally infeasible network designs is an advantage of the GA optimisation technique, particularly for the pipe network optimisation application. The marginally infeasible designs may represent a substantial cost saving for a small pressure head deficiency at a demand node. The total network cost in Table 5.29 is the sum of the pipe material cost and the penalty cost. The pipe costs and penalty costs for the infeasible solutions are provided in Table 5.30. The penalty cost is the product of the pressure head deficiency (psi) at the critical node and some penalty multiplier, k (\$/psi). The penalty multiplier, k , was chosen as $k=\$50,000/\text{psi}$ for the GA runs in the sensitivity analysis after some prior experimentation with the Gessler network. The pressure head deficiencies at the critical node for the critical demand patterns for the infeasible network solutions are given in Table 5.31.

Table 5.30 The costs of the infeasible network solutions

Infeasible network designs	Pipe cost (\$million)	Penalty cost* (\$million)	Total cost (\$million)
1	1.7905	0.0143	1.8048
2	1.7984	0.0170	1.8154
3	1.8205	0.0254	1.8460

* computed using a penalty multiplier, $k=\$50,000/\text{psi}$

Table 5.31 The critical pressure head deficiencies for the infeasible solutions

Infeasible network designs	Critical demand pattern	Critical node	Allowable pressure head (psi)	Actual pressure head (psi)	Pressure head deficiency (psi)
1	2	7	15.0	14.71	0.29
2	2	4	20.0	19.66	0.34
3	2	4	20.0	19.49	0.51

5.4.6 Findings of the sensitivity analysis

The sensitivity analysis of the GA parameters conducted in this chapter indicate good performance of the GA may be achieved with:

- population size, $N=80$ to 200
- probability of crossover, $p_c=0.5$ to 0.9
- probability of mutation, $p_m=0.005$ to 0.03
- and an arbitrary random number generator seed

The findings of the sensitivity analysis compared closely with the suggestions of previous studies. DeJong's (1975) work suggested the use of population sizes of $N=50$ to 200 and Goldberg and Kuo (1987) suggested $N=35$ to 200. The sensitivity analysis performed in this chapter had some success with $N<80$.

Goldberg and Kuo suggested the use of a probability of crossover of $p_c=0.5$ to 1.0. Good performance of the GA is displayed in the sensitivity analysis using a high probability of crossover. DeJong suggested the use of a probability of mutation of $p_m=0.001$ to 0.02. Goldberg and Kuo recommended a $p_m=0.001$ to 0.05 (based on a population size, $N=100$). The sensitivity analysis has emphasised that mutation is a secondary GA mechanism that should be used with low probability. An arbitrarily chosen value of random number seed is suggested. The same seed should be used if the effectiveness of different values of the GA parameters is to be compared.

Only approximate guidelines are established for selection of GA parameters. The selection of the most appropriate GA parameters is an experimental process and usually several GA model runs are required to permit refinement of the selected GA parameters for new problems. It would seem beneficial to choose GA parameters which help maintain a degree of population diversity. The global optimum solution was determined using a population size of $N=20$ and a mutation rate of $p_m=0.0$ which may be an unexpected result. The sensitivity analysis has found the GA is not too highly sensitive to the chosen GA parameters as was also indicated by Goldberg and Kuo (1987).

5.5 Conclusions

A traditional genetic algorithm approach to pipe network optimisation was described in this chapter. The traditional GA approach was applied to the search for the optimal design of the relatively small two-reservoir Gessler pipe network optimisation problem. The Gessler problem was first solved by an exhaustive enumeration to positively identify the optimum configuration. The solution space for the Gessler problem searched by the GA has 16,777,216 possible pipe network solutions if the solutions are represented by a coded string of 24 binary bits. Complete enumeration is possible for this number of alternatives, however when the binary string length exceeds about 30 bits, then the solution space becomes very large and complete enumeration is not possible. The GA technique then becomes particularly useful. Previous attempts to optimise the Gessler problem include the use of partial enumeration (Gessler, 1985) and nonlinear programming (Simpson, Dandy and Murphy, 1994).

A small scale simulation of the traditional genetic algorithm applied to the Gessler problem was performed by hand followed by a number of full scale computer simulated GA evolutions. The GA parameters of populations size, probability of crossover, probability of mutation and random number seed were varied for the full scale GA model runs. A sensitivity analysis of the GA parameters indicates the potential of the traditional GA search, however the global optimum network solution (\$1.7503 million) for the relatively simple Gessler problem is not determined consistently by the GA within the maximum allowed 10,000 solution evaluations. The next chapter investigates various aspects of the traditional GA model formulation to determine if any improvements can be made with respect to pipe network optimisation and specifically, for the GA applied to the Gessler problem.

5 Application of the traditional genetic algorithm to pipe network optimisation

A schema (Holland, 1975) describes a family of coded strings with coding similarities at given positions on the string. Schemata describe coding similarities between strings of the same string length and constructed with the same alphabet.

The order of a schema is the number of fixed positions in the schema. The defining length of a schema is the distance between the outermost specific positions of a schema. The GA exploits the relationship between string similarities and corresponding fitness values in a population of strings. The GA identifies *building blocks* that are schemata of low order and short defining length that are associated with high fitness (Goldberg, 1989). Goldberg considered the effect of reproduction, crossover and mutation on the expected numbers of schemata in subsequent populations and concluded the power of the GA is in the propagation and the combination of building blocks.

The GA selects strings with high fitness more often in reproduction to form successive generations. Hence highly fit string similarities or highly fit schemata survive and accumulate in successive generations.

The offspring strings inherit the combined highly fit string similarities of their parents. Short string similarities or schemata of short defining length and low order are less likely to be disrupted by one-point crossover.

In nature, the weak genetic characteristics of living things are discarded in time while the strong genetic characteristics of living things become prominent. In the GA, low order, short defining length schemata or *building blocks* (Goldberg, 1989) are regenerated by selection or reproduction and are recombined with other building blocks in crossover to construct superior new strings from the best parts of the best old strings. Subsequent populations of coded strings converge on the best regions of the solution space approaching the global optimum solution.

6 Improvements to the Simple Genetic Algorithm for Pipe Network Optimisation

6.1 Introduction

The previous chapter constructed a methodology for applying a traditional genetic algorithm (GA) to water distribution pipe network optimisation. The traditional GA applies three standard GA operators of proportionate selection, one-point crossover and occasional random bit-wise mutations to an evolving population of strings of binary code.

This chapter explores aspects of the GA model formulation to determine what improvements can be made for this application. The effectiveness of the proposed changes to the GA model is measured empirically by application to the Gessler network expansions problem introduced in Chapter 5. The exhaustive enumeration of all the possible solutions to the Gessler problem in Section 5.2 positively identified the best solutions for this problem and led to an improved understanding of the pipe network optimisation solution space to which the GA models will be applied. The experimental analysis performed in this chapter points to advances in the structure of the GA model, to improve the power of the model to search a solution space of this type. Specifically, this chapter trials:

- different penalty methods to incorporate the constraints of the optimisation problem
- the parent selection schemes of proportionate selection and tournament selection
- different string fitness functions (including fitness scaling techniques)
- alternative crossover mechanisms
- various coded string representations
- a creeping mutation operator

6.2 Performance Measures

The traditional GA model was applied to the Gessler network expansions problem for varying GA parameters of population size (N), probability of crossover (p_c), probability of mutation (p_m) and different seeds for the random number generator in Section 5.4. This chapter investigates the performance of modifications to the structure of the GA model developed in Chapter 5 for the optimisation of the Gessler problem. A set of five GA model runs are conducted for each of the modified GA models. This was achieved by using the five sets of the GA parameters shown in Table 6.1. The GA parameter sets were chosen according to the findings of the sensitivity analysis of the GA parameters in the previous chapter (i.e., moderate population sizes, $N=50-100$, high crossover probability, $p_c=0.5-1.0$ and occasional mutations, $p_m=0.005-0.01$).

6 Improvements to the simple GA for pipe network optimisation

The GA parameter sets in Table 6.1 are maintained throughout this study, so that corresponding experiments for each modified GA model are initiated with identical (randomly distributed) starting populations and essentially operate under similar conditions (population size, crossover and mutation rates). The solution space for the Gessler problem considered by the GA consists of about 16.8 million possible network design solutions. All the GA model runs are allowed a maximum of 10,000 solution evaluations (0.06% of the solution space) in which to determine a near-optimum solution. The expected number of generations required by the GA model runs can be calculated by considering the expected number of solution evaluations (new solutions generated by crossover) per generation $(p_c)(N)$.

Table 6.1 The chosen GA parameter sets

FIVE PARAMETER SETS FOR THE GA MODEL RUNS					
GA Parameter	1	2	3	4	5
Population size, N	50	100	100	100	100
Probability of crossover, p_c	0.75	0.75	0.5	1.0	1.0
Probability of random bit-wise mutation, p_m	0.01	0.01	0.01	0.01	0.005
Random number generator seed	1000	1000	1000	1000	1000
Maximum number of solution evaluations	10,000	10,000	10,000	10,000	10,000
Expected evaluations per generation, $(p_c)(N)$	37.5	75	50	100	100
Expected number of generations	267	133	200	100	100

In total, 150 different GA model runs are described in this chapter. A brief summary of the GA model runs is given in Table 6.2. For example, the GA runs designated PEN1, PEN2, . . . , PEN5 are five GA model runs for the five GA parameter sets in Table 6.1 using the traditional GA formulation established in Chapter 5 with some variation to the penalty coefficient. The GA model runs in **bold** in Table 6.2 use the traditional GA formulation exactly as described in Chapter 5. For example, GA runs PEN11, FIT1, CODE1, CROSS1 and CREEP1 represent the same GA run (the results are identical). The GA runs using the traditional GA act as a benchmark for all the GA model runs.

The GA model runs required an average of 10 seconds of CPU computing time for 10,000 solution evaluations (30,000 steady-state hydraulic analyses) on a SUN SPARCstation-10. As network size and complexity increases, the computational effort to perform a GA run will increase significantly. However, computers are getting faster. Only 2 years ago, the GA model runs applied to the Gessler problem required 9 minutes for 10,000 solution evaluations on a SUN 4/280 computer.

Table 6.2 Summary of the GA model runs performed in this chapter

GA model runs	Description
PENALTY FUNCTIONS AND PENALTY COST MULTIPLIER, k (Section 6.3)	
PEN1- PEN5	Penalty costs PC_{max} based on the maximum violations of pressure constraints for each loading condition (Eq. 6.1) with $k = \$25,000/\text{psi}$
PEN11-PEN15	Penalty costs PC_{max} with $k = \\$50,000/\text{psi}$
PEN21-PEN25	Penalty costs PC_{max} with $k = \$75,000/\text{psi}$
PEN31-PEN35	Penalty costs PC_{sum} based on the sum of all violations of pressure constraints for each loading condition (Eq. 6.2) with $k = \$25,000/\text{psi}$
PEN41-PEN45	Penalty costs PC_{sum} with $k = \$50,000/\text{psi}$
PEN51-PEN55	Penalty costs PC_{sum} with $k = \$75,000/\text{psi}$
PEN61-PEN65	Penalty costs PC_{max} with k increasing gradually as the GA run proceeds from $\$10,000/\text{psi}$ to $\$100,000/\text{psi}$
SELECTION METHODS, FITNESS FUNCTIONS AND FITNESS SCALING (Section 6.4)	
FIT1-FIT5	Proportionate (roulette-wheel) selection with inverse fitness function and fitness scaling exponent $n = 1$
FIT11-FIT15	Proportionate selection with inverse fitness function and $n = 2$
FIT21-FIT25	Proportionate selection with linear fitness function and $n = 1$
FIT31-FIT35	Proportionate selection with linear fitness function and $n = 2$
FIT41-FIT45	Proportionate selection with inverse fitness function and increasing fitness scaling exponent as GA run proceeds from $n = 1$ to $n = 4$
FIT51-FIT55	Proportionate selection with linear fitness function and increasing fitness scaling exponent as GA run proceeds from $n = 1$ to $n = 4$
FIT61-FIT65	Binary ($s=2$ competitors) tournament selection and better individual of the two selected with probability $p_t = 1.0$
FIT71-FIT75	Binary tournament selection and better individual selected with $p_t = 0.9$
FIT81-FIT85	Binary tournament selection and better individual selected with $p_t = 0.8$
FIT91-FIT95	Ternary ($s=3$ competitors) tournament selection ($p_t = 1.0$)
CODED STRING REPRESENTATIONS (Section 6.5)	
CODE1-CODE5	Concatenated decision-variable substrings of binary codes
CODE11-CODE15	Substrings of Gray codes
CODE21-CODE25	Coded strings of integers
CODE31-CODE35	Substrings of binary codes with an alternative arrangement of decision-variable substring positions

Table 6.2 cont. Summary of the GA model runs performed in this chapter

GA model runs	Description
CROSSOVER MECHANISMS (Section 6.6)	
CROSS1-CROSS5	One-point crossover
CROSS11-CROSS15	Two-point crossover
CROSS21-CROSS25	Four-point crossover
CROSS31-CROSS35	Uniform crossover
CROSS41-CROSS45	One-point crossover at decision variable substring boundaries
CROSS51-CROSS55	Two-point crossover at decision variable substring boundaries
CROSS61-CROSS65	Uniform crossover at decision variable substring boundaries
CREEPING MUTATION OPERATOR (Section 6.7)	
CREEP1-CREEP5	No creep, $p_a = 0.0$
CREEP11-CREEP15	Probability of creeping mutation, $p_a=0.0625$ with probability of creeping down, $p_d = 0.5$
CREEP21-CREEP25	Probability of creeping mutation, $p_a=0.125$ with $p_d = 0.5$
CREEP31-CREEP35	Probability of creeping mutation, $p_a=0.25$ with $p_d = 0.5$
CREEP41-CREEP45	Probability of creeping mutation, $p_a=0.125$ with $p_d = 0.25$
CREEP51-CREEP55	Probability of creeping mutation, $p_a=0.125$ with $p_d = 0.75$

Note. The **bold** GA model runs all represent the same set of five GA runs. These GA runs use the traditional GA formulation and act as a bench mark for all the GA model runs.

The performance of the genetic algorithm model runs are measured in a number of ways. Ultimately, the success of a GA run is measured by its ability to reach the optimal solution. The two global optima for the Gessler problem were identified by exhaustive enumeration in Section 5.2. The 50 lowest cost (feasible) solutions to the Gessler problem were presented in Table 5.4. In most cases, the GA runs performed in this study arrive at one of these 50 designs. The lowest cost solution (feasible or infeasible) identified by each GA run is observed including the number of generations (and solution evaluations) required to determine this solution.

Some population cost statistics are recorded for each generation of the GA model including the lowest cost solution in the generation and the average generation cost. The lowest average generation cost achieved and the generation at which this is achieved is recorded. In addition, the performance measures used by DeJong (1975) in his pioneering study of GAs for function optimisation, offline performance and online performance, are recorded as the GA run proceeds. Offline performance is essentially a running average cost of the lowest cost solutions in each generation up to the current generation, and online performance is essentially a running

6 Improvements to the simple GA for pipe network optimisation

average cost of all solutions evaluated by the genetic algorithm search up to the current generation (Goldberg, 1989).

The ultimate offline performance and ultimate online performance are observed (after 10,000 solution evaluations have been performed). For most of the GA model runs, the offline performance and online performance improve throughout the GA run up to the last generation performed. Offline performance measures convergence to the best solutions and is best suited to offline applications in which it is important to determine the best alternative and online performance measures ongoing performance for online applications in which it is more important to achieve acceptable performance quickly (Goldberg, 1989).

Other measurements are made for some of the GA model runs in this study, including the variations of the number of infeasible solutions in each generation for alternative penalty methods, the average cost of strings selected for mating by the alternative selection procedures (compared to the average generation cost) and other performance measures of specific facets of the GA formulation. The parents of the optimal solution (if reached) are noted.

A stopping criteria for the GA model runs is not used. Instead, all the GA model runs are allowed to proceed for a maximum of 10,000 solution evaluations. A suitable stopping criteria could be established (for the Gessler problem) by performing further GA model runs and observing the changes in performance conditions such as average generation cost, offline performance and online performance throughout the GA run. The rate of improvement of these conditions when (and if) the global optimum solution to the Gessler problem is reached could help decide when to terminate future GA runs.

6.3 Penalty Functions

The pipe network optimisation problem is a constrained optimisation problem. The pipe network designs are subject to large sets of hydraulic constraints, system performance constraints and other general design constraints. General design constraints such as minimum pipe diameters are usually allowed for in the coded representation of the solution space. The genetic algorithm model, like many other useful pipe network optimisation models satisfy the hydraulic constraints by linking the optimisation model to a hydraulic simulation model. The simulation model balances pipe flows and node pressures for each of the demand conditions. The system performance constraints such as minimum allowable node pressures are compared with the results of the hydraulic simulation. For example, the Gessler network expansions are subject to three patterns of projected (instantaneous) demands and the proposed designs are required to achieve a minimum pressure head profile (Table 5.2).

In the implementation of the GA applied to pipe network optimisation, a penalty function approach is used such that the objective function is based on the network expansion costs and penalty costs. The penalty function applies a penalty cost to *infeasible* network solutions (those that do not satisfy system performance constraints). Other system performance constraints (such as maximum pressure heads, maximum velocity constraints and acceptable tank water level variations during extended period simulation) and some general design constraints may also be considered in the GA search by way of a penalty function(s).

A genetic algorithm search of a constrained objective function will often apply harsh penalties to infeasible solutions which effectively sets the fitness to zero. The pipe network optimisation problem is usually a highly constrained problem and the feasible region (the part of the search space where the constraints are met) may be a relatively small fraction. For example, of the 3.981 million possible solutions to the Gessler problem¹, 3.294 million are infeasible. In addition, the infeasible region may isolate small parts of the feasible region. The global optimum solution is expected to occur at the boundary of the feasible region, adjacent to infeasible solutions. For these reasons, penalties applied to infeasible solutions are a function of the distance from feasibility, such that penalties increase as the quality of the solution decreases. In this way, infeasible solutions can play an important role in the GA search and the GA search profits by the (often) valuable coded information provided by infeasible solutions.

¹ Gessler (1985) originally considered a solution space of 3.981 million solutions, however there are actually 16.8 million (2^{24}) solutions within the solution space we have constructed for the GA search.

Richardson et al. (1989) recommended the use of penalties which are functions of the distance from feasibility. They found penalties which are a function of the degree by which the constraints are violated are better than those which are functions of the number of violated constraints. For the pipe network optimisation problem, the distance from feasibility is an indication of the quality of the solution.

In practical problems such as pipe network optimisation, there are other advantages in maintaining infeasible solutions. The pipe network designs which are only marginally infeasible are often useful designs and in some cases, allowing for factors such as the accuracy of the hydraulic analyses and uncertainties such as the prediction of future demand conditions, designs which are only just infeasible may actually be the designer's preferred alternative.

The penalty function should not be too harsh, otherwise the GA will be too concerned with satisfying the pressure constraints and useful infeasible solutions may be discarded in the selection process. Furthermore, the GA will be less likely to find isolated parts of the feasible region. Alternatively, if the penalty function is too light, the GA search may drift to many different parts of the infeasible region without identifying the best feasible regions. The purpose of the following analysis is to construct a pressure violation penalty function, such that the GA search may approach the optimum solution from both the feasible and infeasible regions of the solution space simultaneously.

6.3.1 Penalty functions for pipe network optimisation

The penalty applied to infeasible pipe network solutions is a linear function of the magnitude of the violations of the minimum pressure constraints. Two penalty functions for pipe network optimisation are considered in this study. The first penalty function in Eq. 6.1 computes a pressure violation penalty cost (PC_{max}) that is the product of the maximum violation of the pressure constraints for each loading pattern (l) and a specified penalty multiplier (k). The alternative penalty function in Eq. 6.2 computes a pressure violation penalty cost (PC_{sum}) that is the product of the sum of all the violations of the pressure constraints for all loading patterns (l) and a penalty multiplier (k). Multiple demand patterns may be considered and the system may fail to achieve the minimum allowable pressures at a number of demand points for any demand condition. Clearly, if the pressure constraints are met, no penalty is applied.

$$PC_{max} = \begin{cases} k \cdot \sum_{l=1}^L \left[\max_m (H_m^{min} - H_m) \right] & \text{for demand nodes } m : H_m < H_m^{min} \\ 0 & \text{if for all } m : H_m \geq H_m^{min} \end{cases} \quad (6.1)$$

$$PC_{sum} = \begin{cases} k \cdot \sum_{l=1}^L \left[\sum_m (H_m^{min} - H_m) \right] & \text{for demand nodes } m : H_m < H_m^{min} \\ 0 & \text{if for all } m : H_m \geq H_m^{min} \end{cases} \quad (6.2)$$

in which m = nodes where the minimum allowable pressure constraints are not satisfied for loading pattern l , H_m^{min} = minimum allowable pressure at node m for loading pattern l (appropriate pressure units), H_m = measured pressure at node m for loading pattern l , L = the number of loading patterns and k = pressure violation penalty multiplier (\$/pressure unit deficit).

6.3.2 The penalty multiplier

The penalty multiplier is problem dependent and should be chosen carefully. Some trial and error adjustment of the penalty multiplier may be required for new problems and for new penalty functions for different design situations.

Richardson et al. (1989) regarded infeasible solutions as incomplete solutions and suggested good penalty functions can be constructed in terms of the expected completion cost (that is, the cost increase to make an infeasible solution feasible). The pressure violation penalty cost may be thought of as the cost of achieving hydraulic feasibility and the penalty multiplier may be thought of as the cost of improving the pressure at the failing node by one unit. The pressure deficiencies are not easy to cost, but penalty costs should be measured on the same scale as network expansion costs.

Consider node **12** of the Gessler network at the extreme downstream end of the system. The exhaustive enumeration showed that this node is one of the most critical, often failing to meet the pressure requirements for demand pattern GE3 (Figure 5.3). The pipe [14] which connects node **12** to the system is a new pipe to be sized. The optimum solution 1 (Table 5.4) sizes pipe [14] as a 10" pipe and the resultant pressure at node **12** is 19.47psi (the pressure at node **11** immediately upstream is 25.94psi and the headloss in the 10" pipe [14] is 24.951ft according to Tables 5.5 and 5.6). Increasing the diameter of pipe [14] (Table 6.3), gives an indication of the increased pipe costs to improve the pressure at node **12**. The flow in

6 Improvements to the simple GA for pipe network optimisation

pipe [14] is 1.782cfs for demand pattern GE3, the length of pipe [14] is 5,280ft and the Hazen-Williams roughness is 120. The calculations in Table 6.3 show that it costs between \$10,000 and \$25,000 to improve the pressure at node 12 by 1 psi. In the following GA model runs, penalty coefficients between $k=\$10,000/\text{psi}$ and $k=\$100,000/\text{psi}$ are tested.

Table 6.3 Extra pipe costs for pipe [14] for a 1 psi improvement in pressure at the critical node 12 (for the optimal solution 1 subject to demand pattern GE3)

New pipe diameter (in)	Headloss in pipe [14] ¹ (ft)	Pressure at node 12 ² (psi)	Unit cost of new pipe (\$/ft)	Total cost of new pipe ³ (\$)	Extra cost (compared with new 10" pipe) for a 1 psi pressure improvement at node 12
10	24.951	19.47	28.9	\$152,592	-
12	10.267	25.82	40.5	\$213,840	\$9,639/psi
14	4.8461	28.17	52.1	\$275,088	\$14,080/psi
16	2.529	29.17	59.4	\$313,632	\$16,597/psi
18	1.425	29.65	70.5	\$372,240	\$21,575/psi
20	0.853	29.90	80.0	\$422,400	\$25,873/psi

¹ for $L_{14}=5,280\text{ft}$ and $C_{14}=120$ and assuming $Q_{14}=1.782\text{cfs}$ (for critical demand pattern GE3)

² assuming pressure at node 11 is 25.94psi (see results for demand pattern GE3 in Table 5.6)

³ for $L_{14}=5,280\text{ft}$

The value of the penalty multiplier should be selected such that near-optimal infeasible solutions cost slightly more than the optimal solution. Of course, the optimal solution is not usually known, but the number of feasible solutions in the populations of the genetic algorithm search and the feasibility of the lowest cost solution obtained by the genetic algorithm search are the best experimental pointers to the suitability of the chosen penalty multiplier.

6.3.3 Varying the penalty multiplier

The starting population of strings is expected to contain a high number of infeasible solutions, since infeasible solutions make up a substantial proportion of the solution space. Many of the randomly generated solutions in the starting population will be highly infeasible. As the selection operator will act quickly to reject these solutions, the starting population will be effectively much smaller. The value of the penalty multiplier may be increased as the GA model run proceeds, from a low value in the diverse early generations to allow a wide search, to a higher value late in the GA run in order to close in on the feasible region. Richardson et al. (1989) suggested there could be some value in varying the penalty coefficient as the GA model run proceeds, starting with relaxed constraints and tightening them as the GA run progresses, and recommended that future studies investigate this further.

Siedlecki and Sklansky (1989) proposed a modified genetic algorithm search to encourage exploration of the feasible region and the promising parts of the infeasible region at a given distance from the feasible region boundary. They devised an algorithm to adjust the penalty coefficient each generation before the reproduction stage, so as to influence which solutions are selected for the mating pool. In this way, they could control the placement of the population in the solution space.

Siedlecki and Sklansky (1989) considered a constrained optimisation problem in which the criterion to be optimised introduced a uniform orientation in the search space and where the minima was expected to occur on the boundary of the feasible region. Optimisation problems of this type are often encountered in the design of statistical pattern classifiers. The variable penalty coefficient method was found to be superior to a fixed penalty coefficient. The method avoids the need for trial and error determination of suitable penalty coefficients.

6.3.4 GA model runs to compare penalty functions and penalty multipliers

The three sets of GA model runs designated PEN1-PEN5, PEN11-PEN15 and PEN21-PEN25 (refer to Table 6.2) used the penalty function in Eq. 6.1 (with penalty costs a linear function of the *maximum* violations of the pressure constraints for each demand condition) for three values of the penalty multiplier, k . The GA model runs PEN1-PEN5 used a penalty multiplier, $k=\$25,000/\text{psi}$, GA runs PEN11-PEN15 used $k=\$50,000/\text{psi}$ and GA runs PEN21-PEN25 used $k=\$75,000/\text{psi}$.

The next three sets of GA model runs PEN31-PEN35, PEN41-PEN45 and PEN51-PEN55 used the penalty function given in Eq. 6.2 (with penalty costs a linear function of the *sum* of all violations of the pressure constraints for all demand conditions) for the values of the penalty multiplier, $k=\$25,000/\text{psi}$, $k=\$50,000/\text{psi}$ and $k=\$75,000/\text{psi}$ respectively.

Finally, the GA model runs PEN61-PEN65 used the penalty function based on the *maximum* violations of the pressure constraints (Eq. 6.1) and increased the penalty multiplier with solution evaluations from $k=\$10,000/\text{psi}$ to $\$100,000/\text{psi}$ according to Table 6.4 (increasing $\$10,000/\text{psi}$ for every 1,000 solution evaluations).

The results of all the GA model runs PEN1-PEN65 using alternative penalty function are summarised in Tables 6.5-6.11.

Table 6.4 Variation of penalty multiplier for GA model runs PEN61-PEN65

Number of evaluations performed	Penalty multiplier, k (\$/psi)
0 - 1,000	10,000
1,000 - 2,000	20,000
2,000 - 3,000	30,000
3,000 - 4,000	40,000
4,000 - 5,000	50,000
5,000 - 6,000	60,000
6,000 - 7,000	70,000
7,000 - 8,000	80,000
8,000 - 9,000	90,000
9,000 - 10,000	100,000

Table 6.5 Search results for genetic algorithm model runs PEN1-PEN5

Penalty costs based on maximum violations of minimum allowable pressure constraints, PC_{max} (Eq. 6.1) for each demand condition with penalty multiplier, $k=\$25,000/\text{psi}$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	PEN1 $N=50$, $p_c=0.75$	PEN2 $p_c=0.75$	PEN3 $p_c=0.5$	PEN4	PEN5 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.8115 (92)	1.8010 (127)	1.7503 [‡] (72)	1.7976 [†] (88)	1.8390 (82)
Lowest cost GA design generated (Table 5.4)	11	8	2	1 [†]	31
Lowest average generation cost (\$m) (after - generations)	2.167 (153)	2.284 (114)	2.009 (148)	2.348 (98)	2.115 (95)
Ultimate offline performance (\$m)	2.049	1.965	1.875	1.965	1.980
Ultimate online performance (\$m)	2.511	2.613	2.340	2.684	2.547
Average infeasible solutions / population	49%	57.9%	33.1%	43.5%	39.3%
Maximum infeasible solutions /population	94%	76%	75%*	75%*	75%*
Minimum infeasible solutions / population	14%	30%	7%	24%	12%

* Number of infeasible solutions in starting population

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

[†] Infeasible design. The solution cost includes the penalty cost. The infeasible design is given in Table 5.29.

GA run PEN4 determined a best feasible design for \$1.8010m (Solution 9 in Table 5.4) after 94 generations.

Table 6.6 Search results for genetic algorithm model runs PEN11-PEN15

Penalty costs PC_{max} with $k=\$50,000/\text{psi}$					
GA RUNS	PEN11	PEN12	PEN13	PEN14	PEN15
Unless specified otherwise GA parameters $N=100, p_c=1.0$ and $p_m=0.01$	$N=50,$ $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m)	1.7503 [‡]	1.7503 [‡]	1.7503 [‡]	1.7725	1.8807
(after - generations)	(59)	(96)	(59)	(93)	(82)
Lowest cost GA design generated (Table 5.4)	2	2	1	3	-
Lowest average generation cost (\$m)	2.045	2.139	2.038	2.271	2.274
(after - generations)	(212)	(73)	(176)	(92)	(86)
Ultimate offline performance (\$m)	1.963	1.896	1.916	1.935	2.026
Ultimate online performance (\$m)	2.560	2.557	2.475	2.766	2.694
Average infeasible solutions / population	19.4%	15.8%	21.3%	38.6%	26.7%
Maximum infeasible solutions /population	72%*	75%*	75%*	75%*	75%*
Minimum infeasible solutions / population	2%	1%	4%	21%	7%

* Number of infeasible solutions in starting population

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.7 Search results for genetic algorithm model runs PEN21-PEN25

Penalty costs PC_{max} with $k=\$75,000/\text{psi}$					
GA RUNS	PEN21	PEN22	PEN23	PEN24	PEN25
Unless specified otherwise GA parameters $N=100, p_c=1.0$ and $p_m=0.01$	$N=50,$ $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m)	1.7999	1.7503 [‡]	1.7503 [‡]	1.7503 [‡]	1.8010
(after - generations)	(153)	(26)	(122)	(52)	(94)
Lowest cost GA design generated (Table 5.4)	6	1	2	2	8
Lowest average generation cost (\$m)	2.109	2.217	1.981	2.284	2.123
(after - generations)	(242)	(122)	(190)	(81)	(96)
Ultimate offline performance (\$m)	1.989	1.915	1.980	1.950	1.951
Ultimate online performance (\$m)	2.746	2.673	2.590	2.917	2.698
Average infeasible solutions / population	16.0%	18.7%	14.6%	23.2%	19.8%
Maximum infeasible solutions /population	72%*	75%*	75%*	75%*	75%*
Minimum infeasible solutions / population	2%	2%	1%	6%	4%

* Number of infeasible solutions in starting population

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.8 Search results for genetic algorithm model runs PEN31-PEN35

Penalty costs based on the sum of all violations of minimum allowable pressure constraints, PC_{sum} (Eq. 6.2) for each demand condition with penalty multiplier, $k=\$25,000/\text{psi}$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	PEN31 $N=50$, $p_c=0.75$	PEN32 $p_c=0.75$	PEN33 $p_c=0.5$	PEN34	PEN35 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.9510 (164)	1.7503 [‡] (64)	1.7503 [‡] (173)	1.7503 [‡] (56)	1.8010 (54)
Lowest cost GA design generated (Table 5.4)	-	1	1	1	7
Lowest average generation cost (\$m) (after - generations)	2.307 (244)	2.252 (103)	2.025 (172)	2.240 (83)	2.107 (77)
Ultimate offline performance (\$m)	2.132	1.904	1.893	1.905	1.997
Ultimate online performance (\$m)	2.756	2.880	2.484	2.830	2.745
Average infeasible solutions / population	48.0%	32.3%	22.1%	33.9%	35.1%
Maximum infeasible solutions /population	90%	75%*	75%*	75%*	75%*
Minimum infeasible solutions / population	16%	15%	5%	9%	9%

* Number of infeasible solutions in starting population

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Lowest cost solutions

The GA model runs PEN41-PEN45 (penalty function PC_{sum} with $k=\$50,000/\text{psi}$) were most successful, achieving the global optimum on four occasions. GA runs PEN11-PEN15 (PC_{max} with $k=\$50,000/\text{psi}$), PEN21-PEN25 (PC_{max} with $k=\$75,000/\text{psi}$) and PEN31-PEN35 (PC_{sum} with $k=\$25,000/\text{psi}$) each achieved the global optimum on three occasions from five attempts.

Of the five GA model runs PEN61-PEN65 (increasing k), three have determined a lowest cost solution that is infeasible, while the other two GA runs determine the global optimum solution (Table 6.11). The penalties are too light early in the GA run, not sufficiently penalising infeasible solutions which are not close to the best feasible regions.

The least effective GA parameter sets were the first GA parameter set (using the smaller population size, $N=50$) and the fifth GA parameter set (using the lower mutation rate, $p_m=0.005$).

Table 6.9 Search results for genetic algorithm model runs PEN41-PEN45

Penalty costs PC_{sum} with $k=\$50,000/\text{psi}$					
GA RUNS	PEN41	PEN42	PEN43	PEN44	PEN45
Unless specified otherwise GA parameters $N=100, p_c=1.0$ and $p_m=0.01$	$N=50,$ $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m)	1.7999	1.7503 [‡]	1.7503 [‡]	1.7503 [‡]	1.7503 [‡]
(after - generations)	(170)	(45)	(124)	(68)	(88)
Lowest cost GA design generated (Table 5.4)	6	1	1	2	2
Lowest average generation cost (\$m)	2.292	2.155	2.079	2.496	2.108
(after - generations)	(119)	(127)	(120)	(92)	(98)
Ultimate offline performance (\$m)	2.038	1.877	1.926	2.022	2.000
Ultimate online performance (\$m)	2.879	2.977	2.743	3.292	2.963
Average infeasible solutions / population	26.6%	20.1%	18.7%	29.5%	15.2%
Maximum infeasible solutions /population	72%*	75%*	75%*	75%*	75%*
Minimum infeasible solutions / population	2%	6%	3%	17%	2%

* Number of infeasible solutions in starting population

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.10 Search results for genetic algorithm model runs PEN51-PEN55

Penalty costs PC_{sum} with $k=\$75,000/\text{psi}$					
GA RUNS	PEN51	PEN52	PEN53	PEN54	PEN55
Unless specified otherwise GA parameters $N=100, p_c=1.0$ and $p_m=0.01$	$N=50,$ $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m)	1.8010	1.8115	1.8115	1.7503 [‡]	1.7503 [‡]
(after - generations)	(91)	(97)	(53)	(94)	(71)
Lowest cost GA design generated (Table 5.4)	9	10	10	1	1
Lowest average generation cost (\$m)	2.132	2.177	1.984	2.405	2.016
(after - generations)	(102)	(64)	(172)	(94)	(83)
Ultimate offline performance (\$m)	2.029	1.963	1.913	2.031	1.902
Ultimate online performance (\$m)	3.137	3.527	3.081	3.551	3.113
Average infeasible solutions / population	13.2%	15.4%	13.0%	17.4%	15.7%
Maximum infeasible solutions /population	72%*	75%*	75%*	75%*	75%*
Minimum infeasible solutions / population	2%	5%	1%	5%	2%

* Number of infeasible solutions in starting population

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.11 Search results for genetic algorithm model runs PEN61-PEN65

Penalty costs PC_{max} with increasing penalty multiplier, k gradually (and uniformly over the 10,000 evaluations) as the GA run proceeds from \$10,000/psi to \$100,000/psi					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	PEN61 $N=50$, $p_c=0.75$	PEN62 $p_c=0.75$	PEN63 $p_c=0.5$	PEN64	PEN65 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.7962 [†] (35)	1.7503 [‡] (105)	1.9970 [†] (17)	1.7503 [‡] (79)	1.8324 [†] (96)
Lowest cost GA design generated (Table 5.4)	1 [†]	1	-	2	2 [†]
Lowest average generation cost (\$m) (after - generations)	2.043 (192)	2.106 (64)	2.384 (171)	2.286 (92)	2.140 (96)
Ultimate offline performance (\$m)	1.919	1.900	2.154	1.914	1.970
Ultimate online performance (\$m)	2.487	2.569	2.672	2.797	2.514
Average infeasible solutions / population	25.4%	29.3%	64.0%	35.7%	48.4%
Maximum infeasible solutions /population	86%	75%*	94%	85%	87%
Minimum infeasible solutions / population	0%	8%	17%	6%	13%

* Number of infeasible solutions in starting population

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

† Infeasible designs. The solution costs include penalty costs. The infeasible designs are given in Table 5.29. GA run PEN61 determined a best feasible design for \$1.8010m (Solution 7 in Table 5.4) after 104 generations, PEN63 determined a best feasible design for \$2.0898m after 108 generations and PEN65 determined a best feasible design for \$1.9288m after 24 generations.

Best of generation costs and offline performance

Figures 6.1-6.3 show the best *feasible* solution costs in each generation against the number of solution evaluations performed for GA runs PEN4, PEN14, . . . , PEN64. By comparison to Figure 6.3, Figure 6.4 shows the best of generation costs (least cost *feasible* or *infeasible* individuals in each population) for GA runs PEN14, PEN44 and PEN64.

The GA run PEN24 (high value of k =\$75,000/psi) is most effective for penalties based on the maximum violations of the pressure constraints (Figure 6.1). In contrast, the GA run PEN34 (low k =\$25,000/psi) is most effective for penalties based on the sum of all the pressure constraint violations (Figure 6.2).

The GA run PEN64 with increasing penalties and an average penalty multiplier of k =\$55,000/psi is quite effective by comparison to GA runs PEN14 and PEN44 with fixed

6 Improvements to the simple GA for pipe network optimisation

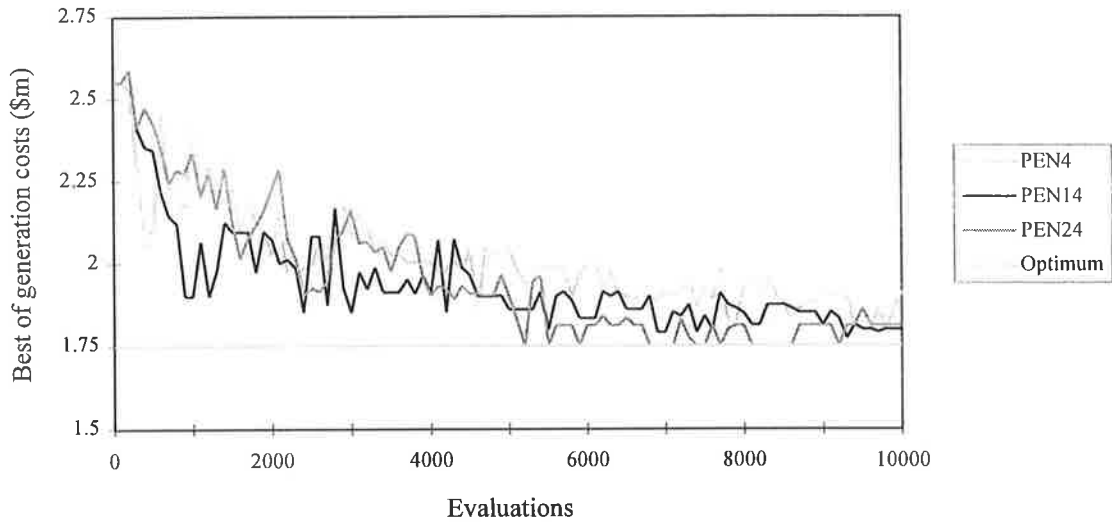


Figure 6.1 Best generation costs (feasible solutions only) for GA runs PEN4 ($k=\$25,000/\text{psi}$), PEN14 ($k=\$50,000/\text{psi}$) and PEN24 ($k=\$75,000/\text{psi}$) with penalties PC_{max} based on *maximum* violations of pressure constraints for each demand pattern

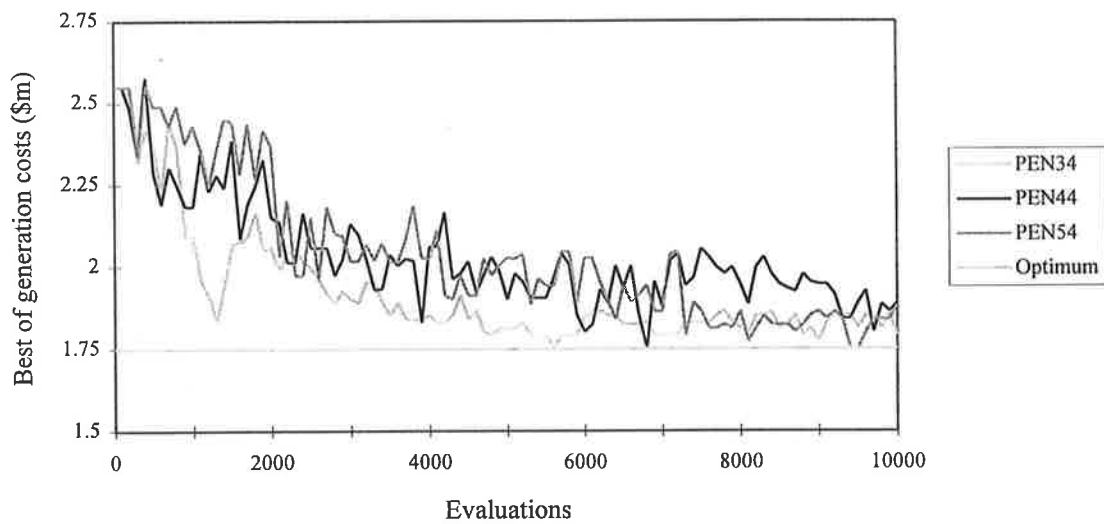


Figure 6.2 Best generation costs (feasible solutions only) for GA runs PEN34 ($k=\$25,000/\text{psi}$), PEN44 ($k=\$50,000/\text{psi}$) and PEN54 ($k=\$75,000/\text{psi}$) with penalties PC_{sum} based on *sum* of violations of pressure constraints for each demand pattern

6 Improvements to the simple GA for pipe network optimisation

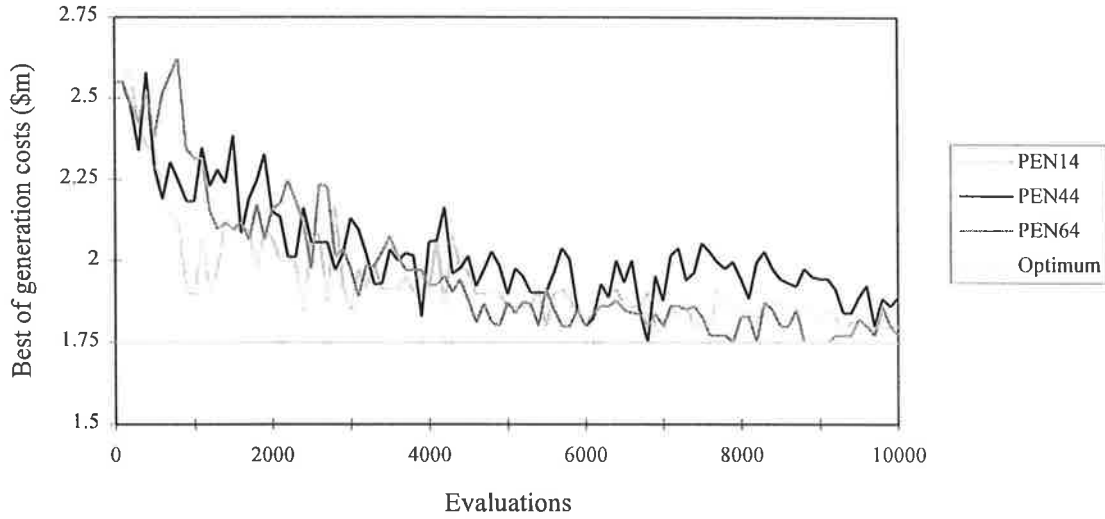


Figure 6.3 Best generation costs (feasible solutions only) for GA runs PEN14 ($k=\$50,000/\text{psi}$ with penalties PC_{max}), PEN44 ($k=\$50,000/\text{psi}$ with penalties PC_{sum}) and PEN64 (increasing k with penalties PC_{max})

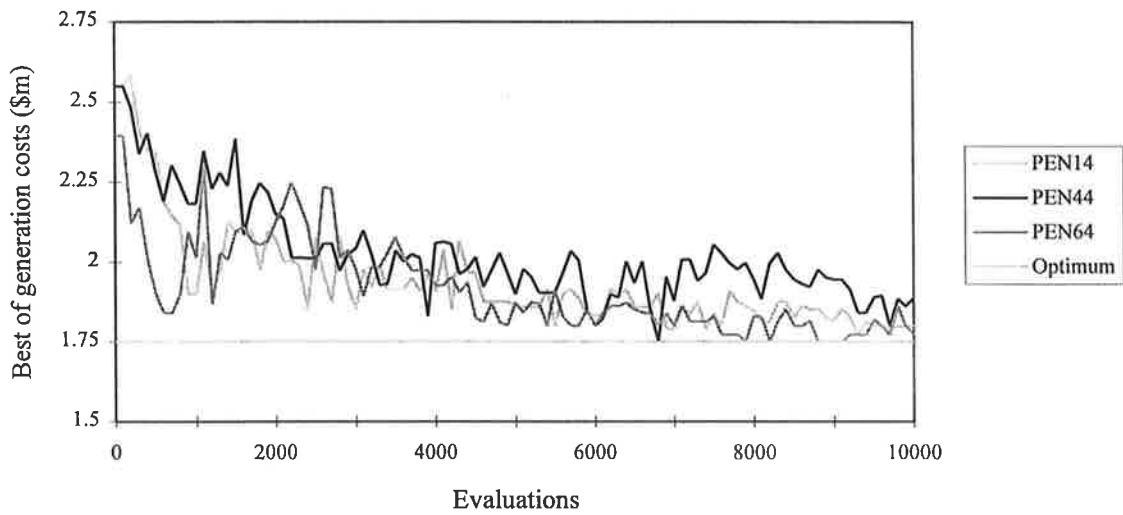


Figure 6.4 Best generation costs (feasible and infeasible solutions) for GA runs PEN14 ($k=\$50,000/\text{psi}$ with penalties PC_{max}), PEN44 ($k=\$50,000/\text{psi}$ with penalties PC_{sum}) and PEN64 (increasing k with penalties PC_{max})

6 Improvements to the simple GA for pipe network optimisation

$k=\$50,000/\text{psi}$ (Figure 6.3). The GA run PEN64 is one of two successful GA runs with varying penalty multiplier k . Figure 6.4 indicates that GA run PEN64 obtains low cost solutions after only a few generations, however Figure 6.3 shows that the low cost solutions are infeasible.

The most consistent performers, in terms of offline performance (running average of best generation costs), are GA runs PEN11-PEN15 (PC_{max} with $k=\$50,000/\text{psi}$) and GA runs PEN31-PEN35 (PC_{sum} with $k=\$25,000/\text{psi}$).

Average generation costs and online performance

Figures 6.5-6.7 show average generation costs (average of feasible and infeasible solutions in the population) against evaluations performed for GA model runs PEN4, . . . , PEN64. The plots of average generation cost for GA runs PEN34, PEN44 and PEN54 using the penalty function based on the sum of all the pressure constraint violations (Figure 6.6) are relatively discontinuous and occasionally average generation costs are very high (as high as \$4 million)².

Sometimes a particularly disruptive mutation or crossover can produce a highly infeasible solution. The penalty function that sums all the violations of allowable pressure heads for all loading cases can compute unreasonably high penalty costs. Poor decisions for influential upstream pipes will cause all of the pressure constraints to be violated downstream of this pipe, regardless of downstream pipe decisions. The downstream pipe sizes may even be optimal. These solutions are harshly penalised (a penalty cost for all the pressure constraint violations) and are not likely to be selected in the reproduction process.

The penalty function based on the maximum pressure constraint violations avoids this situation and still accounts for poorly selected pipe sizes. If an upstream pipe size is inadequate, but downstream pipe sizes are optimal, the deficiency in pressure head exists for all downstream nodes but does not grow. If the downstream pipe sizes are also inadequate, the deficiency in pressure head accumulates so that the maximum violation of the pressure constraints occurs at a node further from the reservoir. If a downstream pipe size only is inadequate the penalty costs obtained by both functions may be similar.

The GA runs using the third GA parameter set (Table 6.1) with the lower crossover rate, $p_c=0.5$ (less disruption by crossover) typically provide the lowest average generation costs and superior online performance.

² In terms of network expansion costs only, the most expensive solution has a cost of \$4.0772 million and the least expensive solution has a cost of 0.3986 million.

6 Improvements to the simple GA for pipe network optimisation

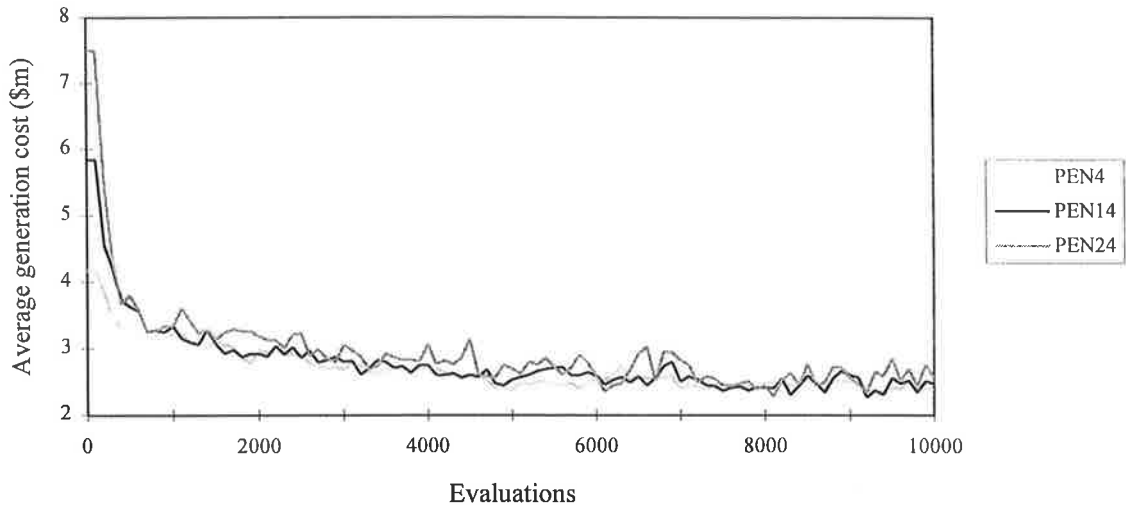


Figure 6.5 Average generation costs (feasible and infeasible) for GA runs PEN4 ($k=\$25,000/\text{psi}$), PEN14 ($k=\$50,000/\text{psi}$) and PEN24 ($k=\$75,000/\text{psi}$) with penalties based on *maximum* pressure constraint violations

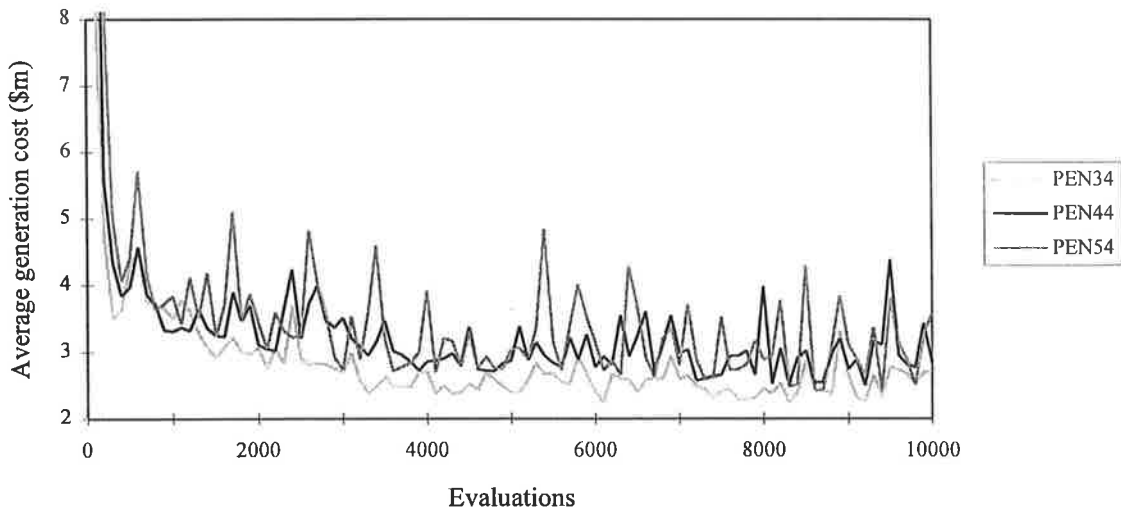


Figure 6.6 Average generation costs (feasible and infeasible) for GA runs PEN34 ($k=\$25,000/\text{psi}$), PEN44 ($k=\$50,000/\text{psi}$) and PEN54 ($k=\$75,000/\text{psi}$) with penalties based on *sum* of all pressure constraint violations

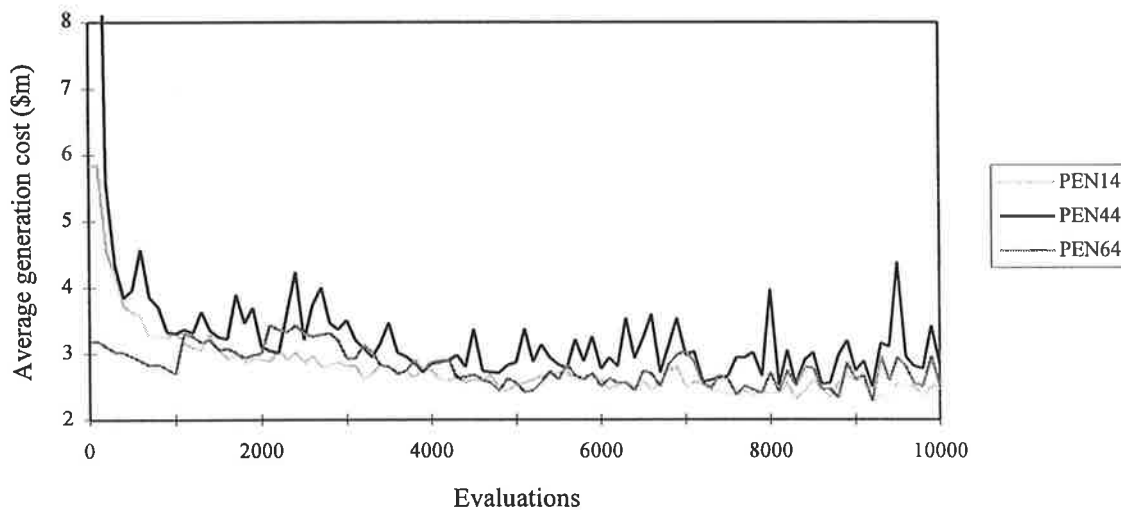


Figure 6.7 Average generation costs (feasible and infeasible) for GA runs PEN14 ($k=\$50,000/\text{psi}$ with penalties PC_{max}), PEN44 ($k=\$50,000/\text{psi}$ with penalties PC_{sum}) and PEN64 (increasing k with penalties PC_{max})

Number of infeasible solutions in the populations

Figures 6.8-6.10 show the variation of the number of infeasible solutions in the fixed population size of $N=100$ members for GA runs PEN4, PEN14, . . . , PEN64. Typically, 75% of the randomly generated solutions in the starting populations for the GA model runs applied to the Gessler problem are infeasible solutions. The number of infeasible solutions tends to drop significantly in the first few generations (with the exception of GA run PEN64 with the low initial value of k).

As expected, the average number of infeasible solutions maintained in a population is decreased with increasing penalty multiplier. The use of penalty costs based on the maximum constraint violations maintains more infeasible solutions than penalty costs based on the sum of all the constraint violations.

The average number of infeasible solutions per population for the GA runs PEN61-PEN65 (increasing k) range from an average of 29% for GA run PEN62 up to an average of 64% for GA run PEN63. The GA run PEN63 maintains high numbers of infeasible solutions and the lowest cost solution obtained by that GA run is infeasible and a long way from the global optimal solution. As previously discussed, it is important to maintain reasonable numbers of infeasible solutions in the GA search (although perhaps not as many as GA run PEN63). Further experimentation is required as to methods for applying suitable adjustments to the penalty multiplier as the GA run proceeds, to regulate the numbers of infeasible solutions.

An alternative approach may be to separate the populations into two subpopulations of feasible and infeasible solutions for the selection process. Specified numbers of feasible solutions and infeasible solutions may be selected for the new generation. The probability of selection of a feasible solution from the subpopulation of feasible solutions may be determined by the fitness of the feasible solution relative to the other feasible solutions. Similarly, the probability of selection of infeasible solutions may be determined by the fitness of the infeasible solution relative only to other infeasible solutions. Using such a procedure, may enable the GA search to hillclimb in the feasible region and infeasible region simultaneously. For highly constrained optimisation problems with very small feasible regions, the starting population could be randomly selected from within the feasible region only.

6 Improvements to the simple GA for pipe network optimisation

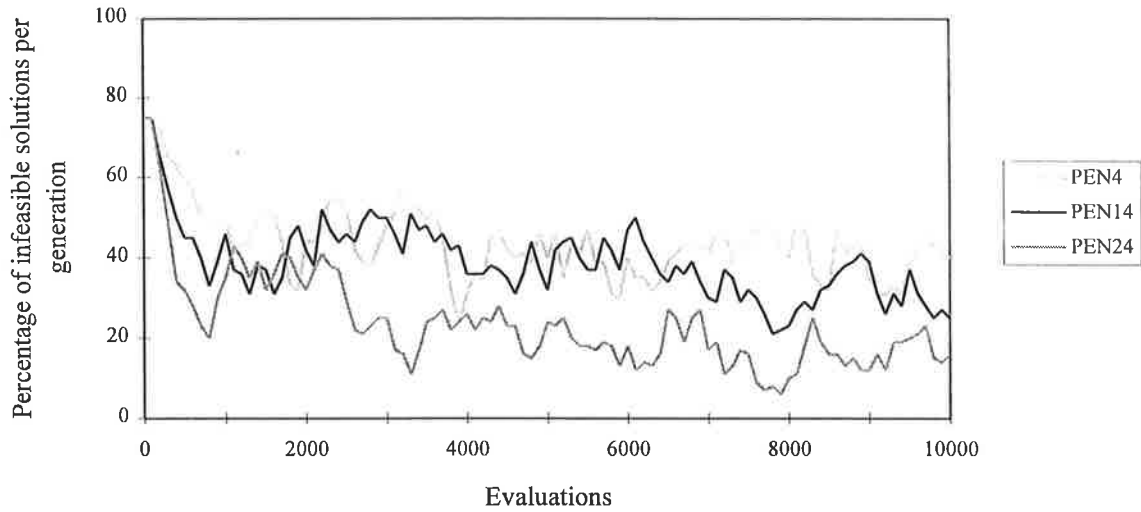


Figure 6.8 Infeasible solutions in the populations (population size, $N=100$) for GA runs PEN4 ($k=\$25,000/\text{psi}$), PEN14 ($k=\$50,000/\text{psi}$) and PEN24 ($k=\$75,000/\text{psi}$) with penalties based on *maximum* constraint violations

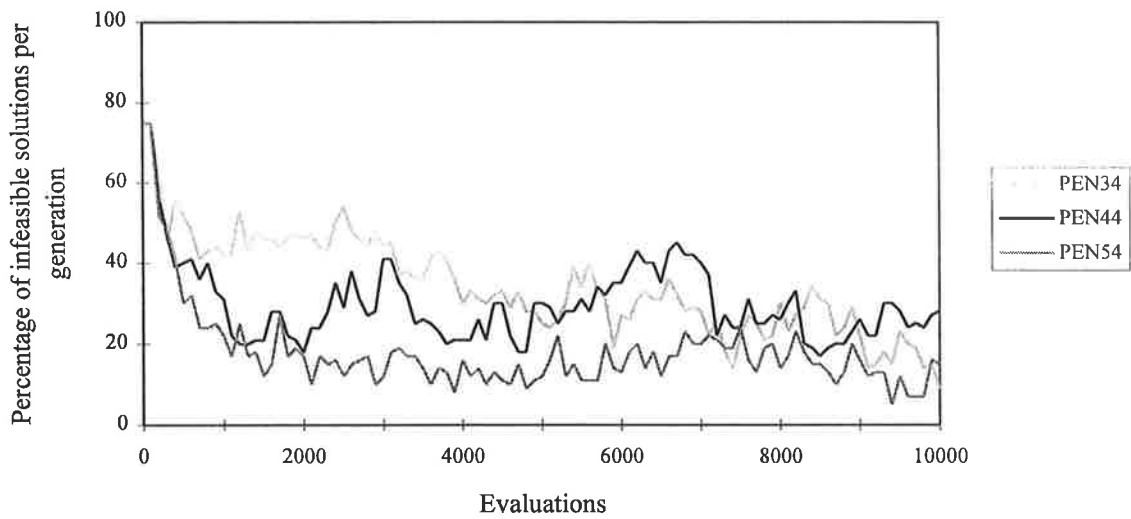


Figure 6.9 Infeasible solutions in the populations ($N=100$) for GA runs PEN34 ($k=\$25,000/\text{psi}$), PEN44 ($k=\$50,000/\text{psi}$) and PEN54 ($k=\$75,000/\text{psi}$) with penalties based on *sum* of all constraint violations

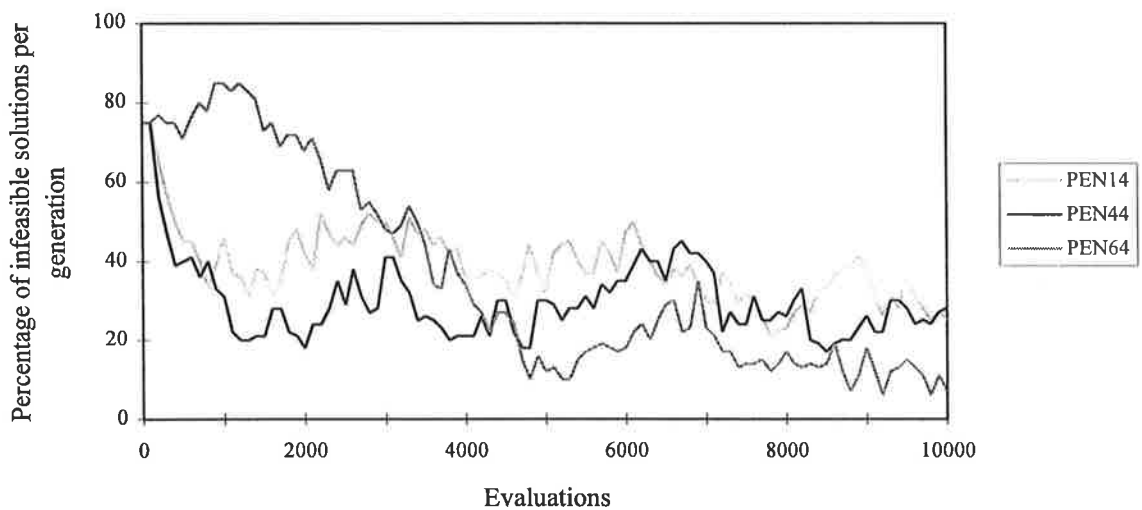


Figure 6.10 Infeasible solutions in the populations for GA runs PEN14 ($k=\$50,000/\text{psi}$ with penalties PC_{max}), PEN44 ($k=\$50,000/\text{psi}$ with penalties PC_{sum}) and PEN64 (increasing k as GA run proceeds with penalties PC_{max})

Parents of the optimal solution

The parents of the global optimal solution (pipe costs \$1.7503 million) were recorded for all the GA runs in this chapter, if the optimum solution was determined. The parents were found to be either two feasible solutions, or (not quite as regularly) one feasible solution and one infeasible solution, but never two infeasible solutions. As an example, the parents of the global optimum solutions obtained by traditional GA runs PEN11, PEN12 and PEN13 are shown below. In each case, the optimal strings may have been obtained from the parent strings by any one of a number of combinations of crossovers and random bit-wise mutations. Mutation was involved in the creation of the optimal strings for GA runs PEN11 and PEN12.

GA run PEN11:		Pipe costs	Penalty costs
Optimal solution:	000-110-000-011-001-010-000-001	\$1.7503m	-
Parent 1:	000-110-000-010-000-010-011-001	\$1.8010m	\$2.112m
Parent 2:	000-110-100-011-001-011-000-001	\$1.964m	-
GA run PEN12:		Pipe costs	Penalty costs
Optimal solution:	000-110-000-011-001-010-000-001	\$1.7503m	-
Parent 1:	000-110-010-011-001-010-000-110	\$2.109m	-
Parent 2:	011-111-010-011-001-010-000-001	\$2.299m	-
GA run PEN13:		Pipe costs	Penalty costs
Optimal solution:	000-110-000-011-001-001-000-010	\$1.7503m	-
Parent 1:	000-110-000-011-000-001-001-011	\$1.8115m	\$1.244m
Parent 2:	001-110-110-100-001-001-000-010	\$2.326m	-

6.3.5 Recommendations for the penalty function

In general, the higher values of the penalty coefficient ($k \geq \$50,000/\text{psi}$) show superior performance for the case where the penalty function depends on the maximum violations of the pressure constraints. Conversely, lower values of the penalty coefficient ($k \leq \$50,000/\text{psi}$) are more suitable when the penalty function depends on the sum of all violations of the pressure constraints. The GA runs using the penalty function which sums all the pressure constraint violations is effective in determining the global optimum, however excessive penalties are applied to some infeasible solutions. There is potentially some value in varying the penalty multiplier as the GA run proceeds, however further testing is required to determine the best method of adjusting the penalty multiplier.

6.4 Selection Methods and Fitness Functions

The power of the GA lies in the trade-off between exploitation of past results by selection and the exploration of new parts of the search space by the crossover and mutation mechanisms. The critical GA operator of selection (or reproduction) exploits past experience in the solution space. Selection simulates a Darwinian survival-of-the-fittest process, regulating the survival of solutions in the search by allowing better (or fitter) solutions a greater chance of survival.

The coded string solutions are evaluated and assigned a measure of fitness which reflects the merit of the trial solution. In the pipe network optimisation problem we try to minimise cost, while in the GA search we try to maximise fitness. The fittest individual in the GA solution space should be the global optimum solution. The fitness is some function of the objective function value. For pipe network optimisation, the fitness is a function of the cost and the hydraulic feasibility (accounted for by penalty costs) of the proposed pipe network design.

The selection operator uses an appropriate method to choose fitter solutions from within a competing population of solutions to generate the new population. The selection methods of proportionate selection and tournament selection are compared in this study. The traditional GA uses a proportionate selection scheme, although there is much theoretical and empirical support for tournament selection. The following experimental analysis is intended to help establish a suitable fitness function and a reliable and efficient parent selection scheme for the GA application to pipe network optimisation.

6.4.1 Proportionate selection

Proportionate selection (also called stochastic selection or biased roulette-wheel selection) chooses strings according to their fitness with respect to the fitness of fellow strings in the population and chance factors. The fitness allocated to a string allows the string an appropriate chance of selection in the proportionate reproduction process. The strings are selected randomly (with replacement) from a competing population of N strings with the probability of selection, p_i of string i given by:

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad \text{for coded strings, } i=1, \dots, N \quad (6.3)$$

where f_i = the fitness of coded string i

A number of methods have been proposed to sample this probability distribution (Goldberg and Deb, 1991) including roulette-wheel (Monte Carlo) selection, stochastic remainder selection and stochastic universal selection. The method used in the traditional GA we have described in Chapter 5 for pipe network optimisation is weighted roulette-wheel sampling (DeJong, 1975). The probability of selection of each individual in the population is computed by Eq. 6.3. The computer algorithm allocates real intervals (or slots) between 0.0 and 1.0 to each individual equal to their probability of selection. The computer generates a random number between 0.0 and 1.0 (a spin) and the individual occupying this interval is chosen. Spins are performed until the new population of N strings has been selected.

Roulette-wheel selection selects individuals according to their fitness and there is a degree of luck. By comparison, in stochastic sampling methods the number of offspring an individual receives in the new population is determined by (and is approximately equal to) the number of expected offspring in the new population. There are combinations and variations of roulette-wheel sampling and stochastic sampling. In this research, only proportionate selection by the traditional method of roulette-wheel selection has been considered.

6.4.2 Tournament selection

The tournament selection approach is as follows:

- (1) Choose two or more individuals for the tournament from the current population (usually without replacement). The number of individuals is the tournament size, s . Binary tournaments ($s=2$ competitors) (Brindle, 1981) and ternary tournaments ($s=3$ competitors) are considered in this study.
- (2) The fittest individual among the competitors is selected as the tournament winner. In a variant described by Goldberg and Deb (1991), the fittest individual in a binary tournament is selected as the winner with some probability, p_t (such that $0.5 < p_t \leq 1.0$).
- (3) Repeat steps (1) and (2) until the new population is filled. A number of passes through the current population will be required.

Tournament selection has an advantage in that a fitness function is not required to transform the pipe network cost minimisation problem to a coded string fitness maximisation problem. The coded string solution with the lowest cost wins the tournament. As a consequence, biases introduced by the form of the fitness function do not influence the GA search.

Goldberg and Deb (1991) provided a comprehensive theoretical analysis of a number of selection schemes used in GAs including proportionate reproduction, tournament selection, ranking selection and Genitor (or 'steady-state') selection and made recommendations for the use of selection methods in practical applications. They compared the expected performance of

the selection schemes on the basis of the solutions to deterministic finite difference equations that describe the change in proportion of different classes of individual in time, and by estimates of the takeover time and expected growth ratio of the fittest individual. The takeover time is the time when the population contains $N-1$ of the best individuals. The expected growth ratio of the fittest individual is the number of times the fittest individual is selected from the population, divided by the number of copies of the fittest individual in the population.

Goldberg and Deb (1991) found tournament selection has a significantly higher expected performance than proportionate reproduction. Tournament selection can achieve higher growth ratios as tournament size increases. The expected performance of proportionate selection can be improved by fitness scaling techniques. The expected growth ratio of proportionate selection is higher in the early generations and quite low in the late generations and this is the reason for the use of fitness scaling mechanisms. An exponential fitness scaling function is suggested to control the degree of competition. Goldberg and Deb made useful recommendations for the correct use of selection methods in GAs to balance the conflict between exploitation and exploration, to prevent premature convergence and to maintain diversity in populations.

6.4.3 Fitness functions

In the implementation of the GA to the pipe network optimisation problem, the fitness function is a relationship between the fitness of coded strings and the objective function value (total cost) of the trial pipe network designs. The total cost of a network solution is the sum of the pipe network expansion costs and penalty costs (for violating system performance constraints). The fitness function maps the total cost of network solutions to be minimised, to fitness values of coded strings to be maximised by the GA search. The fitness values should not be negative. Each string in a GA population of N strings is assigned a value of fitness.

The two alternative fitness functions considered in this study are given by Eqs. 6.4 and 6.5 in which the fitness of string i , f_i is a function of the total cost of string i , $cost_i$. The first is a simple inverse relationship between fitness and objective function value to some power n . The second fitness function is a linear fitness function (Goldberg, 1989) in which the fitness of individuals in a population are normalised with respect to the cost of the worst individual (the maximum cost string) in the population, denoted by $cost_{max}$.

$$f_i = \left(\frac{1}{cost_i} \right)^n \quad \text{for coded strings, } i=1, \dots, N \quad (6.4)$$

$$f_i = (cost_{max} - cost_i)^n \quad \text{for coded strings, } i=1, \dots, N \quad (6.5)$$

Measuring fitness values from a cost extreme such as the maximum cost string should give an increasing progression of fitness values, however, it may be more appropriate to measure fitness relative to some other reference cost other than the maximum cost in the current population, such as the average cost or the minimum cost in the current population or some previous population. In some cases, an additional transformation may be required so that fitness values are non-negative.

The exponent n in the fitness functions may be held constant throughout the GA run (say $n=1$ or $n=2$) or n may vary as the GA run proceeds and in this case the exponent n acts as a fitness scaling mechanism.

6.4.4 Fitness scaling mechanisms

In this research, a form of power law fitness scaling is tested. The exponent (n) in the fitness function is increased in magnitude as the GA model run proceeds to stretch the range of fitness values.

Goldberg (1989) reported on fitness scaling mechanisms used to adjust the raw fitness values to maintain appropriate levels of competition between the strings in a population. Fitness scaling mechanisms regulate the number of copies of strings selected for the mating pool. Goldberg believed the choice of the value of the exponent is usually problem-dependent and the value of the exponent can be changed to shrink or stretch the range of fitness values.

Gillies³ (1985) considered a power law form of fitness scaling where the scaled fitness is some power (n) of the raw fitness. Gillies used a value of $n=1.005$ in a machine vision application.

The starting population of strings is generated randomly and as a consequence this population and early generations are likely to contain diverse strings of genetic code. The average string fitness in the starting population will be low, and the individual string fitnesses may vary significantly, yet all of the strings may possess potentially valuable genetic information. A low value of the exponent n should be employed at the start of the GA run, say $n = 1$, so the GA can sort through the potential strengths of the ordinary strings in the early generations. A low value of n preserves population diversity in the early generations. A large value of the exponent n is not appropriate here since the GA search may be misguided by encouraging one better than average (but not near-optimal) string to dominate the new populations.

Over a number of generations, populations are likely to tend towards homogeneity to the point of convergence and fitness levels may become very similar. In time, a large number of useful

³ Reported by Goldberg (1989).

string similarities are recognised and become established. The value of the exponent n may be increased in steps during the intermediate generations. As the GA run progresses further, the highly fit string similarities that have evolved begin to dominate the populations. The strings are constructed of similar genetic code and their magnitudes of raw fitness may be very similar. A high value of the exponent n , say 3 or 4, is needed late in the GA run to accentuate the small differences in string fitness.

6.4.5 Experimental analysis of selection methods and fitness functions

In the following analysis, ten sets of GA model runs are performed to compare the selection methods of proportionate reproduction and tournament selection and various forms of the fitness function. The ten sets of five GA runs designated FIT1-FIT5, FIT11-FIT15, . . . , FIT91-FIT95 are described in Table 6.2. The first six sets of GA runs use proportionate selection and the remaining four sets of GA runs use tournament selection.

The first two sets of GA model runs FIT1-FIT5 and FIT11-FIT15 adopt the inverse fitness function given by Eq. 6.4. The GA runs FIT1-FIT5 use a fixed value of the exponent, $n=1$ while GA runs FIT11-FIT15 use a fixed value of the exponent, $n=2$. The next two sets of GA runs FIT21-FIT25 and FIT31-FIT35 adopt the linear fitness function given by Eq. 6.5 for fixed exponents $n=1$ and $n=2$ respectively.

The final two sets of GA runs employing the proportionate selection scheme FIT41-FIT45 and FIT51-FIT55 increase the fitness scaling exponent, n throughout the GA run according to Table 6.12. The GA runs FIT41-FIT45 use the inverse fitness function and GA runs FIT51-FIT55 use the linear fitness function.

Table 6.12 Variation of fitness scaling exponent, n for GA runs FIT41-FIT45 (inverse fitness function) and FIT51-FIT55 (linear fitness function)

Fitness scaling exponent, n	Solution evaluations performed
1	0 - 2,500
2	2,500 - 5,000
3	5,000 - 7,500
4	7,500 - 10,000

6 Improvements to the simple GA for pipe network optimisation

The GA model runs FIT61-FIT65 select the winners of binary tournaments ($s=2$ competitors) for the reproduction process. The GA runs FIT71-FIT75 and GA runs FIT81-FIT85 select the better individual in binary tournaments with probability $p_t=0.9$ and $p_t=0.8$ respectively. Finally, the GA runs FIT91-FIT95 choose the winners of a ternary tournaments ($s=3$ competitors) for the reproduction process.

The results of this series of GA runs are given in Tables 6.13-6.22.

Best cost solutions

The GA model runs FIT41-FIT45 (proportionate selection using the inverse fitness function with fitness scaling) and GA runs FIT81-FIT85 (binary tournament selection with $p_t=0.8$) both achieve the global optimum solution (\$1.7503m) in all of the five GA runs.

For proportionate selection, the inverse fitness function achieves the global optimum on 11 occasions from 15 attempts. The inverse fitness function is considerably superior to the linear fitness function which achieves the global optimum on only 3 occasions from 15 GA runs. The GA runs which scaled fitness values by varying the exponent in the fitness functions achieved the global optimum more often than the GA runs which fixed the exponent.

The binary tournament selection is extremely effective, achieving the global optimum solution on 13 occasions from 15 attempts. In addition, binary tournament selection is very efficient, in most cases determining the global optimum after less than 3,000 solution evaluations of the maximum 10,000 evaluations. The higher probabilities of selecting the best individual $p_t=1.0$ (in GA runs FIT61-FIT65) and $p_t=0.9$ (GA runs FIT71-FIT75) fail to achieve the global optimum when population size is small, $N=50$. The most effective GA parameters for tournament selection are the larger population size, $N=100$, the high crossover rate, $p_c=1.0$, and high mutation rate, $p_m=0.01$. We suspect the small populations are dominated too quickly by better than average solutions that are not necessarily near-optimal. The ternary tournament selection (3 competitors) finds the global optimum solution just once, using the high crossover rate and high mutation rate. The relentless exploitation of past results of ternary tournament selection is best combined with increased exploration by crossover and mutation to provide more opportunities for useful variations in the converging populations. Ternary tournament selection is remarkably efficient, identifying low cost solutions in only 10 to 15 generations.

Table 6.13 Search results for genetic algorithm model runs FIT1-FIT5

Proportionate (roulette-wheel) selection with replacement using the inverse fitness function (Eq.6.4) and fitness scaling exponent, $n=1$. GA runs FIT1-FIT5 are equivalent to GA runs PEN11-PEN15.					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	FIT1 $N=50$, $p_c=0.75$	FIT2 $p_c=0.75$	FIT3 $p_c=0.5$	FIT4	FIT5 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.7503 [‡] (59)	1.7503 [‡] (96)	1.7503 [‡] (59)	1.7725 (93)	1.8807 (82)
Lowest cost GA design generated (Table 5.4)	2	2	1	3	-
Lowest average generation cost (\$m) (after - generations)	2.045 (212)	2.139 (73)	2.038 (176)	2.271 (92)	2.274 (86)
Ultimate offline performance (\$m)	1.963	1.896	1.916	1.935	2.026
Ultimate online performance (\$m)	2.560	2.557	2.475	2.766	2.694
Average cost of all strings selected (\$m)	2.421	2.394	2.373	2.599	2.569

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.14 Search results for genetic algorithm model runs FIT11-FIT15

Proportionate (roulette-wheel) selection using inverse fitness function and fitness scaling exponent, $n=2$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	FIT11 $N=50$, $p_c=0.75$	FIT12 $p_c=0.75$	FIT13 $p_c=0.5$	FIT14	FIT15 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.7725 (211)	1.7503 [‡] (34)	1.8897 (170)	1.7503 [‡] (51)	1.7503 [‡] (48)
Lowest cost GA design generated (Table 5.4)	3	2	-	1	1
Lowest average generation cost (\$m) (after - generations)	1.885 (253)	1.940 (83)	2.039 (140)	2.041 (87)	1.945 (94)
Ultimate offline performance (\$m)	1.850	1.811	2.004	1.839	1.946
Ultimate online performance (\$m)	2.240	2.323	2.325	2.427	2.444
Average cost of all strings selected (\$m)	2.047	2.099	2.218	2.199	2.282

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.15 Search results for genetic algorithm model runs FIT21-FIT25

Proportionate (roulette-wheel) selection with replacement using the linear fitness function (Eq.6.5) and fitness scaling exponent, $n=1$					
GA RUNS	FIT21	FIT22	FIT23	FIT24	FIT25
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.8781 (204)	2.058 (107)	1.8131 (34)	1.9400 [†] (52)	1.8115 (82)
Lowest cost GA design generated (Table 5.4)	-	-	14	-	12
Lowest average generation cost (\$m) (after - generations)	2.127 (192)	2.452 (119)	2.058 (194)	2.514 (72)	2.331 (97)
Ultimate offline performance (\$m)	2.061	2.214	1.967	2.101	2.050
Ultimate online performance (\$m)	2.518	2.901	2.466	2.967	2.785
Average cost of all strings selected (\$m)	2.395	2.766	2.361	2.820	2.663

[†] Infeasible design. The solution cost includes the penalty cost. GA run FIT24 determined a best cost feasible design for \$1.9420 million after 44 generations.

Table 6.16 Search results for genetic algorithm model runs FIT31-FIT35

Proportionate (roulette-wheel) selection using the linear fitness function and fitness scaling exponent, $n=2$					
GA RUNS	FIT31	FIT32	FIT33	FIT34	FIT35
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.7503 [‡] (139)	1.7910 (74)	1.8390 (41)	1.8390 (77)	1.8385 (31)
Lowest cost GA design generated (Table 5.4)	2	5	32	33	27
Lowest average generation cost (\$m) (after - generations)	1.885 (200)	2.244 (133)	1.921 (91)	2.191 (96)	2.088 (98)
Ultimate offline performance (\$m)	1.883	1.955	1.876	1.955	1.920
Ultimate online performance (\$m)	2.269	2.682	2.207	2.622	2.453
Average cost of all strings selected (\$m)	2.106	2.497	2.092	2.451	2.316

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.17 Search results for genetic algorithm model runs FIT41-FIT45

Proportionate (roulette-wheel) selection using the inverse fitness function and a gradually increasing (uniformly over 10,000 evaluations) fitness scaling exponent as GA run proceeds from $n=1$ to $n=4$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	FIT41 $N=50$, $p_c=0.75$	FIT42 $p_c=0.75$	FIT43 $p_c=0.5$	FIT44	FIT45 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.7503 [‡] (59)	1.7503 [‡] (80)	1.7503 [‡] (59)	1.7503 [‡] (52)	1.7503 [‡] (99)
Lowest cost GA design generated (Table 5.4)	2	1	1	1	1
Lowest average generation cost (\$m) (after - generations)	1.799 (237)	1.925 (122)	1.794 (172)	1.968 (83)	1.914 (96)
Ultimate offline performance (\$m)	1.857	1.856	1.869	1.878	1.943
Ultimate online performance (\$m)	2.298	2.356	2.272	2.546	2.527
Average cost of all strings selected (\$m)	2.121	2.153	2.147	2.321	2.343

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.18 Search results for genetic algorithm model runs FIT51-FIT55

Proportionate (roulette-wheel) selection using the linear fitness function and a gradually increasing fitness scaling exponent as GA run proceeds from $n=1$ to $n=4$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	FIT51 $N=50$, $p_c=0.75$	FIT52 $p_c=0.75$	FIT53 $p_c=0.5$	FIT54	FIT55 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.8390 (227)	2.0069 (64)	1.7503 [‡] (121)	1.7910 (100)	1.7503 [‡] (48)
Lowest cost GA design generated (Table 5.4)	33	-	1	5	2
Lowest average generation cost (\$m) (after - generations)	1.951 (235)	2.120 (133)	1.877 (183)	2.268 (87)	1.924 (81)
Ultimate offline performance (\$m)	2.053	2.142	1.911	1.998	1.958
Ultimate online performance (\$m)	2.444	2.726	2.367	2.825	2.544
Average cost of all strings selected (\$m)	2.302	2.579	2.234	2.635	2.409

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.19 Search results for genetic algorithm model runs FIT61-FIT65

Binary ($s=2$ competitors) tournament selection and best individual selected with probability, $p_t=1.0$					
GA RUNS	FIT61	FIT62	FIT63	FIT64	FIT65
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	135	197	100	100
Lowest solution cost (\$m)	1.7910	1.7503‡	1.7503‡	1.7503‡	1.7503‡
(after - generations)	(51)	(18)	(22)	(20)	(15)
Lowest cost GA design generated (Table 5.4)	5	1	1	1	1
Lowest average generation cost (\$m)	1.8060	1.792	1.758	1.858	1.760
(after - generations)	(235)	(134)	(155)	(57)	(75)
Ultimate offline performance (\$m)	1.824	1.790	1.780	1.803	1.801
Ultimate online performance (\$m)	2.079	2.088	1.984	2.231	2.071
Average cost of all strings selected (\$m)	1.859	1.863	1.825	1.924	1.889

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.20 Search results for genetic algorithm model runs FIT71-FIT75

Binary tournament selection and best individual selected with probability, $p_t=0.9$					
GA RUNS	FIT71	FIT72	FIT73	FIT74	FIT75
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	268	133	200	100	100
Lowest solution cost (\$m)	1.8154†	1.7503‡	1.7503‡	1.7503‡	1.7503‡
(after - generations)	(227)	(18)	(20)	(34)	(18)
Lowest cost GA design generated (Table 5.4)	2 †	1	1	1	1
Lowest average generation cost (\$m)	1.848	1.791	1.766	1.875	1.787
(after - generations)	(220)	(120)	(166)	(53)	(89)
Ultimate offline performance (\$m)	1.875	1.801	1.787	1.837	1.817
Ultimate online performance (\$m)	2.122	2.171	2.041	2.367	2.180
Average cost of all strings selected (\$m)	1.959	1.955	1.887	2.063	1.984

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

† Infeasible design. The solution cost includes the penalty cost. The design is given in Table 5.29.

GA model run FIT71 determined a best cost feasible design for \$1.8897 million after 14 generations.

Table 6.21 Search results for genetic algorithm model runs FIT81-FIT85

Binary tournament selection and best individual selected with probability, $p_t=0.8$					
GA RUNS	FIT81	FIT82	FIT83	FIT84	FIT85
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	268	133	200	100	100
Lowest solution cost (\$m)	1.7503 [‡]	1.7503 [‡]	1.7503 [‡]	1.7503 [‡]	1.7503 [‡]
(after - generations)	(55)	(93)	(32)	(31)	(26)
Lowest cost GA design generated (Table 5.4)	1	1	1	1	2
Lowest average generation cost (\$m)	1.782	1.827	1.770	1.996	1.871
(after - generations)	(97)	(110)	(166)	(46)	(85)
Ultimate offline performance (\$m)	1.793	1.842	1.803	1.837	1.843
Ultimate online performance (\$m)	2.207	2.334	2.152	2.566	2.356
Average cost of all strings selected (\$m)	2.011	2.131	1.994	2.289	2.157

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.22 Search results for genetic algorithm model runs FIT91-FIT95

Ternary ($s=3$ competitors) tournament selection					
GA RUNS	FIT91	FIT92	FIT93	FIT94	FIT95
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	268	133	198	100	100
Lowest solution cost (\$m)	1.8115	1.8115	1.8010	1.7503 [‡]	1.7999
(after - generations)	(10)	(17)	(11)	(13)	(13)
Lowest cost GA design generated (Table 5.4)	10	10	8	1	6
Lowest average generation cost (\$m)	1.816	1.834	1.808	1.827	1.813
(after - generations)	(84)	(53)	(38)	(44)	(68)
Ultimate offline performance (\$m)	1.823	1.840	1.822	1.782	1.837
Ultimate online performance (\$m)	2.089	2.162	1.990	2.149	2.095
Average cost of all strings selected (\$m)	1.836	1.875	1.839	1.831	1.882

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Best of generation costs and offline performance

Figures 6.11-6.13 plot best of generation costs against the number of solution evaluations performed (100 solution evaluations per generation) for the alternative selection operators and fitness functions for GA runs FIT4, FIT14, . . . , FIT94. The plots of the best generation costs for GA runs using proportionate selection and the inverse fitness function in Figure 6.11 are clearly superior to those of the GA runs using proportionate selection and the linear fitness function in Figure 6.12. The performance GA runs using tournament selection in Figure 6.13 cannot be faulted. The binary tournament selection is very effective. The GA run FIT94 is the only time ternary tournament selection achieves the optimum solution, however it does so after only 1,300 solution evaluations.

Figures 6.14-6.16 depict offline performance (running average of best generation costs) for the alternative selection operators and fitness functions. The offline performances of GA runs FIT4 (proportionate selection using inverse fitness function with fixed $n=1$) and FIT44 (proportionate selection using inverse fitness function with increasing n) are identical for the first 2,500 evaluations since $n=1$ for both GA runs up to this point. Thereafter, the effect of increasing the value of the exponent n for GA run FIT44 is evident. The proportionate selection method using the inverse fitness function consistently achieves superior ultimate offline performance compared to the linear fitness function. The fixed value of $n=2$ with its headstart in the first 2,500 evaluations always achieves a better offline performance compared to the increasing n for the linear fitness function, however, on occasions the increasing n achieves an equivalent ultimate offline performance for the inverse fitness function.

The GA runs using tournament selection achieve very low values of ultimate offline performance (i.e., high performance). The populations are dominated by the best individuals for most of the GA runs. The use of tournament selection with small populations and of ternary tournament selection without adequate crossover and mutation rates can cause the populations to be dominated by good solutions other than the optimum solution.

Average generation costs and online performance

Figures 6.17-6.19 plot average generation cost against the number of solution evaluations and Figures 6.20-6.22 plot online performance for the GA runs using alternative selection processes. The use of the proportionate selection with the inverse fitness function and scaled fitness values achieves quite low average generation costs. The average generation costs determined with tournament selection are lower than those determined with proportionate selection. The lowest average generation costs achieved by tournament selection are in most cases only a fraction more than the best cost solution obtained, indicating the populations have been almost completely dominated by this solution. The plots of average generation costs and

6 Improvements to the simple GA for pipe network optimisation

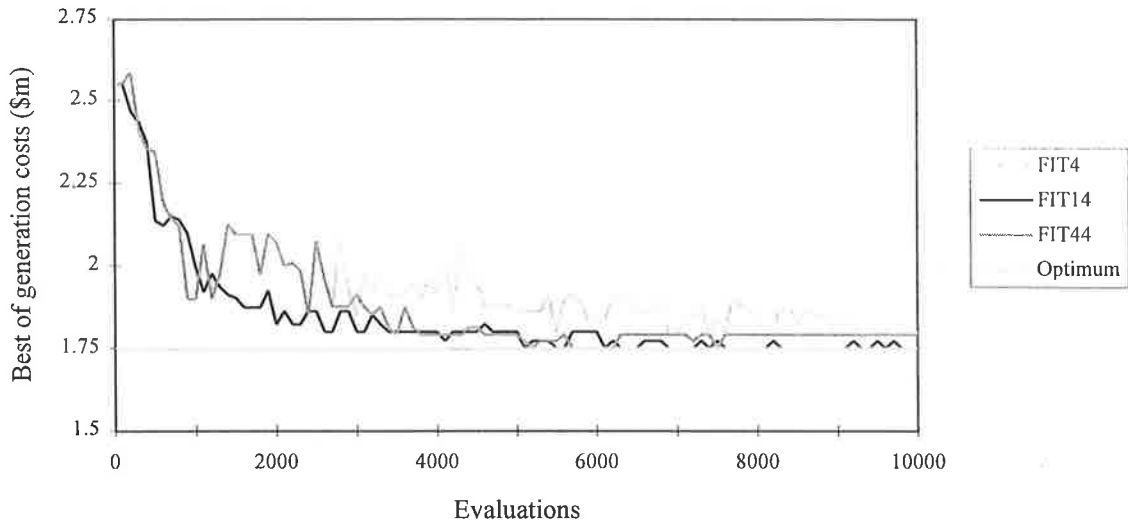


Figure 6.11 Best generation costs for GA runs FIT4 ($n=1$ throughout), FIT14 ($n=2$ throughout) and FIT44 (n is increasing as GA run proceeds) using the *inverse* fitness function

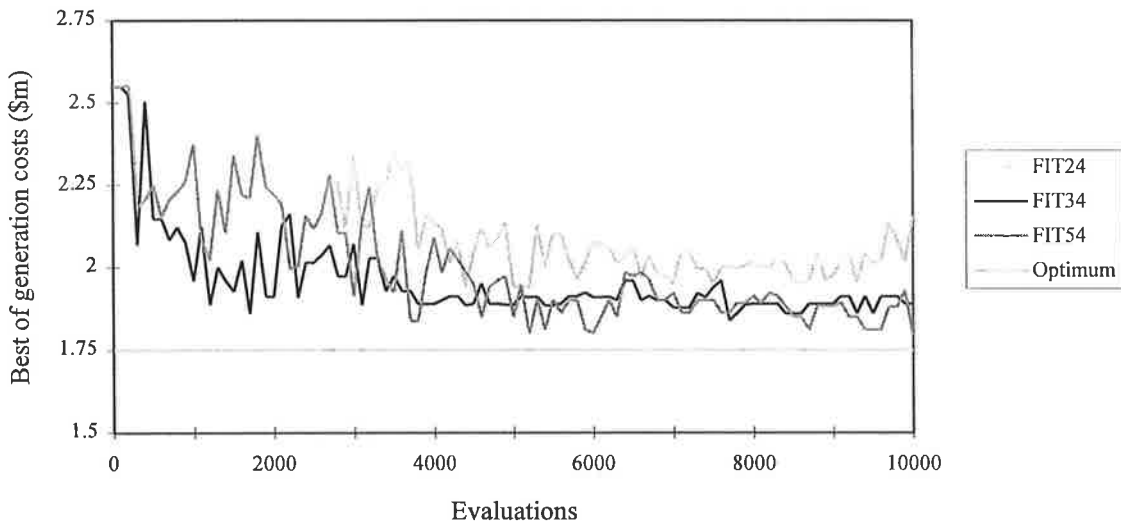


Figure 6.12 Best generation costs for GA runs FIT24 ($n=1$ throughout), FIT34 ($n=2$ throughout) and FIT54 (n is increasing as GA run proceeds) using the *linear* fitness function

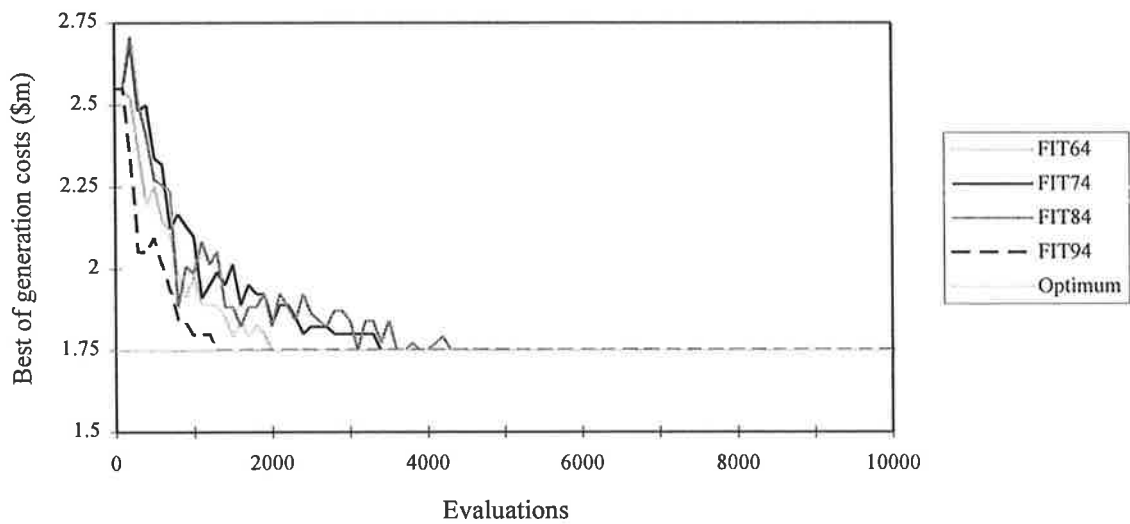


Figure 6.13 Best generation costs for GA runs FIT64 (binary tournament with $p_t=1.0$), FIT74 (binary tournament with $p_t=0.9$), FIT84 (binary tournament with $p_t=0.8$) and FIT94 (ternary tournament)

6 Improvements to the simple GA for pipe network optimisation

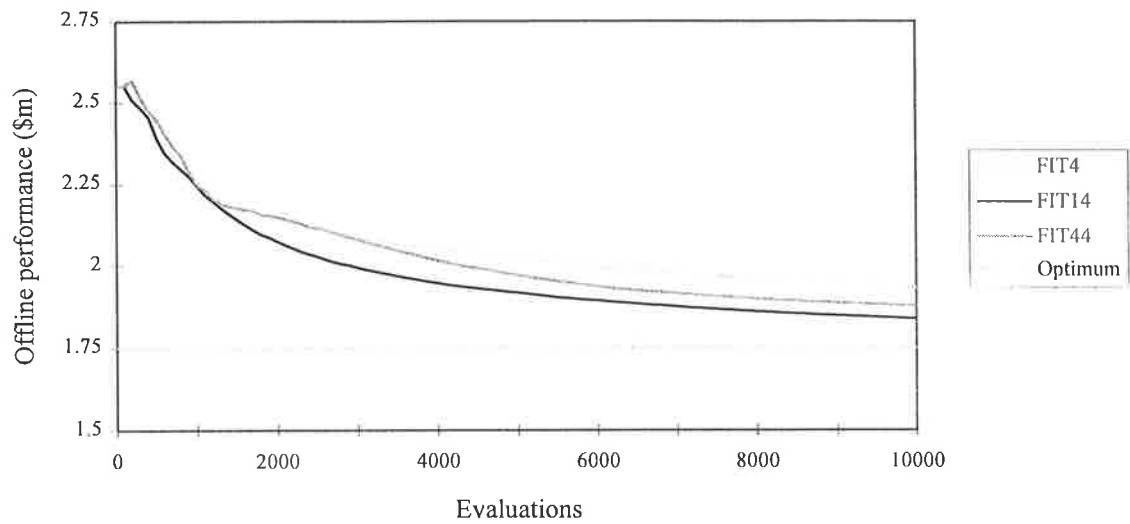


Figure 6.14 Offline performance (running average of best cost solutions) for GA runs FIT4 ($n=1$ throughout), FIT14 ($n=2$ throughout) and FIT44 (n is increasing as GA run proceeds) using the *inverse* fitness function

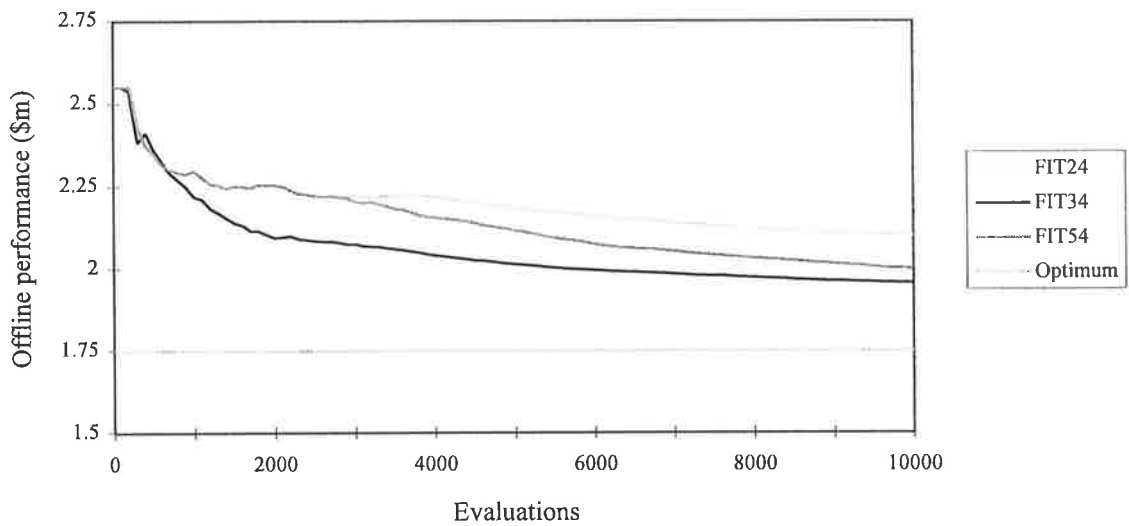


Figure 6.15 Offline performance (running average of best cost solutions) for GA runs FIT24 ($n=1$ throughout), FIT34 ($n=2$ throughout) and FIT54 (n is increasing as GA run proceeds) using the *linear* fitness function

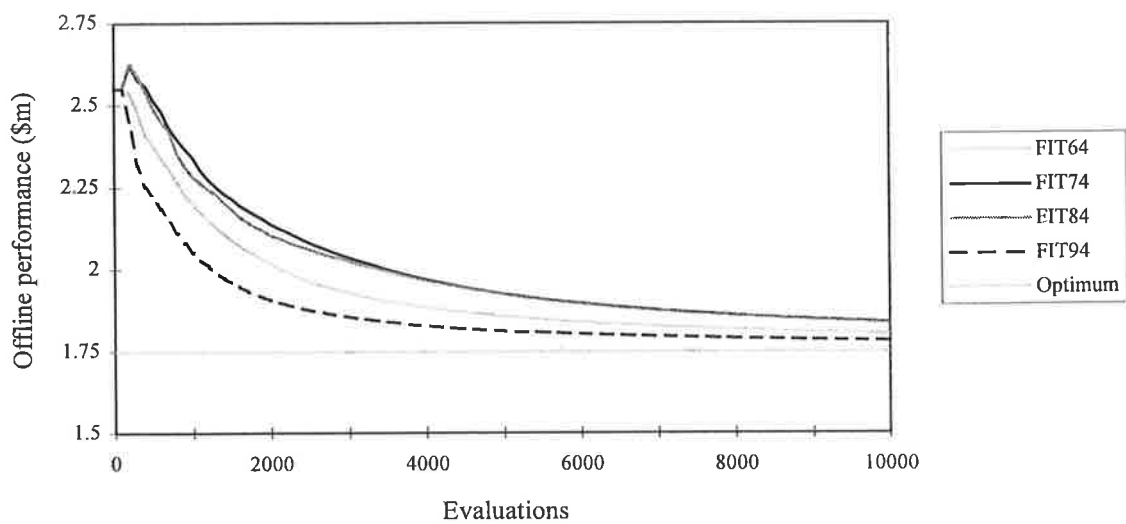


Figure 6.16 Offline performance for GA runs FIT64 (binary tournament with $p_t=1.0$), FIT74 (binary tournament with $p_t=0.9$), FIT84 (binary tournament with $p_t=0.8$) and FIT94 (ternary tournament)

6 Improvements to the simple GA for pipe network optimisation

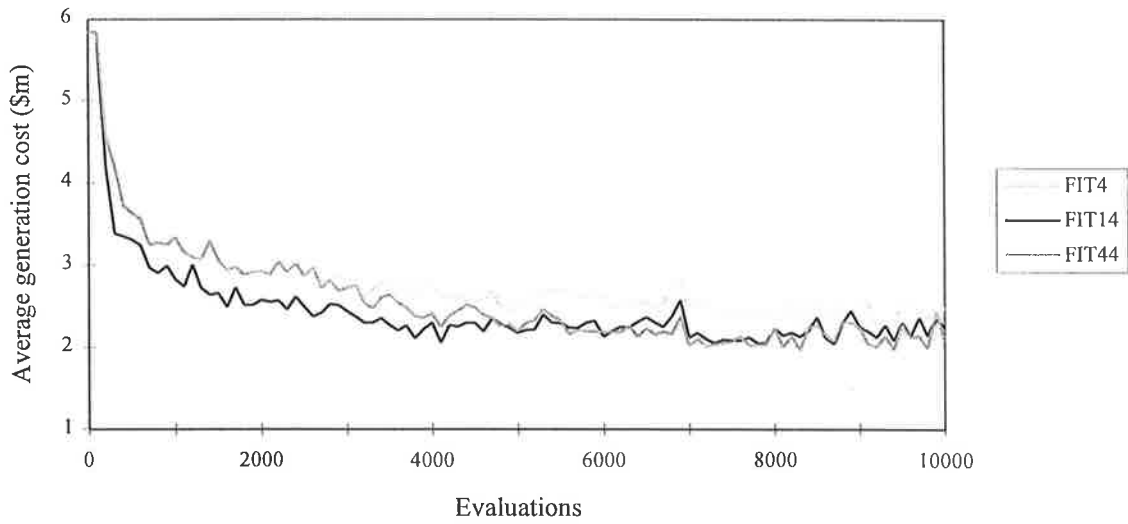


Figure 6.17 Average generation costs for GA runs FIT4 ($n=1$ throughout), FIT14 ($n=2$ throughout) and FIT44 (n is increasing as GA run proceeds) using the *inverse* fitness function

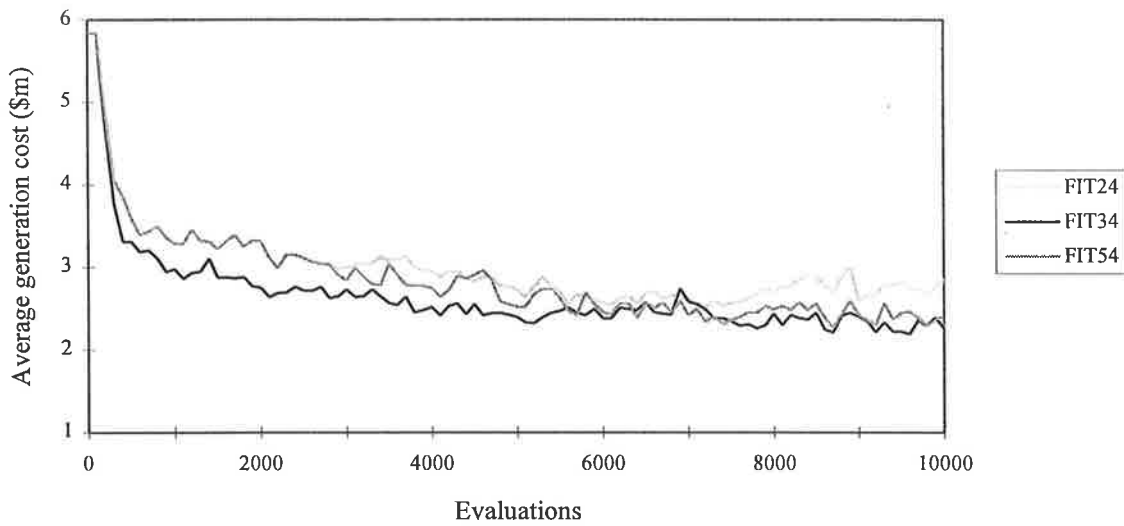


Figure 6.18 Average generation costs for GA runs FIT24 ($n=1$ throughout), FIT34 ($n=2$ throughout) and FIT54 (n is increasing as GA run proceeds) using the *linear* fitness function

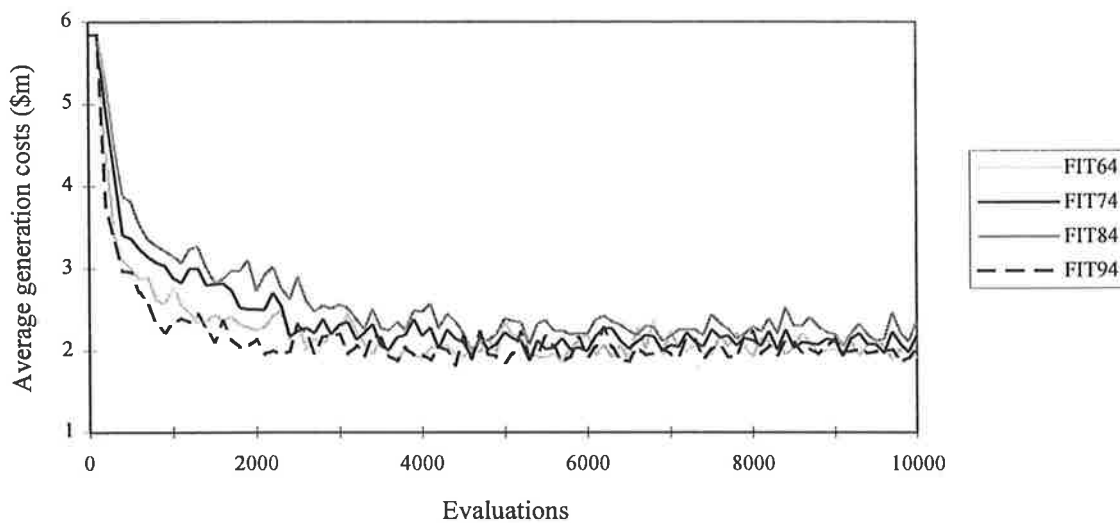


Figure 6.19 Average generation costs for GA runs FIT64 (binary tournament with $p_t=1.0$), FIT74 (binary tournament with $p_t=0.9$), FIT84 (binary tournament with $p_t=0.8$) and FIT94 (ternary tournament)

6 Improvements to the simple GA for pipe network optimisation

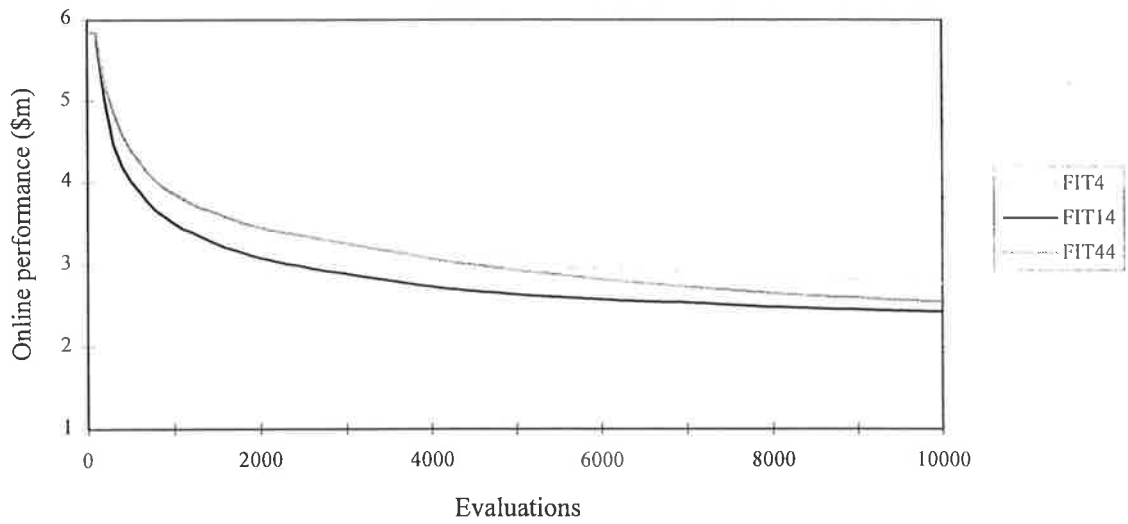


Figure 6.20 Online performance (running average of all solutions) for GA runs FIT4 ($n=1$ throughout), FIT14 ($n=2$ throughout) and FIT44 (n is increasing as GA run proceeds) using the *inverse* fitness function

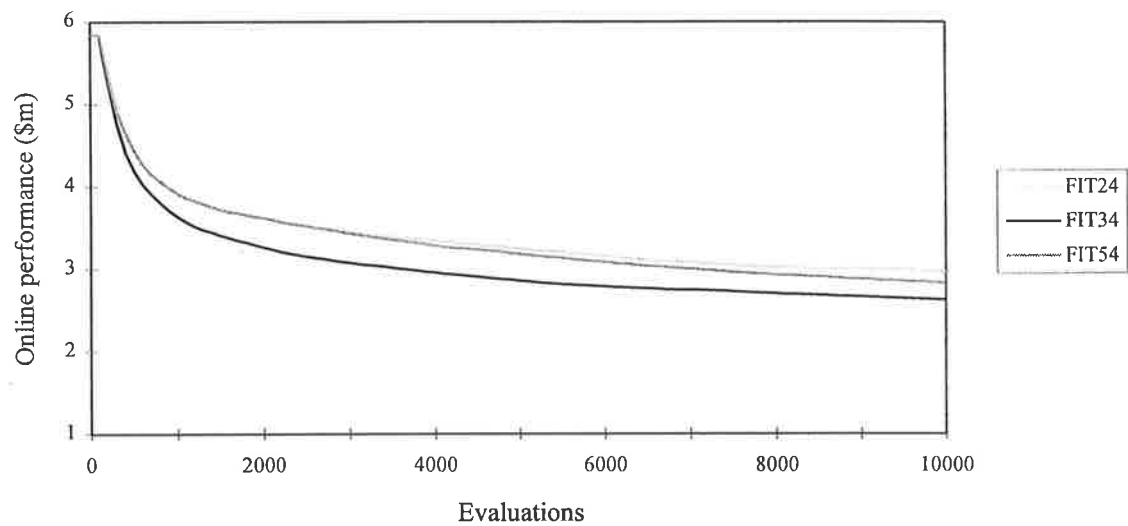


Figure 6.21 Online performance (running average of all solutions) for GA runs FIT24 ($n=1$ throughout), FIT34 ($n=2$ throughout) and FIT54 (n is increasing as GA run proceeds) using the *linear* fitness function

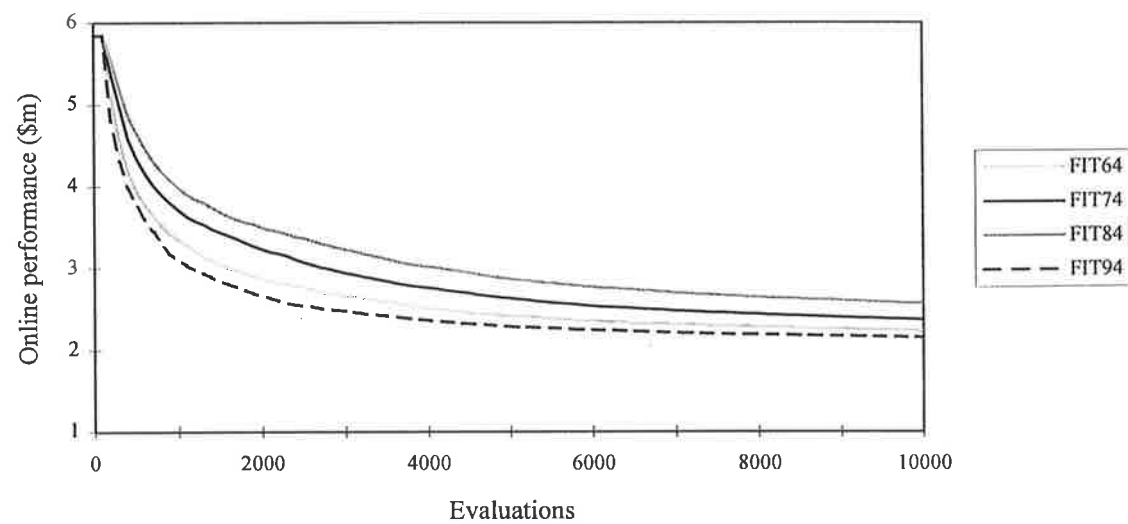


Figure 6.22 Online performance for GA runs FIT64 (binary tournament with $p_t=1.0$), FIT74 (binary tournament with $p_t=0.9$), FIT84 (binary tournament with $p_t=0.8$) and FIT94 (ternary tournament)

online performance (Figures 6.19 and 6.22) show rapid convergence in the first half of the GA run and little further improvement for the second half of the GA runs.

Average cost of all strings selected for mating

Figures 6.23-6.25 compare average generation cost with the average cost of the strings selected for mating in the new population, for each generation of the GA search for GA runs FIT4 (proportionate selection with fixed $n=1$ and inverse fitness function), FIT44 (proportionate selection with variable n and inverse fitness function) and FIT64 (binary tournament selection). The purpose of these comparisons is to observe the ability of the fitness functions to separate the good solutions from average solutions in the population. Proportionate selection with increasing values on the fitness scaling exponent n is better able to differentiate the good strings from average strings in more competitive populations than the fixed value of the exponent $n=1$.

6.4.6 Recommendations for selection schemes

The empirical analysis of two commonly used selection schemes of proportionate selection and tournament selection has shown proportionate selection to be very effective for an appropriate fitness function (the inverse fitness function in Eq. 6.4 is recommended for the pipe network optimisation application) and using power law fitness scaling. Binary tournament selection is effective and very efficient for suitable combinations of the standard GA parameters (large population sizes, and relatively high crossover and mutation rates are recommended) and there is some value in allowing the weaker individual a small chance to win the tournament. Ternary tournament selection can be used to determine reasonably good solutions very quickly. Early in the GA model run, care must be taken to avoid premature convergence of the population due to domination by a better than average (but not optimal) solution and late in the GA model run the parent selection scheme should maintain appropriate levels of competition between high numbers of highly fit solutions, so that the search is not left to drift aimlessly.

There is a possible explanation for the poor performance of the linear fitness function. Late in the GA run when the populations have effectively converged, the coded strings in the population look very similar and their associated fitness values are much closer. The linear fitness function compares the cost of a string to the maximum cost string in the population as a means of measuring fitness. Often, a very weak individual is formed in the new population by some particularly disruptive crossover or mutation. Since the very weak string is distant from the rest of the population in terms of cost, the distances between the fit individuals in the population in comparison to this distance will be small. This would make it difficult for the linear fitness function to differentiate between highly fit strings in the population. Overall, binary tournament selection with a probability of selecting the better individual of 0.8 gives the best results for this example.

6 Improvements to the simple GA for pipe network optimisation

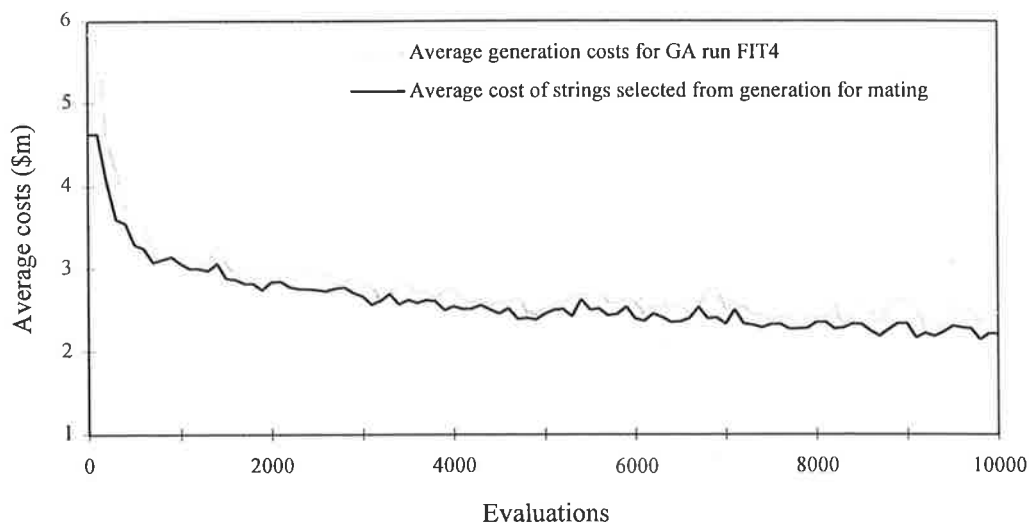


Figure 6.23 Average generation costs and average cost of strings selected from the generation for mating for GA run FIT4 ($n=1$ throughout using the *inverse* fitness function)

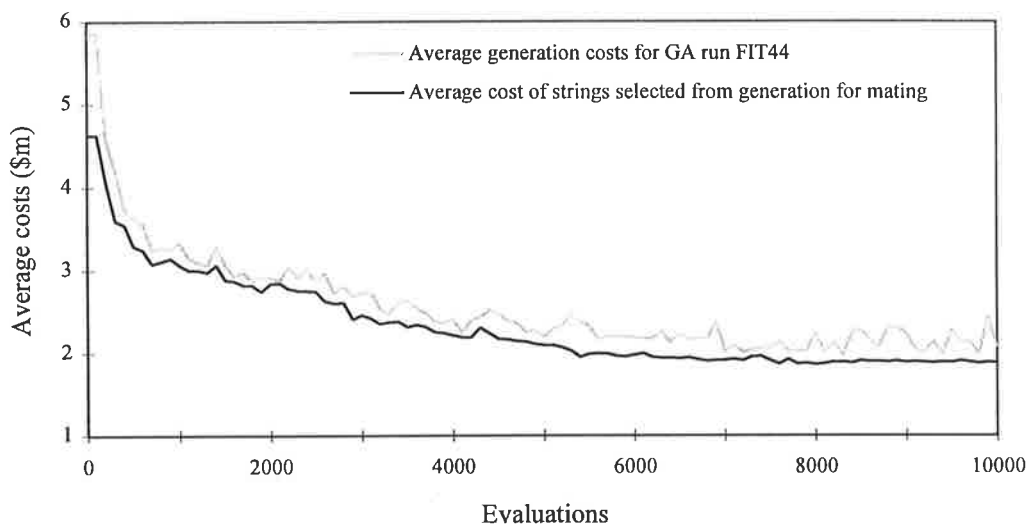


Figure 6.24 Average generation costs and average cost of strings selected from the generation for mating for GA run FIT44 (n is increasing as GA run proceeds using the *inverse* fitness function)

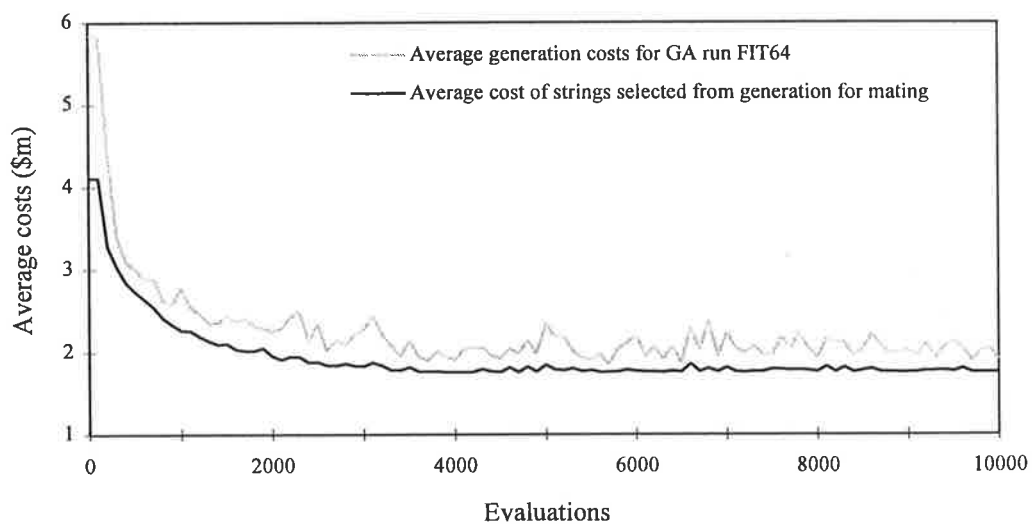


Figure 6.25 Average generation costs and average cost of strings selected from the generation for mating for GA run FIT64 (binary tournament with $p_r=1.0$)

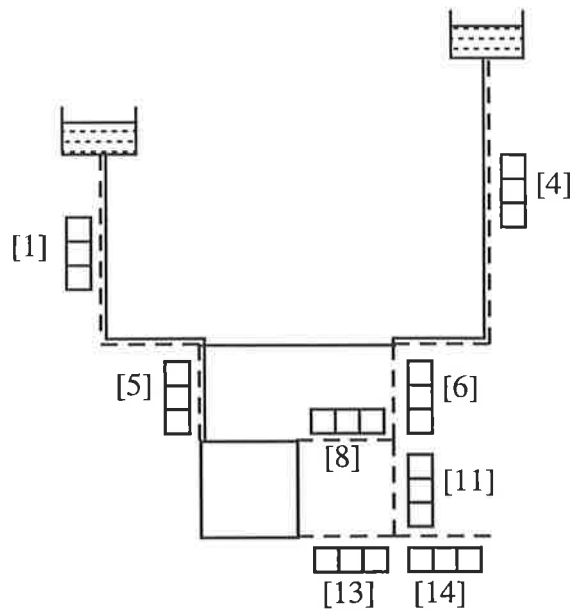
6.5 Coding Schemes

In the genetic algorithm search, the set of design parameters describing trial solutions are represented by some coded structure such as a string or array of symbols. Davidson and Goulter (1992a, 1992b) used arrays of 1's and 0's to represent solutions to a pipe network layout problem. In most cases, trial solutions are most efficiently represented by a unique fixed-length string of symbols such as coded strings of 1's and 0's. The coded string simulates the structure of a chromosome of genetic code in the artificial evolution of the GA search.

Coded strings are formed to represent solutions to the Gessler problem in Figure 6.26. The coded strings are composed of eight decision-variable substrings of symbols representing the eight decision-variables of the optimisation. The decision-variables are each allocated substring positions in the coded string. For example, the first substring position is associated with the existing pipe [1]. The symbol(s) at the substring positions in a coded string decode to the pipe network design parameters by observing some specified mapping between the decision-variable substring symbols and the design parameters such as new pipe sizes. In this way, the artificial genetic code describes a pipe network design solution.

Traditionally, decision-variable substrings of binary codes are used to specify the mapping between substring code and design parameters. In the empirical analysis that follows, three alternative mappings between decision-variable substrings and design parameters are considered. The alternative representation mappings considered include substrings of the traditional binary codes, substrings of Gray codes (both based on the binary alphabet) and an integer coding scheme. The alternative mappings between substring symbols and design parameters are given in Table 6.23. The design parameters in Table 6.23 are the available diameters for new pipes and the options for existing pipes (cleaning, duplication or 'do nothing') for the Gessler network expansions.

As an illustration, the optimal network solutions 1 and 2 (Table 5.4) for the Gessler problem determined by the exhaustive enumeration are shown as coded strings of substrings of binary codes, substrings of Gray codes and substrings of integer numbers (*octal* code) in Figure 6.26.



Two-reservoir Gessler network

[1]	[4]	[5]	[6]	[8]	[11]	[13]	[14]	Construction of the coded string
□□□□	□□□□	□□□□	□□□□	□□□□	□□□□	□□□□	□□□□	
leave	d14"	leave	12"	8"	8"	6"	10"	Optimum solution 1 (Table 5.4)
000	110	000	011	001	001	000	010	Solution 1 in binary codes
000	101	000	010	001	001	000	011	Solution 1 in Gray codes
1	7	1	4	2	2	1	3	Solution 1 in integer codes
leave	d14"	leave	12"	8"	10"	6"	8"	Optimum solution 2 (Table 5.4)
000	110	000	011	001	010	000	001	Solution 2 in binary codes
000	101	000	010	001	011	000	001	Solution 2 in Gray codes
1	7	1	4	2	3	1	2	Solution 2 in integer codes

Figure 6.26 The formation of the coded string for the Gessler problem

Table 6.23 Representation mappings

Pipe network design parameters		Decision-variable substrings encodings		
If existing pipe [1]-[4]-[5]	If new pipe [6]-[8]-[11]-[13]-[14]	Binary codes	Gray codes	Integer codes
do nothing	new 6" pipe	000	000	1
duplicate with 6"	new 8" pipe	001	001	2
clean existing pipe	new 10" pipe	010	011	3
duplicate with 8"	new 12" pipe	011	010	4
duplicate with 10"	new 14" pipe	100	110	5
duplicate with 12"	new 16" pipe	101	111	6
duplicate with 14"	new 18" pipe	110	101	7
duplicate with 16"	new 20" pipe	111	100	8

There are eight decision-variable choices for the existing pipes in the Gessler network (pipes [1], [4] and [5]) including cleaning, duplication with new parallel pipes and the 'do nothing' option. The order of the list of allowable decision-variable choices for existing pipes in Table 6.23 was chosen as it best approximates increasing cost, performance and equivalent diameter for the decisions as demonstrated in Tables 6.24 and 6.25 for existing pipes [1] and [4] respectively. The performance is measured by a frictional head loss (ft/1000ft) assuming an arbitrary constant flow in the pipe of 1000gpm (=2.2278cfs). The equivalent diameters (in) are for an assumed Hazen-Williams roughness coefficient of the equivalent pipe of $C=120$.

6.5.1 Binary codes and Gray codes

In accordance with fundamental GA theory, it was decided to test coding schemes based on the minimum {0-1} binary alphabet including binary codes and Gray codes. Small (low-cardinality) alphabets such as the binary alphabet generate longer coded strings and hence maximise the number of string similarities or schemata present in a population of coded strings (Goldberg, 1990). The preservation of greater numbers of schemata or string similarities is desirable so that there is a greater depth of genetic information available to be interpreted and processed in the GA search (Holland, 1975; Goldberg, 1989).

Table 6.24 Ranked design parameters for the upgrade of existing pipe [1]

PIPE [1]			
Existing properties $L=15840\text{ft}$, $C=75$, $D=14\text{in}$			
Design parameters	Unit cost (\$/ft)	Performance (ft/1000ft)	Equivalent diameter (in)
do nothing	0	3.31	11.709
duplicate with 6"	15.1	2.468	12.439
clean existing pipe	18.5	1.387	14.0
duplicate with 8"	19.3	1.856	13.188
duplicate with 10"	28.9	1.295	14.199
duplicate with 12"	40.5	0.864	15.431
duplicate with 14"	52.1	0.565	16.838
duplicate with 16"	59.4	0.369	18.379

Table 6.25 Ranked design parameters for the upgrade of existing pipe [4]

PIPE [4]			
Existing properties $L=21120\text{ft}$, $C=80$, $D=10\text{in}$			
Design parameters	Unit cost (\$/ft)	Performance (ft/1000ft)	Equivalent diameter (in)
do nothing	0	15.137	8.571
duplicate with 6"	15.1	8.212	9.718
clean existing pipe	16.8	7.144	10.0
duplicate with 8"	19.3	4.922	10.795
duplicate with 10"	28.9	2.774	12.144
duplicate with 12"	40.5	1.550	13.685
duplicate with 14"	52.1	0.884	15.356
duplicate with 16"	59.4	0.522	17.114

The Gray code representation is such that adjacent decision-variable substrings are separated by a Hamming distance of 1 (differ by one bit). For example, only one bit changes between neighbouring substrings of 000 and 001 and 011. By comparison, adjacent substrings of binary codes may differ by more than one bit. For example, the substring 100 which follows 011 in the list of binary-coded substrings (Table 6.23) differs at all three bit positions. This extreme Hamming distance of 3 is referred to as a Hamming cliff.

Using substrings of Gray codes, similar code represents adjacent design variable choices, and thus similar network designs are represented by strings of code with a close resemblance and the coded string solutions are closer in the GA search space.

Hollstien (1971) concluded Gray codes may be superior to binary codes, since adjacent codes differ by only one bit and bit-wise complement mutations cause less disruption to the solution. Based on experimental results, Bethke (1981) found Gray codes improved the performance of the GA and suggested the reason for this is that a Gray code maps Euclidean neighbourhoods into Hamming neighbourhoods.

Caruana and Schaffer (1988) compared the performance of binary codes and a Gray code representation and found the Gray code to be better than or equivalent to binary coding for six functions tested including the five functions considered by DeJong (1975) to test the GAs performance. Caruana and Schaffer concluded that by eliminating the Hamming Cliff of binary coding, the Gray codes might improve the performance of the GA.

The GA process is not concerned with the method of evaluation of the coded string. The GA is blind to the mapping that occurs between a coded string and the set of design parameters the string describes. Caruana and Schaffer suggested the GA may be misled by biases introduced by the mapping such as a the Hamming cliff of binary codes.

The GA is concerned only with the production of new populations of superior strings by subjecting the old population of strings to a series of GA operators. The GA achieves this by interpreting the information in the special relationships between string similarities and the string fitness. The GA may expect strings that appear to be constructed of similar genetic code to possess a similar fitness (that is, strings that have similar 0-1 structures will describe similar trial network designs).

6.5.2 Integer codes

Since the design parameters for the pipe network optimisation problems are often sets of discrete decisions (such as new pipe sizes), an integer coding scheme is considered in which integer numbers map directly to the design parameters (Table 6.23).

Goldberg (1990) outlined a number of reasons why a large (high-cardinality) alphabet and shorter coded strings may be preferred in GAs. The combination of large alphabets and a creeping mutation operator for hill-climbing overcomes the problem of Hamming Cliffs, experienced in binary-coded GAs using bit-wise complement mutations. Gray codes are considered in this study to overcome problems such as Hamming Cliffs, however Goldberg

(1990) suggests the use of Gray codes may introduce higher order non-linearities with respect to recombination. Instead, he suggests the use of both bit-wise and decision-variable-wise mutations in binary-coded GAs. Decision-variable-wise mutations are considered later in this chapter (Section 6.7). High-cardinality alphabets may be preferred to low-cardinality alphabets because they are known to converge faster (under certain operating conditions), however as the alphabet cardinality increases, the quality of the solution obtained decreases.

The use of larger alphabets reduces the dimensionality of the solution space and may reduce the size of the solution space. The number of allowable design parameters should be some power of 2 to be efficiently represented by substrings of binary codes. Fortunately, for the original Gessler problem there are 8 discrete design options for the three existing pipes (including cleaning, 6 pipe sizes for duplicate pipes and the 'do nothing' option). The 8 discrete options are associated with 1 of the 8 unique 3-bit binary coded substrings as shown in Table 6.23. However, only 6 new pipe sizes were originally available for the five new pipes and the 18" and 20" pipe sizes were introduced to 'make up the numbers'. This action increased the size of the solution space to be searched by the GA from about 4 million solutions to 16.8 million solutions. There may be more effective ways to utilise the two redundant binary numbers.

Beasley et al. (1993b) mentioned some possible solutions to this problem of redundant codes. A chromosome containing code which does not map to a valid gene value may be discarded as illegal or assigned a low fitness. These solutions are not recommended as good gene values elsewhere in the chromosome may be discarded. The preferred solution is to map the invalid code to a valid gene value. Beasley et al. (1993b) suggested this could be achieved by fixed remapping (map the redundant code to a specific valid gene value), random remapping (map the redundant code to a random valid gene value) or by probabilistic remapping (every gene value is remapped to one of two valid values in a probabilistic way).

The integer coding scheme does not need to introduce new pipe sizes for the redundant binary codes, but instead could limit the integer numbers to the interval 1 to 6 in the string positions corresponding to new pipes. To allow for a fair comparison between the integer coded GAs and the binary-coded GAs in the following experimental analysis, the integer coded GAs were applied to the extended solution space of 16.8 million solutions.

Davis and Coombs (1987) studied an application of GAs to the design of packet switching communication networks. They used coded strings that were lists of link speeds. The link speeds in the coded string corresponded to links in the backbone communication network, in much the same way as the pipe sizes represented in our coded strings correspond to pipes in the pipe network. The link speeds were chosen from a list of allowable link speeds for the design such that each link speed in the coded string was coded as a single 'letter'. The motivation for

their chosen coding scheme was to admit the introduction of an advanced GA operator called ‘creep’ (discussed in Section 6.7). In the pipe network optimisation problem, the pipe sizes are chosen from a discrete list of allowable pipe sizes and the integer coding scheme considered here effectively represents the pipe sizes in the coded string as a single ‘letter’. Davis and Coombs believed the use of the high-cardinality alphabet was not detrimental to the GA search in this particular solution space, in which the best link speeds were restricted to one or two regions of the link size list. If the best link speeds were found in periodic regions of the list of link sizes, the use of smaller alphabets and longer coded strings would have been appropriate.

In fact, GAs that use high-cardinality codings and floating-point codings (real-coded GAs) have enjoyed success in a number of practical applications. Goldberg (1990) reviews the history of real-coded GAs before presenting a theory for real-coded GAs consistent with the fundamental GA Schema Theory of Holland (1975). Goldberg accounts for the convergence of GAs with high-cardinality alphabets by introducing the theory of ‘virtual characters’ and ‘virtual alphabets’. The power of the selection operator is shown to reduce high-cardinality alphabets to low-cardinality virtual alphabets quite early in the GA search, and the GA continues the search over the smaller alphabet. The theory explains the empirical success of real-coded GAs, however it demonstrates that real-coded GAs may be ‘blocked’ from accessing the global optimum solution (become stuck on local optima) for some problems. To overcome the limitations of real-coded GAs in such cases, it may be necessary to introduce new genetic mechanisms or some other modifications to the real-coded GA.

6.5.3 The optimum arrangement of decision-variable substring positions within the coded string

The pipes in the system are associated with decision-variable substring positions in the coded string structure. Some consideration should be given to the arrangement of the decision-variable substrings within the string. The GA theory suggests the decision-variable substrings which are likely to develop close relationships in a solution should be positioned nearby in the string to assist the GA in identifying *building blocks* or short, low-order schemata.

The arrangement of decision-variable substrings in the coded string built for the Gessler pipe network optimisation problem was chosen as shown in Figure 6.26. The pipes in the two main supply lines (i.e. pipes [1] and [5] and pipes [4] and [6]) are all situated at the beginning of the string. The substring positions represent pipes further downstream from the sources as the coded string structure is traversed.

These decisions for the upstream pipes are most influential in terms of the economics and hydraulic feasibility of the proposed pipe network design. By closely grouping these decision-

variable substrings we hope to create one short, powerful building block for the GA search. An alternative substring arrangement is tested in the following GA runs that arranges the decision-variable substrings in the coded string structure according to Figure 6.27. In this way, the influential decisions regarding the main supply line (pipes [1] and [5]) are located at the beginning of the string and the influential decisions regarding the alternative main supply line (pipes [4] and [6]) are located at the opposite end of the string. The less influential downstream pipes make-up the intermediate segment of the coded string structure. This arrangement of decision-variable substring positions maximises the opportunities to trial, through crossover, different combinations of these alternative main supply lines.

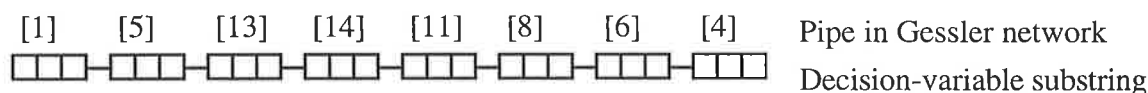


Figure 6.27 An alternative arrangement of the decision substrings in the string

6.5.4 Genetic algorithm runs to compare coding schemes

A series of GA model runs designated CODE1-CODE35 are carried out to compare the performance of the various coding schemes (Table 6.2). The GA runs CODE1-CODE5 are the traditional GA runs using coded strings formed with substrings of 3-bit binary codes mapped to the design parameters (according to Table 6.23). The GA runs CODE11-CODE15 use a string made up of substrings of Gray codes. The set of GA runs CODE21-CODE25 use a string of integer numbers to represent network solutions. Finally, GA runs CODE31-CODE35 use substrings of binary codes, but the decision-variable substring positions in the string have been shuffled (as shown in Figure 6.27). The results of the GA model runs (including lowest cost solutions and lowest average generation costs achieved, ultimate offline and online performance) are presented in Tables 6.26-6.29.

Lowest cost solutions

The binary codes (GA runs CODE1-CODE5) find the optimal network solution most often. The Gray codes (GA runs CODE11-CODE15) and binary codes with the shuffled substring positions (GA runs CODE31-CODE35) are both capable of finding good low cost solutions. The integer codes fail to determine the optimum solution in any of the five GA runs CODE21-CODE25 (Table 6.28).

Table 6.26 Search results for genetic algorithm model runs CODE1-CODE5

Substrings of binary codes. GA runs CODE1-CODE5 equivalent to GA runs PEN11-PEN15.					
GA RUNS	CODE1	CODE2	CODE3	CODE4	CODE5
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m)	1.7503 [‡]	1.7503 [‡]	1.7503 [‡]	1.7725	1.8807
(after - generations)	(59)	(96)	(59)	(93)	(82)
Lowest cost GA design generated (Table 5.4)	2	2	1	3	-
Lowest average generation cost (\$m)	2.045	2.139	2.038	2.271	2.274
(after - generations)	(212)	(73)	(176)	(92)	(86)
Ultimate offline performance (\$m)	1.963	1.896	1.916	1.935	2.026
Ultimate online performance (\$m)	2.560	2.557	2.475	2.766	2.694

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.27 Search results for genetic algorithm model runs CODE11-CODE15

Concatenated decision-variable substrings of Gray codes					
GA RUNS	CODE11	CODE12	CODE13	CODE14	CODE15
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m)	1.7999	1.8115	1.7503 [‡]	1.7503 [‡]	1.8390
(after - generations)	(219)	(45)	(95)	(75)	(95)
Lowest cost GA design generated (Table 5.4)	6	10	1	2	28
Lowest average generation cost (\$m)	2.025	2.140	1.937	2.323	2.080
(after - generations)	(253)	(109)	(189)	(51)	(98)
Ultimate offline performance (\$m)	1.974	1.919	1.842	1.926	1.966
Ultimate online performance (\$m)	2.473	2.495	2.311	2.733	2.489

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.28 Search results for genetic algorithm model runs CODE21-CODE25

Coded strings of integers					
GA RUNS	CODE21	CODE22	CODE23	CODE24	CODE25
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	267	134	197	100	100
Lowest solution cost (\$m)	1.8510 [†]	1.8337	1.7910	1.9119	1.8612
(after - generations)	(184)	(55)	(130)	(89)	(74)
Lowest cost GA design generated (Table 5.4)	-	23	5	-	-
Lowest average generation cost (\$m)	2.105	2.026	1.863	2.078	1.955
(after - generations)	(171)	(113)	(195)	(100)	(96)
Ultimate offline performance (\$m)	2.099	1.930	1.863	2.097	1.957
Ultimate online performance (\$m)	2.379	2.357	2.171	2.641	2.342

[†] Infeasible design. The solution cost includes the penalty cost. GA run CODE21 determined a best cost feasible design for \$1.9536 million after 86 generations.

Table 6.29 Search results for genetic algorithm model runs CODE31-CODE35

Substrings of binary codes with an alternative arrangement of decision-variable substrings within the string.					
GA RUNS	CODE31	CODE32	CODE33	CODE34	CODE35
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m)	1.7725	1.8300	1.7503 [‡]	1.7503 [‡]	1.8285
(after - generations)	(255)	(60)	(71)	(44)	(91)
Lowest cost GA design generated (Table 5.4)	3	22	2	1	20
Lowest average generation cost (\$m)	2.044	2.394	1.946	2.374	2.289
(after - generations)	(266)	(100)	(143)	(49)	(98)
Ultimate offline performance (\$m)	2.119	2.044	1.811	1.855	2.010
Ultimate online performance (\$m)	2.715	2.799	2.354	2.774	2.756

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Best generation costs and offline performance

The comparison of the plots of best of generation costs and offline performance (running average of best of generation costs) shown in Figures 6.28-6.29 demonstrate the inferior performance of integer codes (at least for this set of GA parameters). The binary codes with the alternative arrangement of substring positions is clearly the most effective in this case. The binary codes in GA run CODE4 and Gray codes in GA run CODE14 converge at the same slower rate, but are eventually successful in determining the best regions of the solution space. The Gray codes are shown to achieve good ultimate offline performance a little more consistently than binary codes across the set of five GA runs.

Average generation costs and online performance

There is little to separate the behaviour of the four alternative coding schemes in terms of the plots of average generation costs and online performance in Figures 6.30-6.31. The plot of average generation costs for the GA run CODE24 using the integer coding scheme is less variable compared to the plots of average generation costs for the coding schemes using the binary alphabet. This is likely due to the limited power of exploration of crossover applied to the relatively short strings of integer numbers.

In terms of ultimate online performance (average of all solution costs) and lowest average generation costs achieved, the Gray codes are the most effective of the coding schemes using the minimum binary alphabet and the binary codes adopting the original coded string structure are generally more effective than the binary codes adopting the alternative arrangement of decision-variable substring positions in Figure 6.27.

Further discussion

There is not enough evidence to suggest which of the alternative coding schemes using the low-cardinality binary alphabet is superior. Each of the binary coding schemes were capable of identifying low cost solutions. In terms of offline performance and online performance and lowest average generation costs achieved, the Gray codes hold a slender advantage and are certainly worthy of further consideration. There appears to be little advantage in one or the other arrangements of decision-variable substrings in the coded string solutions to the relatively small Gessler pipe network optimisation problem. The performance of the GA runs using strings of integer numbers is clearly inferior to the coding schemes based on the binary alphabet. The higher cardinality alphabet may be better suited to larger problems requiring longer coded strings and the use of a creeping mutation operator is recommended (Section 6.7).

6 Improvements to the simple GA for pipe network optimisation

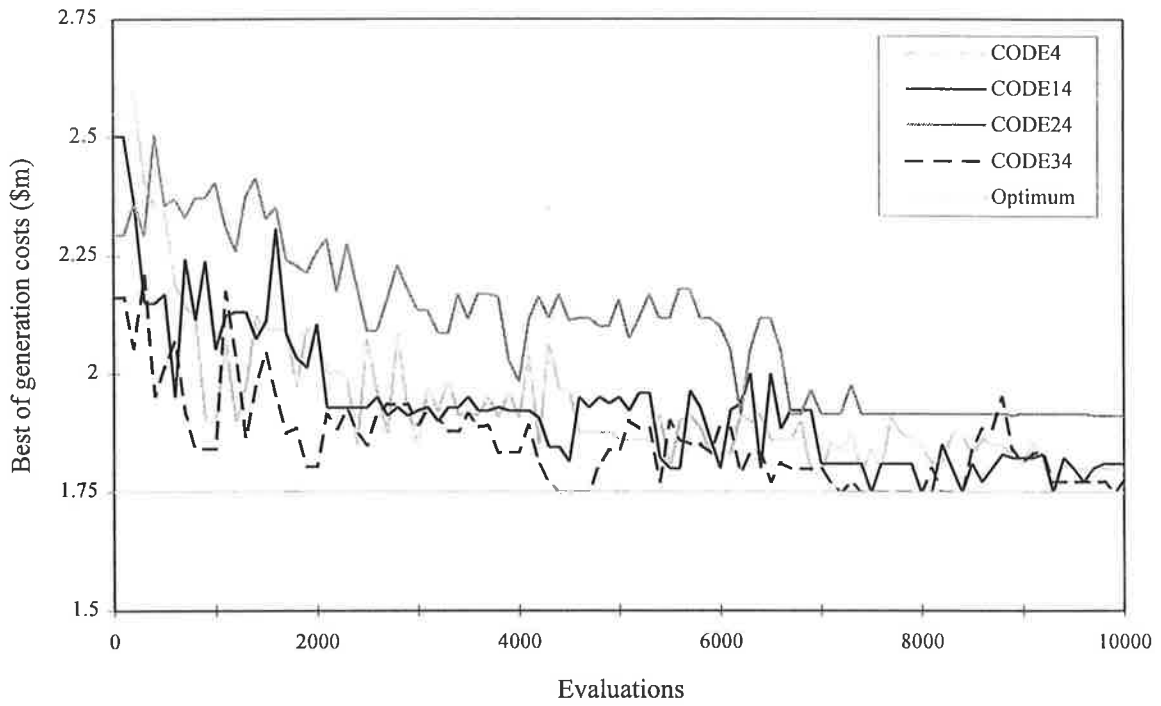


Figure 6.28 Best generation costs for GA runs CODE4 (substrings of binary codes), CODE14 (Gray codes), CODE24 (strings of integers) and CODE34 (binary codes with an alternative arrangement of substring positions)

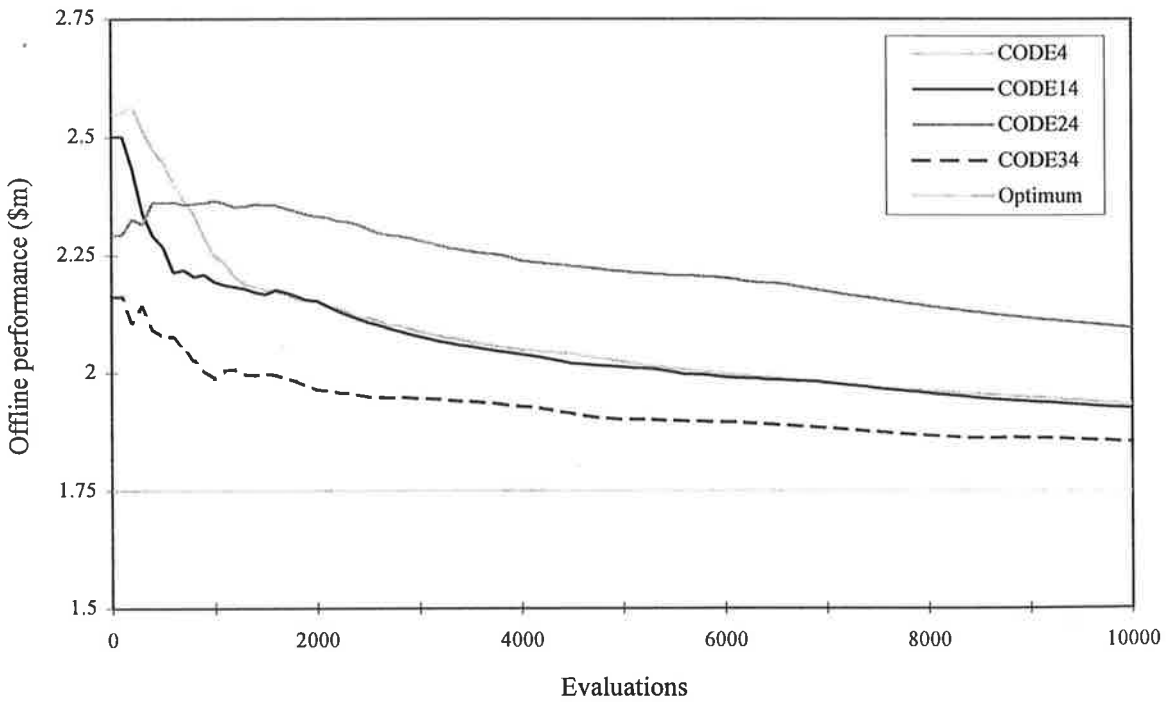


Figure 6.29 Offline performance for GA runs CODE4 (substrings of binary codes), CODE14 (Gray codes), CODE24 (strings of integers) and CODE34 (binary codes with an alternative arrangement of substring positions)

6 Improvements to the simple GA for pipe network optimisation

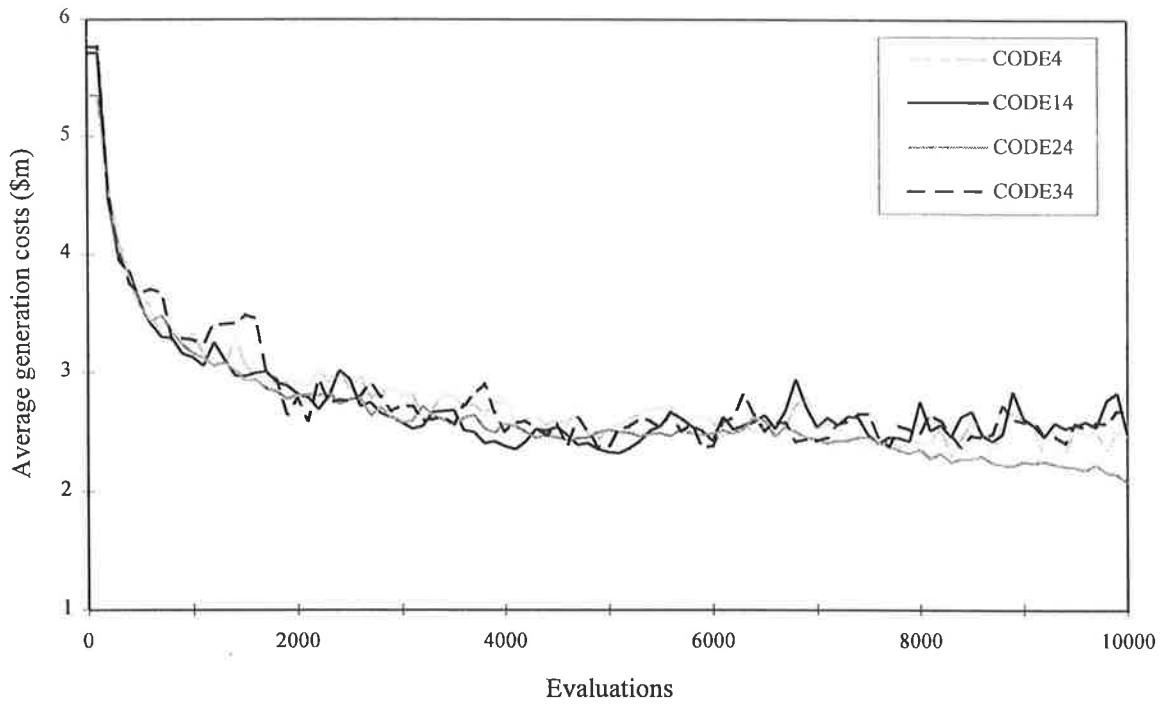


Figure 6.30 Average generation costs for GA runs CODE4 (substrings of binary codes), CODE14 (Gray codes), CODE24 (strings of integers) and CODE34 (binary codes with an alternative arrangement of substring positions)

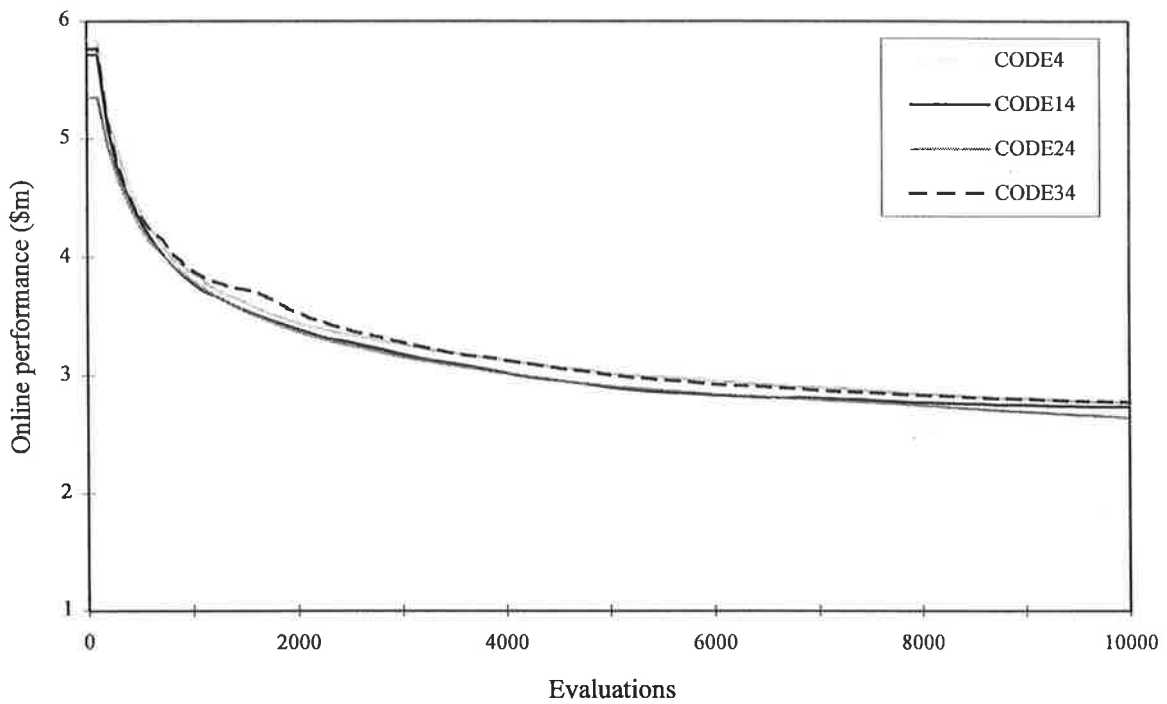


Figure 6.31 Online performance for GA runs CODE4 (substrings of binary codes), CODE14 (Gray codes), CODE24 (strings of integers) and CODE34 (binary codes with an alternative arrangement of substring positions)

6.5.5 Counting the numbers of decision-variable substrings

The number of copies of decision-variable substrings in a given substring position for each generation can be counted as the population evolves. The analysis shows which pieces of code dominate the population of coded strings as it advances and shows the shifting position of the population in the solution space. Every fifth population of 100 coded strings is considered in turn, through the 100 generations of GA runs CODE4 (traditional GA using binary codes) and CODE14 (substrings of Gray codes). The number of copies of the eight classes of unique 3-bit substring codes (000, 001, . . . , 111) at each substring position are recorded. The varying numbers of copies of substring code at each substring position are summarised for GA runs CODE4 and CODE14 in Figures 6.32-6.47. The trends may indicate how the chosen coding scheme influences the evolution of the string population. The decision-variable substrings are considered to be significant *building blocks* (Goldberg, 1989) in the GA search. The charts can be used to observe the existence of other shorter or longer building blocks and building blocks of different order.

The optimal combination of substring codes are known for the Gessler problem. The substring codes that constitute the optimal network solutions are shaded in Figures 6.32-6.47. The GA model run CODE4 (substrings of binary codes) determined a near-optimal solution for \$1.7725m (Solution 3 in Table 5.4) after 93 generations. The string of binary code representing Solution 3 is shown below. The GA run CODE14 (substrings of Gray codes) determined one of the global optimal solutions for \$1.7503m (Solution 2 in Figure 6.26) after 75 generations. The string of Gray codes representing Solution 2 is shown.

Substring position:	1	2	3	4	5	6	7	8
Corresponding pipe in Gessler network:	[1]	[4]	[5]	[6]	[8]	[11]	[13]	[14]
Solution 3 using binary codes (GA run CODE4):	000	110	000	011	001	001	001	010
Solution 2 using Gray codes (GA run CODE14):	000	101	000	010	001	011	000	001

Since each bit in the starting population of coded strings is randomly generated, the number of substring codes are randomly (and approximately evenly) distributed among the eight possible substring codes in any substring position in generation 1. As the population develops, the distribution of substring codes in a substring position usually congregate about the substring codes associated with high fitness, and in time one highly fit substring code may dominate the population at the substring position.

The first decision-variable substring position in the coded string is associated with the existing pipe [1] in the Gessler network. Figure 6.32 counts the number of copies of the 3-bit substrings of binary codes at the first substring position for GA run CODE4 and Figure 6.33

6 Improvements to the simple GA for pipe network optimisation

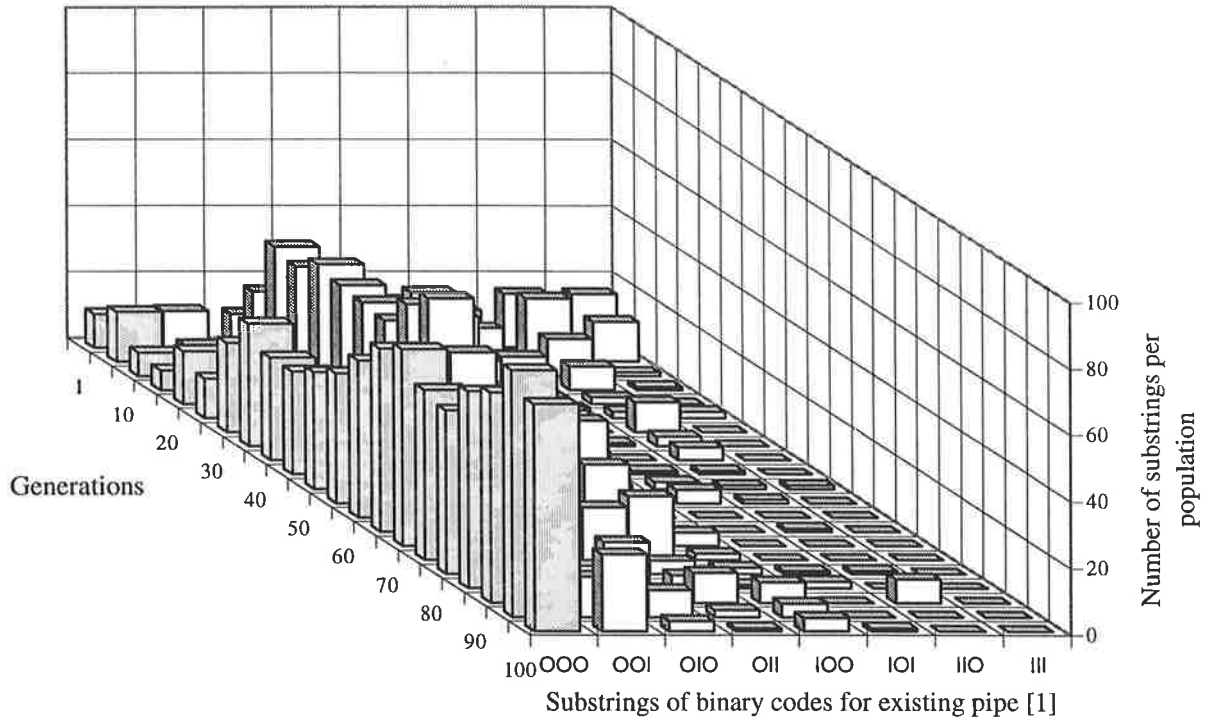


Figure 6.32 The variations with time of numbers of **decision-variable substrings of binary codes** at the first substring position (corresponding to existing pipe [1]) for GA run CODE4 (optimum substring is **000**)

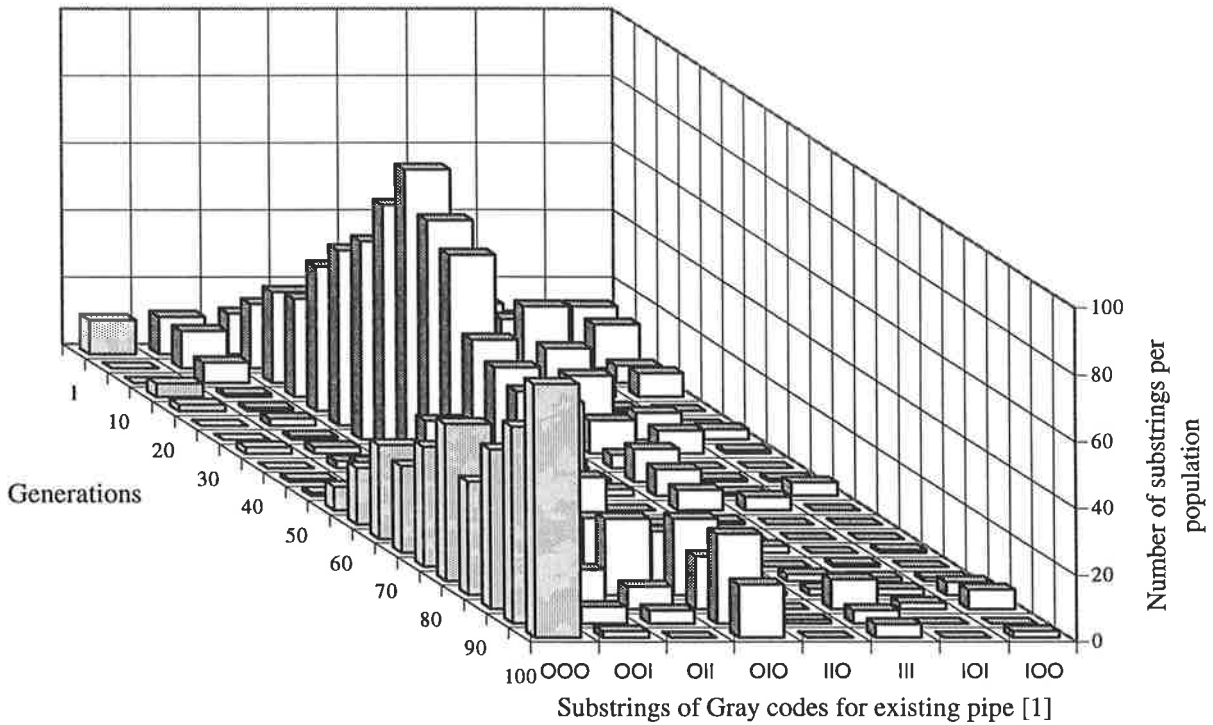


Figure 6.33 The variations with time of numbers of **decision-variable substrings of Gray codes** at the first substring position (corresponding to existing pipe [1]) for GA run CODE14 (optimum substring is **000**)

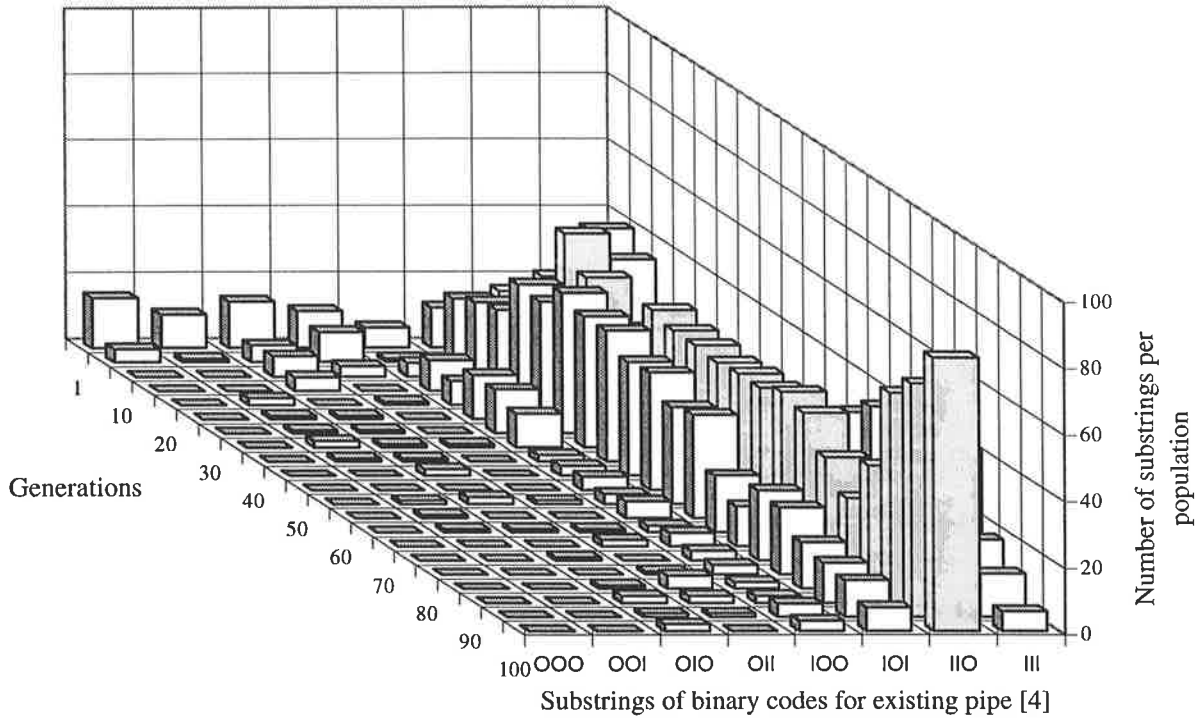


Figure 6.34 The variations with time of numbers of **decision-variable substrings of binary codes** at the second substring position (corresponding to existing pipe [4]) for GA run CODE4 (optimum substring is **110**)

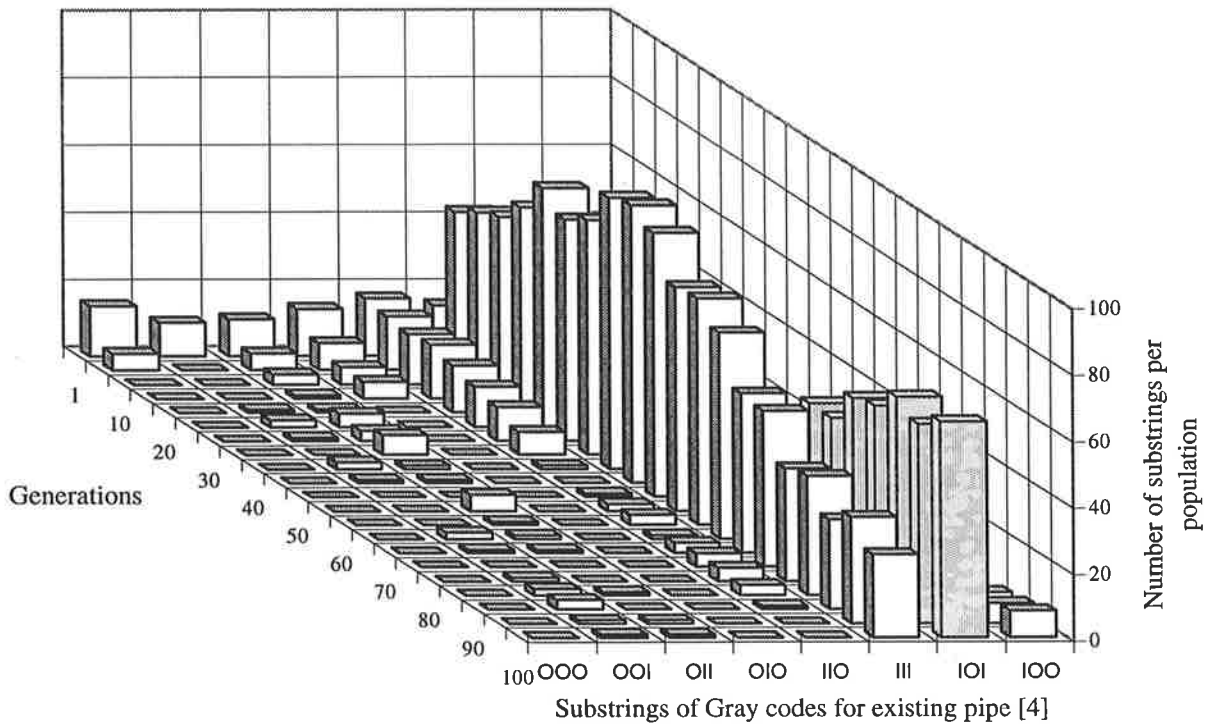


Figure 6.35 The variations with time of numbers of **decision-variable substrings of Gray codes** at the second substring position (corresponding to existing pipe [4]) for GA run CODE14 (optimum substring is **101**)

6 Improvements to the simple GA for pipe network optimisation

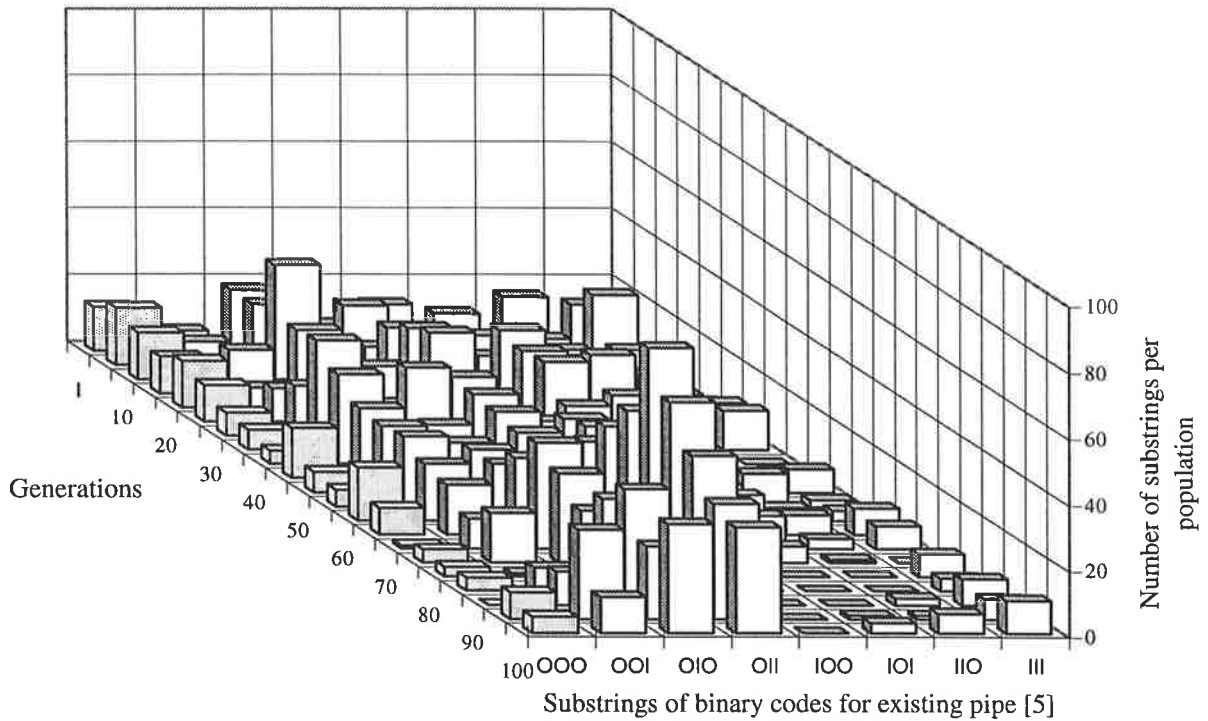


Figure 6.36 The variations with time of numbers of **decision-variable substrings of binary codes** at the third substring position (corresponding to existing pipe [5]) for GA run CODE4 (optimum substring is **000**)

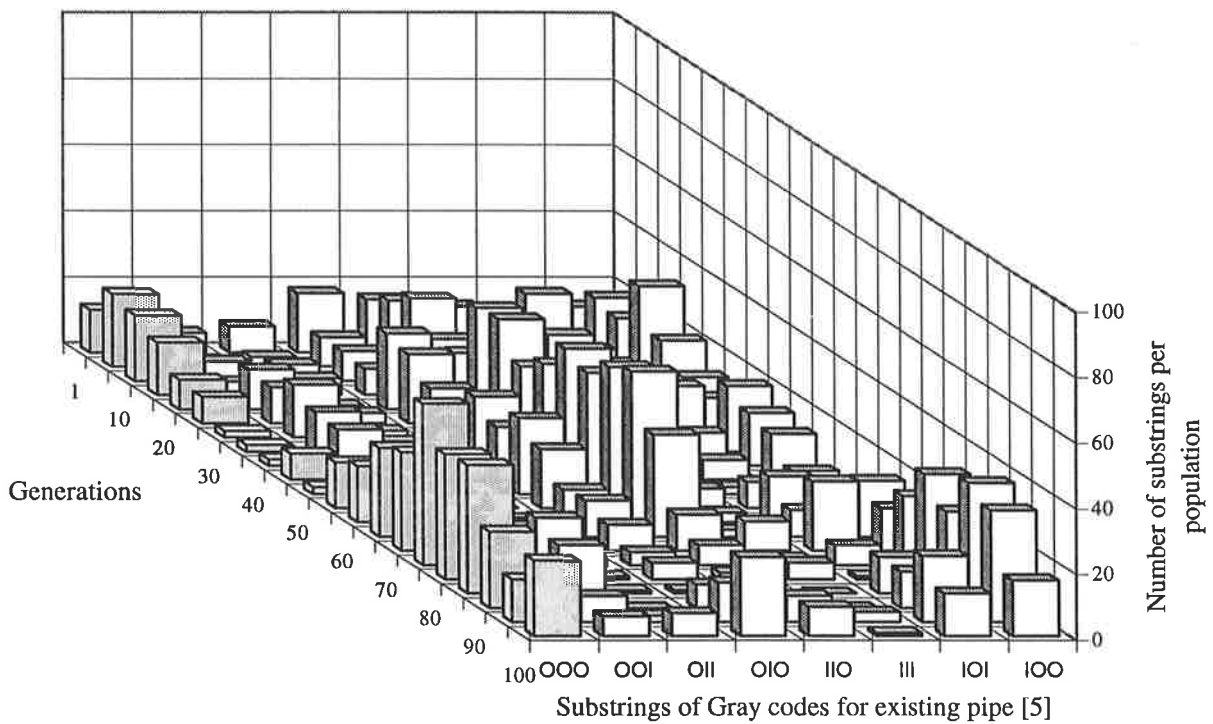


Figure 6.37 The variations with time of numbers of **decision-variable substrings of Gray codes** at the third substring position (corresponding to existing pipe [5]) for GA run CODE14 (optimum substring is **000**)

6 Improvements to the simple GA for pipe network optimisation

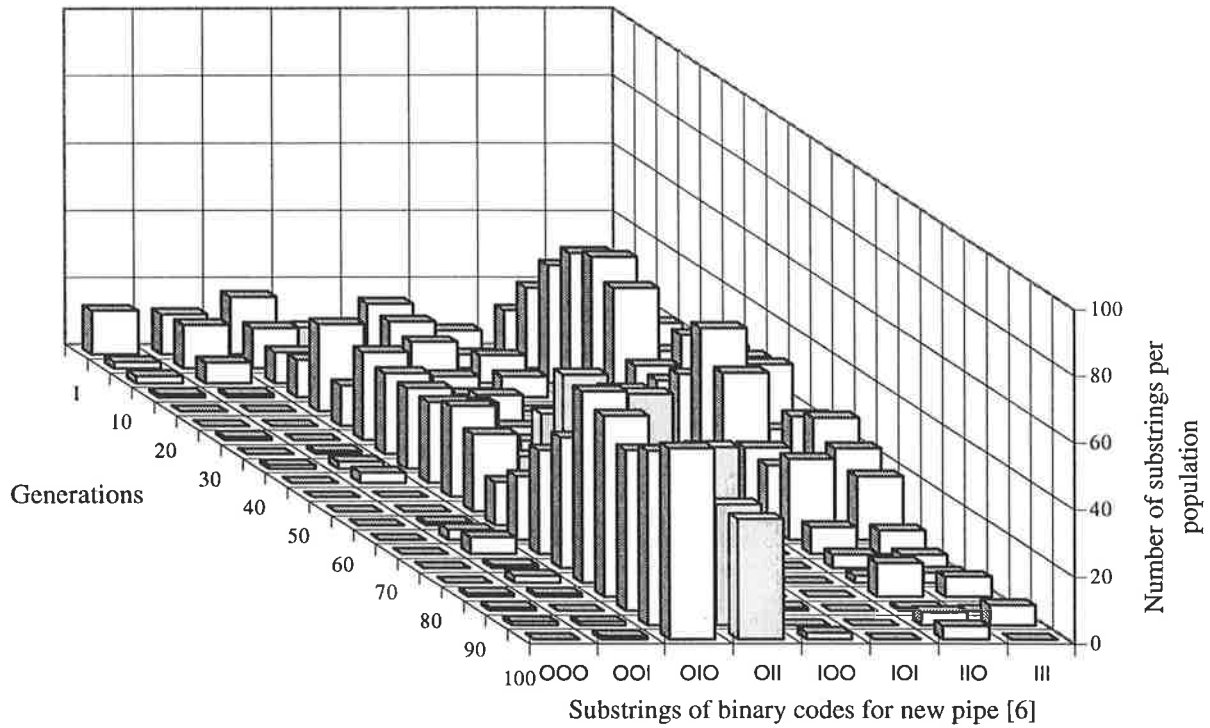


Figure 6.38 The variations with time of numbers of **decision-variable substrings of binary codes** at the fourth substring position (corresponding to new pipe [6]) for GA run CODE4 (optimum substring is **011**)

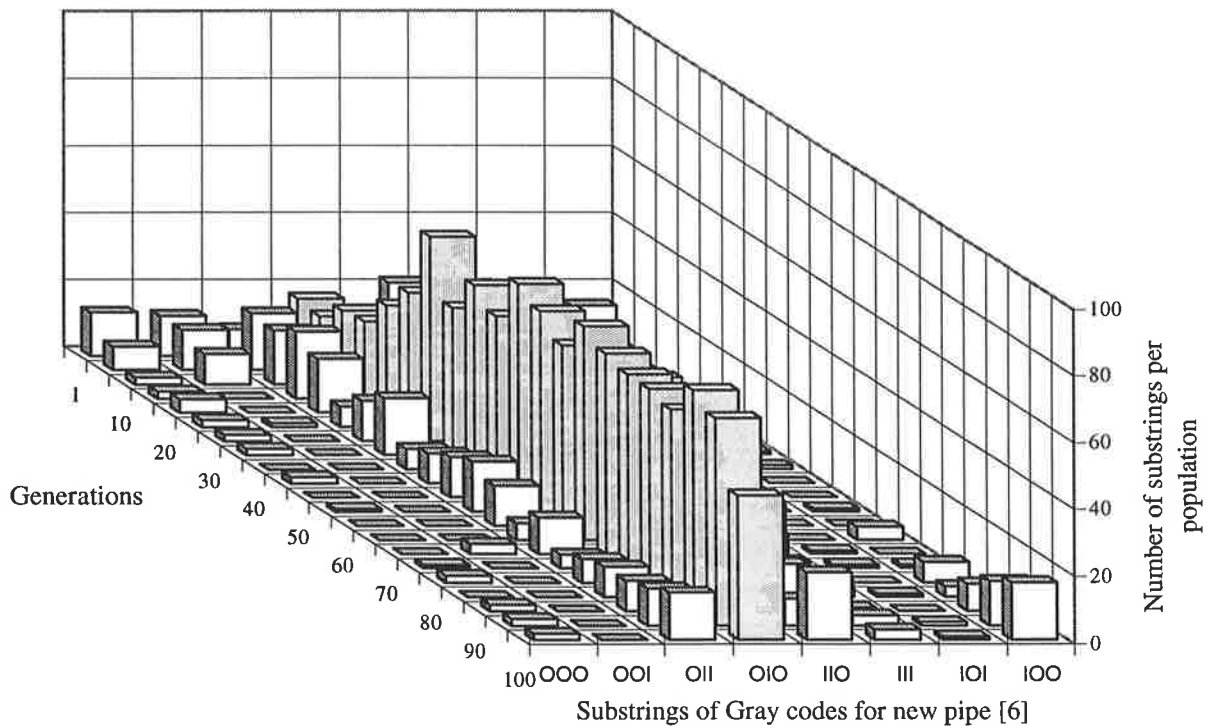


Figure 6.39 The variations with time of numbers of **decision-variable substrings of Gray codes** at the fourth substring position (corresponding to new pipe [6]) for GA run CODE14 (optimum substring is **010**)

6 Improvements to the simple GA for pipe network optimisation

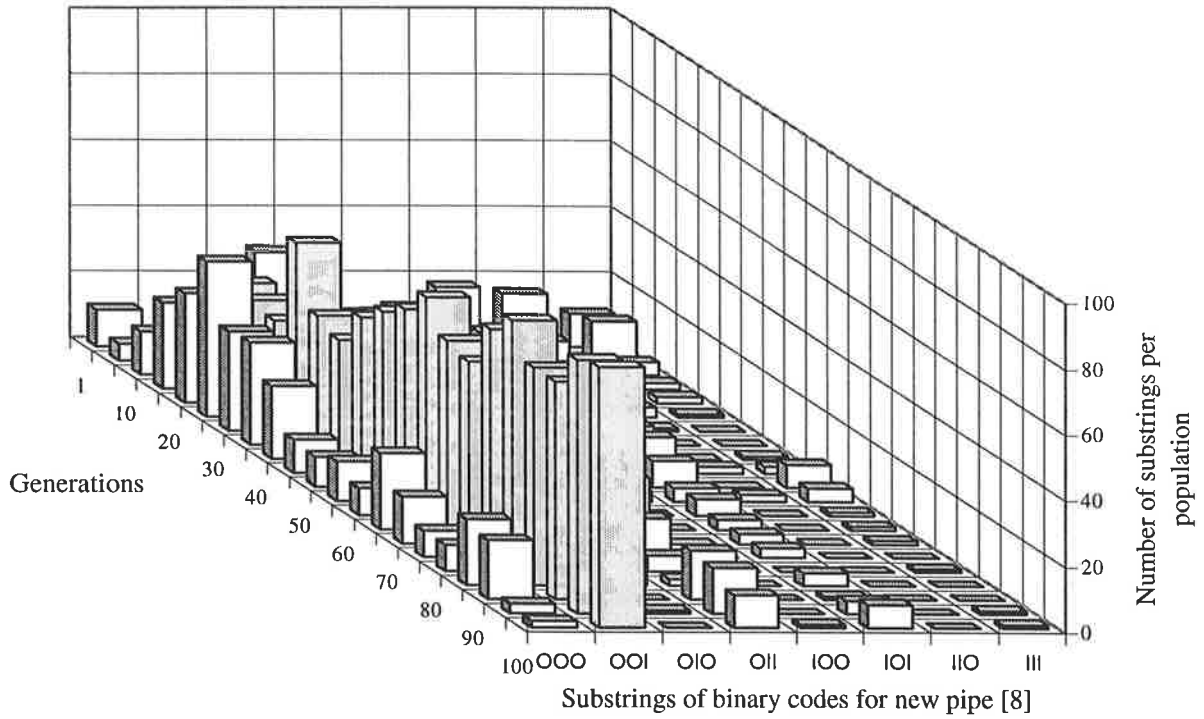


Figure 6.40 The variations with time of numbers of **decision-variable substrings of binary codes** at the fifth substring position (corresponding to new pipe [8]) for GA run CODE4 (optimum substring is **001**)

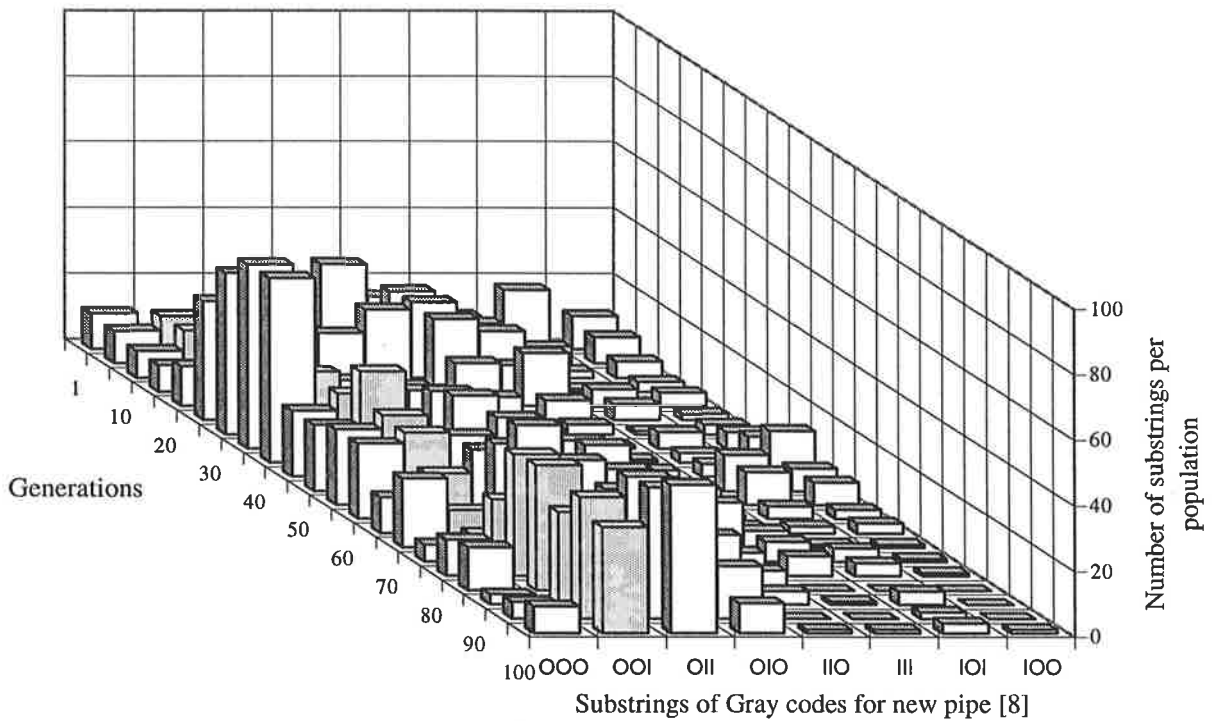


Figure 6.41 The variations with time of numbers of **decision-variable substrings of Gray codes** at the fifth substring position (corresponding to new pipe [8]) for GA run CODE14 (optimum substring is **001**)

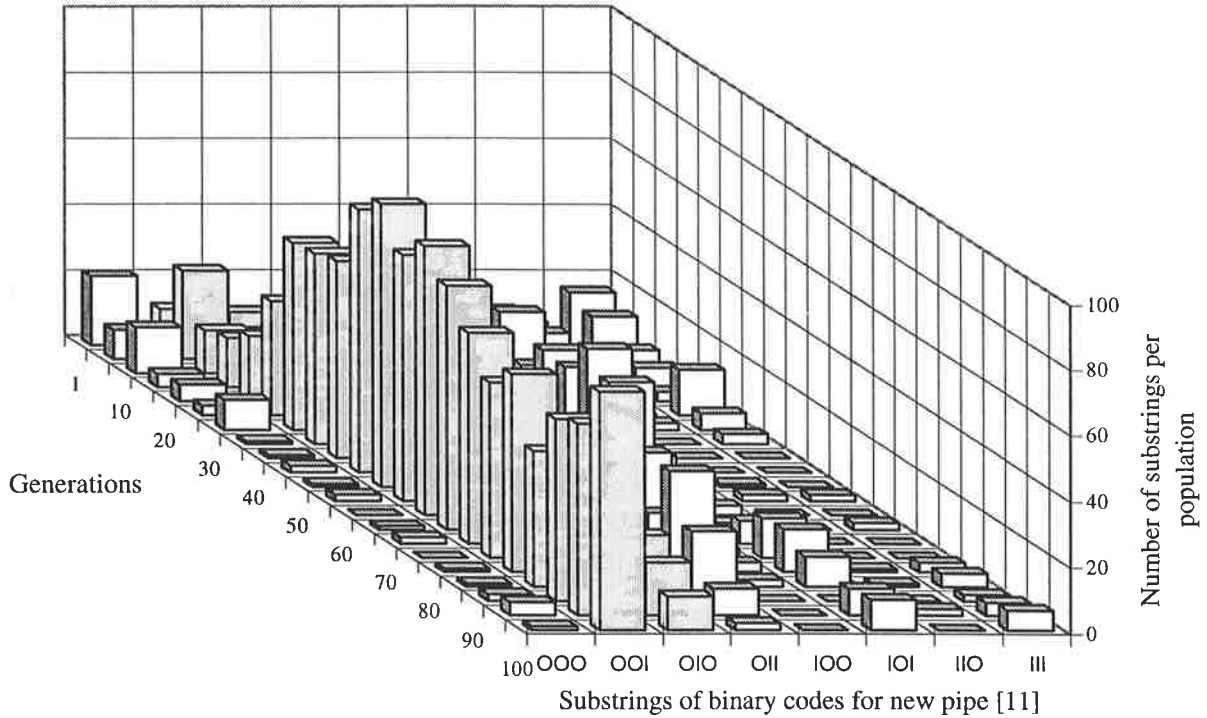


Figure 6.42 The variations with time of numbers of **decision-variable substrings of binary codes** at the sixth substring position (corresponding to new pipe [11]) for GA run CODE4 (optimum substring is **001** or **010**)

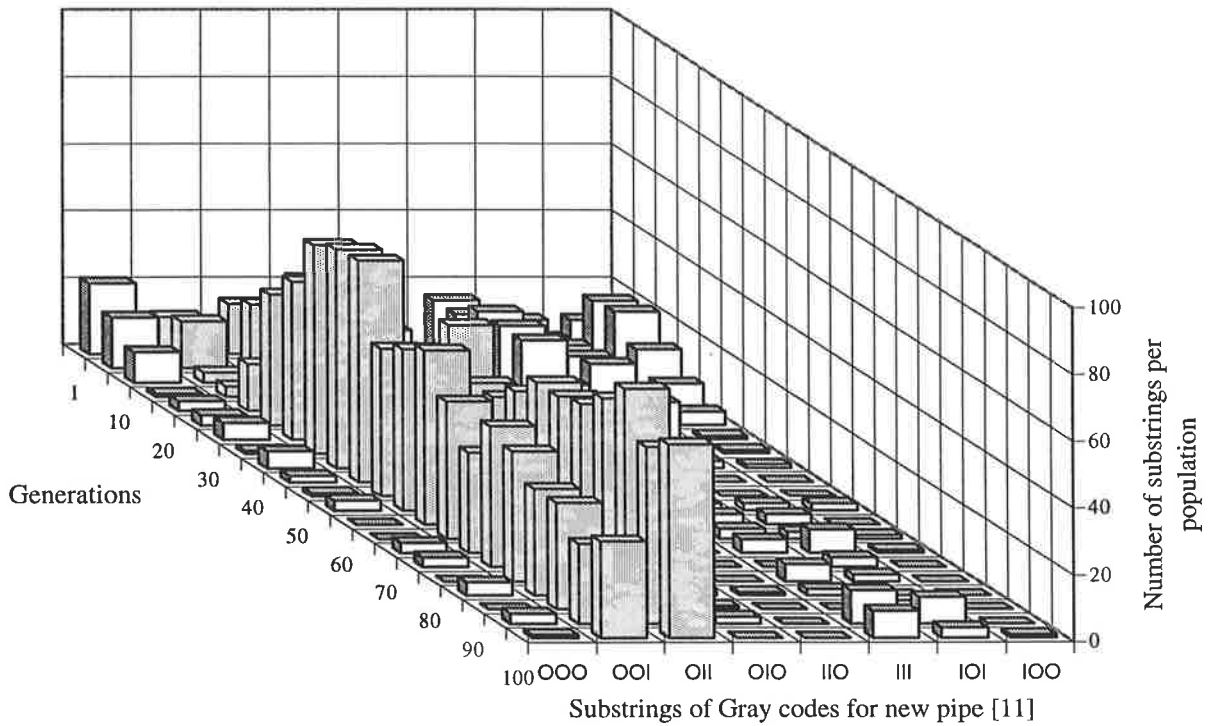


Figure 6.43 The variations with time of numbers of **decision-variable substrings of Gray codes** at the sixth substring position (corresponding to new pipe [11]) for GA run CODE14 (optimum substring is **001** or **011**)

6 Improvements to the simple GA for pipe network optimisation

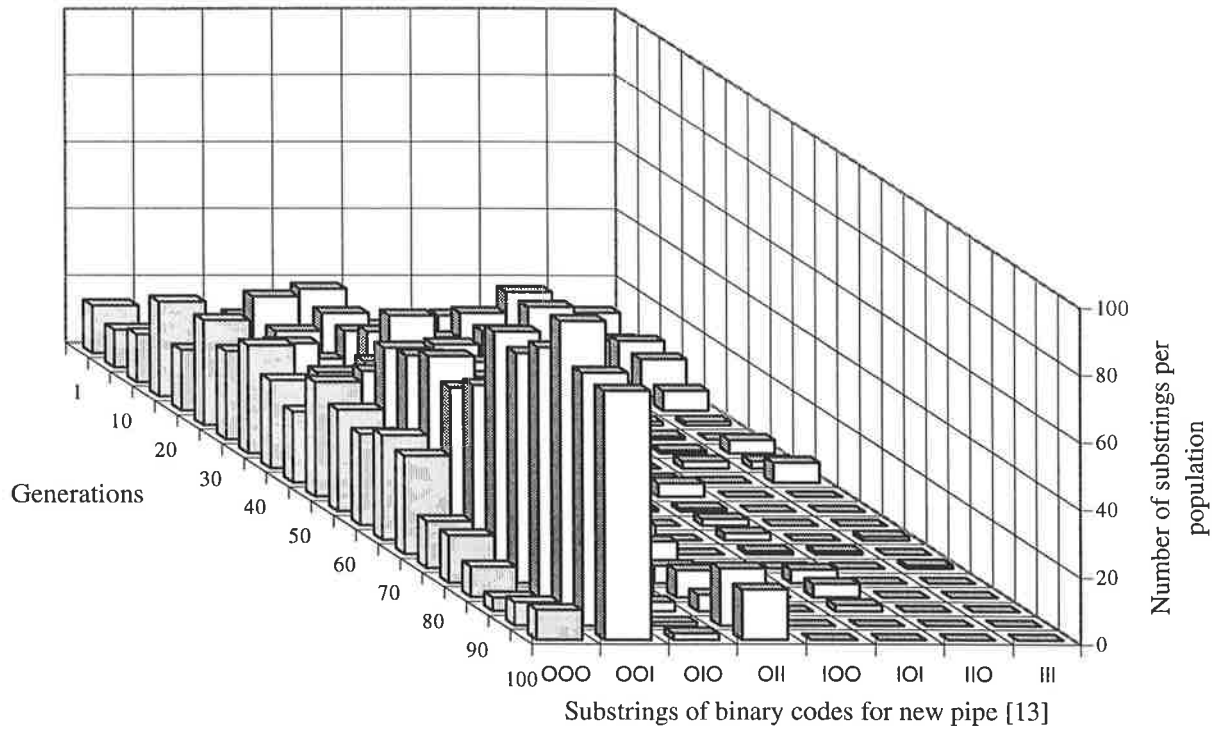


Figure 6.44 The variations with time of numbers of **decision-variable substrings of binary codes** at the seventh substring position (corresponding to new pipe [13]) for GA run CODE4 (optimum substring is **000**)

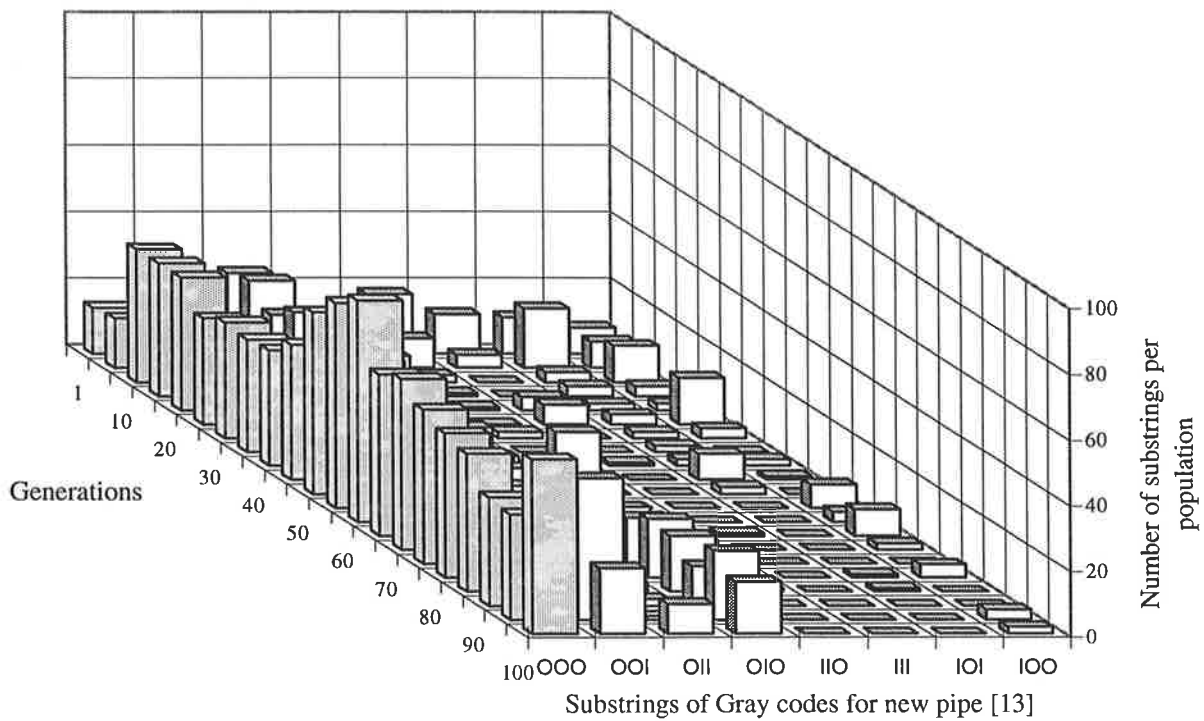


Figure 6.45 The variations with time of numbers of **decision-variable substrings of Gray codes** at the seventh substring position (corresponding to new pipe [13]) for GA run CODE14 (optimum substring is **000**)

6 Improvements to the simple GA for pipe network optimisation

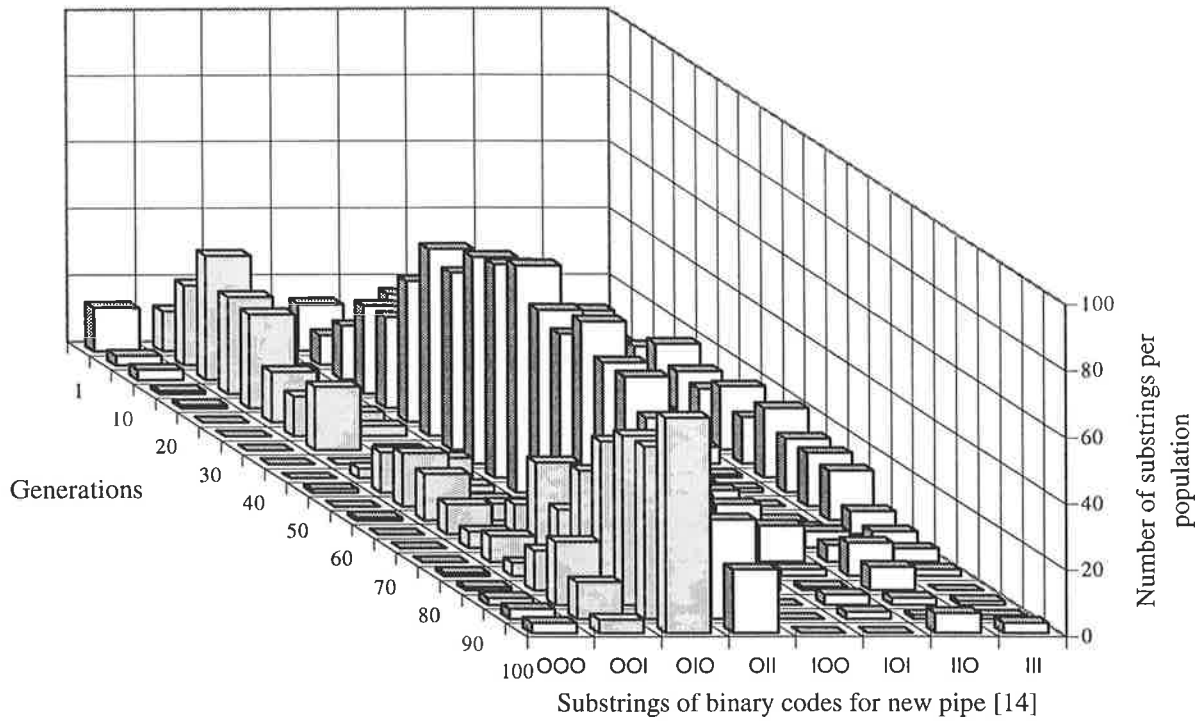


Figure 6.46 The variations with time of numbers of **decision-variable substrings of binary codes** at the last substring position (corresponding to new pipe [14]) for GA run CODE4 (optimum substring is **OIO** or **OOI**)

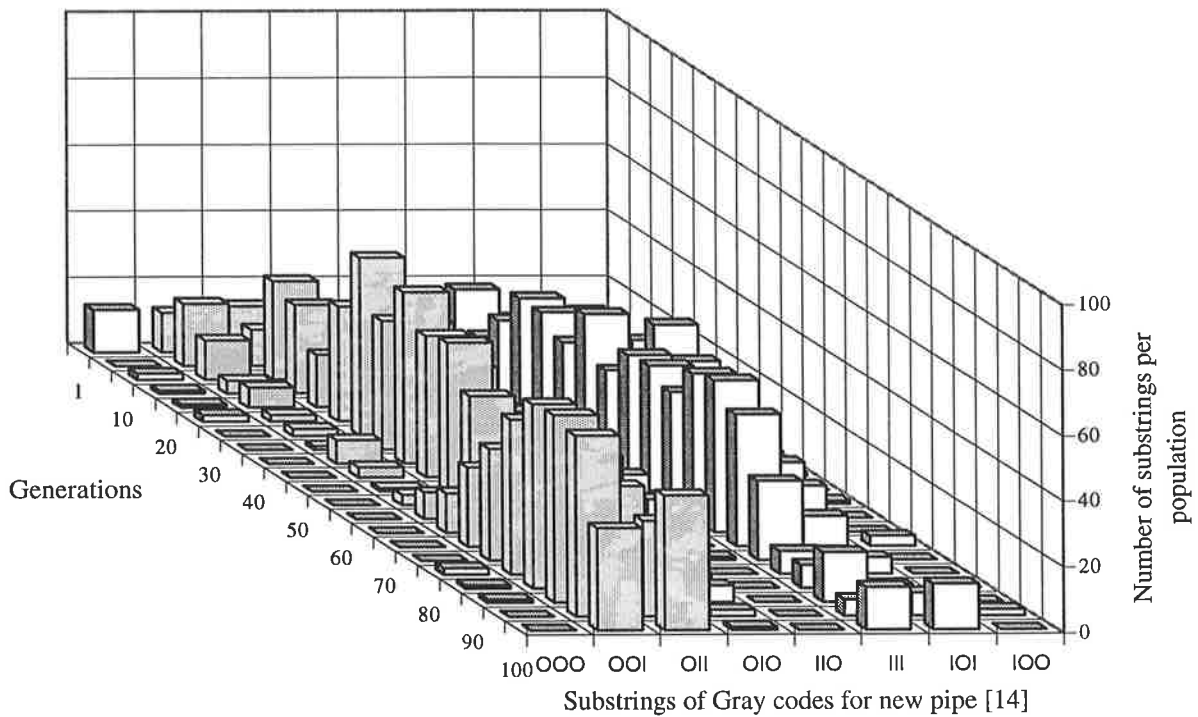


Figure 6.47 The variations with time of numbers of **decision-variable substrings of Gray codes** at the last substring position (corresponding to new pipe [14]) for GA run CODE14 (optimum substring is **O11** or **OOI**)

6 Improvements to the simple GA for pipe network optimisation

counts the number of copies of the eight possible substrings of Gray codes at the first substring position for GA run CODE14. In Figure 6.32, the substrings of binary code in the first substring position are eventually dominated by the substring 000 which represents the optimal substring code ('do nothing' for existing pipe [1]) for this substring position. Similarly in Figure 6.33, the substrings of Gray code in the first substring position are eventually dominated by the optimal substring 000, however, in both cases the substring code representing the cleaning option (010 for the binary codes and 011 for Gray codes) is prominent for some time.

The Gray codes were in greater danger of overlooking the optimal substring code 000 in the first substring position, as the Gray substring code 011 associated with cleaning pipe [1] is a greater Hamming distance from the optimal substring code 000 than the binary substring code 010. The Gray codes are disadvantaged in this situation. A new bias may have been introduced into the coding scheme when the cleaning alternative for pipe [1] was ranked after duplication with a 6" pipe in the list of decision variable choices in Table 6.24. Fortunately, the genetic algorithm is robust enough to overcome such biases that are inadvertently introduced into the GA formulation.

In the second substring position (Figures 6.34 and 6.35), the number of optimal substrings associated with duplicating the existing pipe [4] with a 14" diameter pipe (110 for binary codes and 101 for Gray codes) gather numbers late in the GA run, however, the substring codes associated with duplicating pipe [4] with a 12" pipe (101 for binary codes and 111 for Gray codes) are prominent for a time. The prominence of the substring codes representing the duplication of pipe [4] with a 12" pipe in the second substring position lasted for about as long as the prominence of the substring codes representing cleaning pipe [1] in the first substring position. Early in both GA runs CODE4 and CODE14, quite good solutions that cleaned existing pipe [1] and duplicated pipe [4] with a 12" diameter pipe (such as solutions 27, 28, 32 and 33 in Table 5.4) are prominent. However, eventually both GA populations were able to shift to the best regions of the solution space.

In the Gessler network, pipes [1] and [4] represent the alternative main supply lines, and the choice of pipe size for pipe [4] is considerably dependent on the choice of pipe size for pipe [1]. In the pipe network optimisation problem, it is evident that the fitness of a particular decision variable choice is dependent on the other decision variable choices in the same solution and some relationships between decision variable choices are stronger than others.

Both GA runs of binary code and Gray code are inconclusive in their search for the optimal substring code for the third substring position (associated with the existing pipe [5]) as shown in Figures 6.36 and 6.37. In the Gessler problem, the best decision variable choice for pipe [5]

is likely to be highly dependent on the preferred choices for the main lines [1] and [4]. In addition, the position and shorter length of pipe [5] compared to pipes [1] and [4] would suggest the choice for pipe [5] in the solution is less significant than the choices for pipes [1] and [4]. Therefore, confusion in the third substring position (pipe [5]) may be related to uncertainties in the first and second substring positions (pipes [1] and [4] respectively). This raises concerns about some pieces of code carrying more weight than other pieces of code. A possible solution to this problem is to split long lengths of pipe into smaller segments.

The fourth substring position provides the best chance to see any evidence of the Hamming Cliff obstructing populations of strings formed with binary codes, since the optimal substring code in the fourth substring position 011 is on one side of the 011-100 step. In Figure 6.38, there are good numbers of the optimal substring code 011 and virtually no occurrences of the 100 substring code. Fortunately in this case, the population has assumed the right side of the Hamming Cliff. The Gray substring codes for the fourth substring position in Figure 6.39 collect about the optimal substring code 010. Gray codes eliminate the potential hazard of the Hamming Cliff of binary coding.

The exhaustive enumeration of the Gessler problem in Section 5.2.2 identified two optimum network solutions. The solutions differ in that Solution 1 assigns pipe [11] an 8" diameter pipe and pipe [14] a 10" diameter pipe, while the alternative Solution 2 assigns pipe [11] a 10" pipe and pipe [14] an 8" pipe. The strings of Gray code in Figure 6.43 carry healthy numbers of both optimal substring codes 001 and 011 at the sixth substring position (corresponding to new pipe [11]). In contrast, the strings of binary code in Figure 6.42 carry high numbers of the optimal substring code 001 and low numbers of the optimal substring code 010 at the sixth substring position. A similar condition is exhibited for the last substring position (corresponding to new pipe [14]) in Figures 6.46 and 6.47.

6.5.6 An ideal coded structure

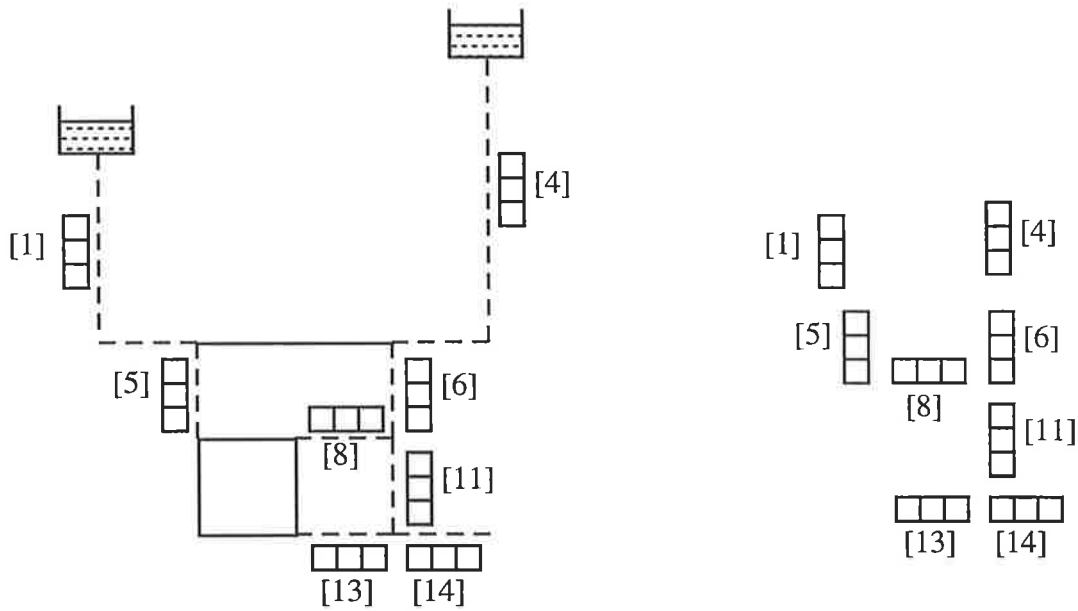
If a coded string structure is used to represent solutions, the best arrangement of the decision variable substrings within the string is a significant issue for the use of the traditional GA for pipe network optimisation. The pipes nearby in the pipe system which are likely to have an influence on each other in the economics and hydraulics of the pipe network system design should be allocated substring positions nearby on the string.

The original substring arrangement shown in Figure 6.26 was found to be superior to the alternative substring arrangement considered in Figure 6.27 for the GA runs performed on the Gessler network. The more influential pipes were allocated adjacent substring positions at the start of the string providing the opportunity for the development of short, highly fit string

6 Improvements to the simple GA for pipe network optimisation

similarities at these string positions. The string similarities would be less disrupted by the effects of crossover and would be propagated by reproduction to quickly establish themselves in the population. It would be more difficult to determine the best substring arrangement for a larger and more complicated system. Perhaps advanced GA operators such as inversion operators or Goldberg's messy genetic algorithms hold the key to this ordering problem.

Ideally, a coded structure in the form of an array of symbols which more closely represents the layout of the pipe network itself would best utilise the interactions and relationships between decision variable choices in the solution. The development of a coded array structure for the Gessler network is shown in Figures 6.48 and 6.49.



Gessler network expansions problem

Construction of the ideal coded structure

Figure 6.48 An ideal coded structure representation of the Gessler problem

0	*	*	*	*	*	*	1	*	*
0	*	*	*	*	*	*	1	*	*
0	*	*	*	*	*	*	0	*	*
*	0	*	*	*	*	0	*	*	*
*	0	*	*	*	*	1	*	*	*
*	0	*	*	*	*	1	*	*	*
*	*	*	0	0	1	*	*	*	*
*	*	*	*	*	*	0	*	*	*
*	*	*	*	*	*	0	*	*	*
*	*	*	*	*	*	1	*	*	*
*	*	*	0	0	0	*	0	1	0

Figure 6.49 Array representing ideal coded structure

6 Improvements to the simple GA for pipe network optimisation

The coded array in Figure 6.49 represents one of the optimum solutions for the Gessler problem with substrings of binary codes. The GA operators of selection and mutation could be performed in the usual way. To implement crossover, one or more crossover cuts in the array could be selected randomly across the expected direction of flow in the pipe network system as shown in Figure 6.50.

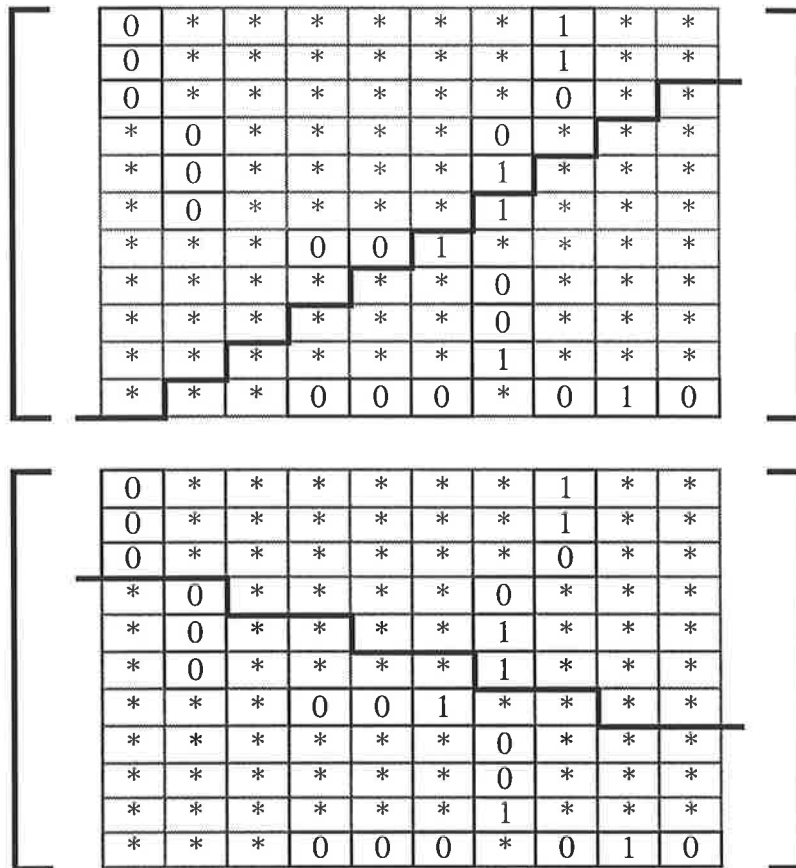


Figure 6.50 Possible crossover cuts for the ideal coded structure

No testing has been carried out on the proposed coded array. It is thought that its potential would not be as apparent for the relatively simple Gessler problem. It is potentially the best form of coded solution for large, complex water distribution systems. The coded array should promote the relationships between symbols related in the solution and could be most effective for large, highly interconnected systems with many loops and with a number of decision variable substrings associated with pipes and other system components. The symbols representing other system components (such as tank sizes at prospective tank locations) could be positioned in the array according to their location in the network.

6.6 Crossover Mechanisms

Crossover is the genetic algorithm mechanism of recombination which operates on two parent coded strings to create two new offspring coded strings for the new generation by exchanging segments from the parent coded strings. The crossover points are randomly selected positions in the fixed-length string that mark the segments to be exchanged. There may be one or more crossover points.

Simple one-point crossover is used in the traditional GA model developed in Chapter 5 for pipe network optimisation. The action of one-point crossover is demonstrated in Figure 6.51 for two 24-bit coded string solutions.

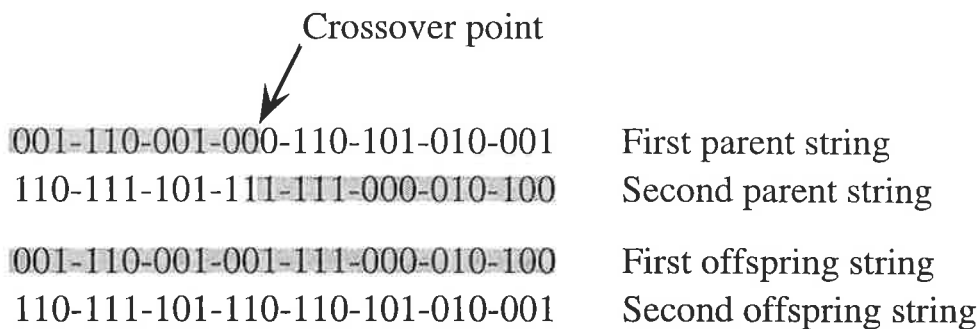


Figure 6.51 Simple one-point crossover

Two-point crossover has much theoretical and experimental support among GA researchers (Holland, 1975; DeJong, 1975). Two crossover points are randomly selected on the string and the two points mark the beginning and the end of a segment to be exchanged by the parent strings. In two-point crossover, the coded string is treated as a circular string (no beginning or end). The action of two-point crossover is demonstrated in Figure 6.52.

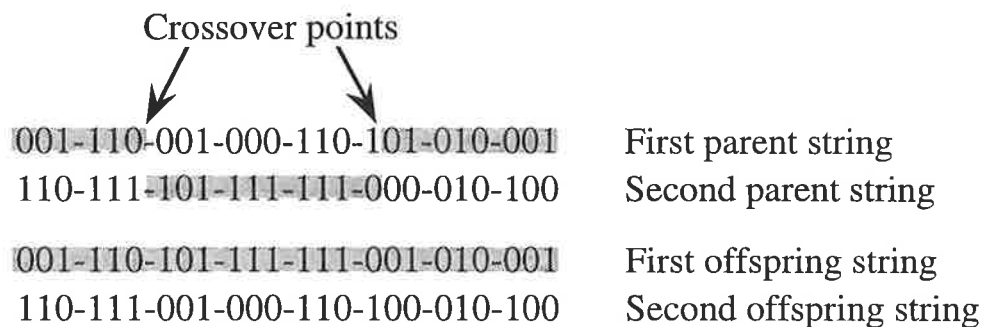


Figure 6.52 Two-point crossover

More crossover points may be selected to the point where uniform crossover occurs. In uniform crossover, bits of code are randomly exchanged. The parent strings exchange bits in corresponding bit positions randomly (or with some probability) to construct two offspring strings. Uniform crossover produces $L/2$ crossover points on average for strings of length L (Syswerda, 1989). The action of uniform crossover is demonstrated in Figure 6.53.

001-110-001-000-110-101-010-001	First parent string
110-111-101-111-111-000-010-100	Second parent string
000-111-101-101-110-101-010-100	First offspring string
111-110-001-010-111-000-010-001	Second offspring string

Figure 6.53 Uniform crossover (multiple random crossover points)

Syswerda (1989) compared one-point, two-point and uniform crossover theoretically and empirically and found in most cases, uniform crossover to be superior to two-point crossover and in turn, two-point crossover to be superior to one-point crossover. Syswerda introduced the notion of *crossover masks* to replace the traditional idea of crossover points to compare the crossover mechanisms. Uniform crossover was found to be more effective at combining schemata and exhibited superior performance applied to a diverse set of function optimisation problems.

DeJong and Spears (1990) presented an analysis of the more general picture of the interacting roles of crossover and population size. Their theoretical and experimental analyses suggest the less disruptive crossover operators such as two-point crossover are likely to perform better with larger population sizes. However, they found more disruptive crossover operators such as uniform crossover and multi-point crossover (with crossover points > 2) are likely to perform better with smaller population sizes and explain this by the increased exploration in the limited information base of small populations. The more disruptive crossover operators are less likely to produce offspring which are identical to their parents in small, homogeneous populations.

A GA search combines powers of exploitation of past results in selection and exploration of new areas of the search space by crossover and mutation. Eshelman et al. (1989) classify various crossover mechanisms by their powers of exploration. The amount of exploration provided by crossover depends on the crossover rate, and also on the amount of exploitation performed by selection. As exploitation is increased, the exploratory power of crossover is decreased, since there are fewer differences in the population which crossover can explore by recombination. The chosen coded representation of the solution has a significant affect on the exploratory powers of crossover. As an example, for traditional one-point crossover, bits which are related in the solution and far apart on the string are more likely to be separated. If

there are known relationships between bits in a coded string solution and one-point or two-point crossover is to be used, these bits should be located close to one another in the string. The experimental results of Eshelman et al. (1989) showed the traditional one-point crossover operator to be the least effective crossover mechanism.

Other types of crossover mechanisms have been proposed for the exchange of information between parent strings to create offspring strings. Schaffer and Morishima (1987) propose a GA model incorporating an adaptive crossover mechanism they called a 'genetic algorithm with punctuated crossover' (GAPC). The GAPC model doubles the length of the original coded string solution by attaching a second coded string to the end of the first string. The second coded string represents the crossover punctuation (the 1's in the second coded string of binary bits represent the crossover points). When the starting population is generated, the bits of the first part of the coded string representing the solution are randomly generated, but the 1's in the second part of the string are generated with some specified probability. The strings are interpreted as the first coded string separated by punctuation marks at the crossover points. The punctuation marks are inherited by the offspring strings and the punctuation marks are lost if a solution is not selected for the new generation by the selection operator. The mutation operator may mutate bits on the first part of the string (the solution) or the second part of the string (the crossover points).

Schaffer and Morishima (1987) present empirical evidence to suggest their model performs as well or better than the traditional genetic algorithm model. As the populations of coded string solutions evolve, the punctuation part of the coded string representing the distribution of crossover points is expected to represent the best points to perform crossover.

Other crossover mechanisms such as cycle crossover, partially-matched crossover and order crossover reviewed in Goldberg (1989) combine properties of crossover and reordering operators such as inversion in one step.

This following analysis investigates the performance of simple one-point crossover, multi-point crossover and uniform crossover for the pipe network optimisation application. In addition, crossover operators which choose crossover points at any bit position of the string or only at the boundaries of the decision-variable substrings are considered. Figure 6.54 demonstrates the action of two-point crossover occurring at crossover points at the substring boundaries.

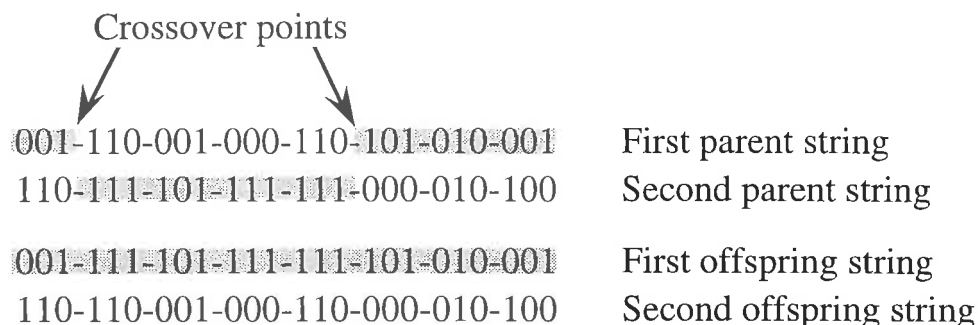


Figure 6.54 Two-point crossover (crossover points at substring boundaries)

6.6.1 GA model runs to compare crossover mechanisms

A series of GA model runs CROSS1-CROSS65 were conducted to measure the performance of the various crossover mechanisms for the application of the GA model to the Gessler pipe network optimisation problem (refer to Table 6.2). The GA runs CROSS1-CROSS5 used the traditional GA crossover mechanism of one-point crossover. The GA runs CROSS1-CROSS5 are identical to GA runs PEN11-PEN15, GA runs FIT1-FIT5 and GA runs CODE1-CODE5 presented earlier in this chapter. The GA runs CROSS11-CROSS15 use the two-point crossover operator. The GA runs CROSS21-CROSS25 use the four-point crossover operator and GA runs CROSS31-CROSS35 use the uniform crossover operator.

The GA model runs CROSS41-CROSS45, CROSS51-CROSS55 and CROSS61-CROSS65 use the one-point crossover, two-point crossover and uniform crossover mechanisms respectively, but the crossover points are chosen randomly at the decision-variable substring boundaries. The results of the GA model runs are summarised in Tables 6.30-6.36.

Lowest cost solutions

The traditional one-point crossover operator (GA runs CROSS1-CROSS5) is the most successful, finding the global optimum in three GA runs (Table 6.30). None of the other crossover mechanisms find the global optimum three times. The two-point crossover operator crossing at random bit positions and both uniform crossover mechanisms obtain the optimum solution twice in five attempts. The least effective crossover mechanism is two-point crossover at substring boundaries (GA runs CROSS51-CROSS55) which fails to find the global optimum solution in five attempts. The high crossover rate, $p_c=1.0$ with population size, $N=100$ is the most successful parameter combination when used with mutation rate, $p_m=0.01$ (fourth GA parameter set in Table 6.1), and yet this is the least successful combination of crossover rate and population size when used with low mutation rate, $p_m=0.005$ (fifth GA parameter set).

Table 6.30 Search results for genetic algorithm model runs CROSS1-CROSS5

One-point crossover. GA runs CROSS1-CROSS5 equivalent to GA runs PEN11-PEN15.					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	CROSS1 $N=50$, $p_c=0.75$	CROSS2 $p_c=0.75$	CROSS3 $p_c=0.5$	CROSS4	CROSS5 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.7503‡ (59)	1.7503‡ (96)	1.7503‡ (59)	1.7725 (93)	1.8807 (82)
Lowest cost GA design generated (Table 5.4)	2	2	1	3	-
Lowest average generation cost (\$m) (after - generations)	2.045 (212)	2.139 (73)	2.038 (176)	2.271 (92)	2.274 (86)
Ultimate offline performance (\$m)	1.963	1.896	1.916	1.935	2.026
Ultimate online performance (\$m)	2.560	2.557	2.475	2.766	2.694

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.31 Search results for GA model runs CROSS11-CROSS15

Two-point crossover					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	CROSS11 $N=50$, $p_c=0.75$	CROSS12 $p_c=0.75$	CROSS13 $p_c=0.5$	CROSS14	CROSS15 $p_m=0.005$
Number of generations required	265	133	194	100	100
Lowest solution cost (\$m) (after - generations)	1.7503‡ (152)	1.8154† (106)	1.7725 (133)	1.7503‡ (84)	1.9288 (70)
Lowest cost GA design generated (Table 5.4)	2	2 †	3	1	-
Lowest average generation cost (\$m) (after - generations)	2.124 (99)	2.218 (119)	2.050 (168)	2.357 (62)	2.480 (54)
Ultimate offline performance (\$m)	1.998	1.958	1.909	1.988	2.156
Ultimate online performance (\$m)	2.581	2.735	2.564	2.911	2.875

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

† Infeasible design. The solution cost includes the penalty cost. The design is given in Table 5.29. GA run CROSS12 determined a best cost feasible design for \$1.8385m (Solution 27 in Table 5.4) after 69 generations.

Table 6.32 Search results for GA model runs CROSS21-CROSS25

Four-point crossover					
GA RUNS	CROSS21	CROSS22	CROSS23	CROSS24	CROSS25
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	267	133	196	100	100
Lowest solution cost (\$m)	1.8115	1.7503 [‡]	1.8300	1.7725	1.8871
(after - generations)	(173)	(52)	(154)	(93)	(66)
Lowest cost GA design generated (Table 5.4)	10	1	22	3	-
Lowest average generation cost (\$m)	2.158	2.165	2.101	2.570	2.190
(after - generations)	(179)	(132)	(161)	(97)	(74)
Ultimate offline performance (\$m)	2.051	1.930	1.944	2.066	2.037
Ultimate online performance (\$m)	2.736	2.790	2.464	3.039	2.728

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.33 Search results for GA model runs CROSS31-CROSS35

Uniform crossover					
GA RUNS	CROSS31	CROSS32	CROSS33	CROSS34	CROSS35
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	198	100	100
Lowest solution cost (\$m)	1.9393	1.8300	1.7503 [‡]	1.7503 [‡]	1.8115
(after - generations)	(193)	(21)	(83)	(89)	(75)
Lowest cost GA design generated (Table 5.4)	-	21	1	1	11
Lowest average generation cost (\$m)	2.367	2.520	2.090	2.361	2.214
(after - generations)	(238)	(121)	(138)	(87)	(100)
Ultimate offline performance (\$m)	2.168	2.098	1.979	2.103	1.996
Ultimate online performance (\$m)	2.744	2.997	2.576	3.214	2.875

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.34 Search results for GA model runs CROSS41-CROSS45

One-point crossover at decision-variable substring boundaries					
GA RUNS	CROSS41	CROSS42	CROSS43	CROSS44	CROSS45
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.7503‡ (242)	1.8047† (125)	1.7725 (86)	1.8871 (73)	1.8385 (43)
Lowest cost GA design generated (Table 5.4)	1	1 †	3	-	27
Lowest average generation cost (\$m) (after - generations)	1.998 (242)	2.322 (133)	2.023 (140)	2.372 (75)	2.067 (98)
Ultimate offline performance (\$m)	1.976	1.979	1.941	2.028	1.982
Ultimate online performance (\$m)	2.552	2.710	2.467	2.832	2.571

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

† Infeasible design. The solution cost includes the penalty cost. The design is given in Table 5.29. GA run CROSS42 determined a best cost feasible design for \$1.8337m (Solution 25 in Table 5.4) after 60 generations.

Table 6.35 Search results for GA model runs CROSS51-CROSS55

Two-point crossover at decision-variable substring boundaries					
GA RUNS	CROSS51	CROSS52	CROSS53	CROSS54	CROSS55
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	265	133	198	100	100
Lowest solution cost (\$m) (after - generations)	1.7910 (216)	1.7910 (124)	1.7910 (126)	1.8385 (54)	1.8997 (30)
Lowest cost GA design generated (Table 5.4)	5	5	5	27	-
Lowest average generation cost (\$m) (after - generations)	2.047 (219)	2.193 (122)	1.972 (165)	2.417 (83)	2.232 (96)
Ultimate offline performance (\$m)	1.921	1.892	1.906	2.026	2.099
Ultimate online performance (\$m)	2.486	2.560	2.404	2.851	2.821

‡ Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.36 Search results for GA model runs CROSS61-CROSS65

Uniform crossover at decision-variable substring boundaries					
GA RUNS	CROSS61	CROSS62	CROSS63	CROSS64	CROSS65
Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	$N=50$, $p_c=0.75$	$p_c=0.75$	$p_c=0.5$		$p_m=0.005$
Number of generations required	267	133	199	100	100
Lowest solution cost (\$m)	1.7950 [†]	1.7503 [‡]	1.8115	1.7725	1.7725
(after - generations)	(211)	(93)	(127)	(58)	(79)
Lowest cost GA design generated (Table 5.4)	1 [†]	1	10	4	3
Lowest average generation cost (\$m)	2.002	2.122	2.041	2.257	2.023
(after - generations)	(250)	(107)	(130)	(89)	(79)
Ultimate offline performance (\$m)	1.877	1.978	1.963	1.919	1.954
Ultimate online performance (\$m)	2.471	2.692	2.451	2.760	2.592

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

[†] Infeasible design. The solution cost includes the penalty cost. The design is given in Table 5.29. GA run CROSS61 determined a best cost feasible design for \$1.8010m (Solution 9 in Table 5.4) after 167 generations.

Best of generation costs and offline performance

The less disruptive crossover mechanisms such as two-point crossover (GA run CROSS14) are more effective than the more disruptive crossovers if the crossover points may be selected at any bit position linkage (Figures 6.55 and 6.58). Conversely, the more disruptive crossover mechanisms such as uniform crossover (GA run CROSS64) display superior convergence when crossovers occur only at decision-variable substring boundaries (Figures 6.56 and 6.59).

The uniform crossover operator crossing at random bit positions (GA run CROSS34) converges at a greater rate in the second half of the GA search and eventually determines the global optimum not long after the two-point crossover mechanism in Figure 6.55. The use of one-point and two-point crossovers at substring boundaries is not effective.

6 Improvements to the simple GA for pipe network optimisation

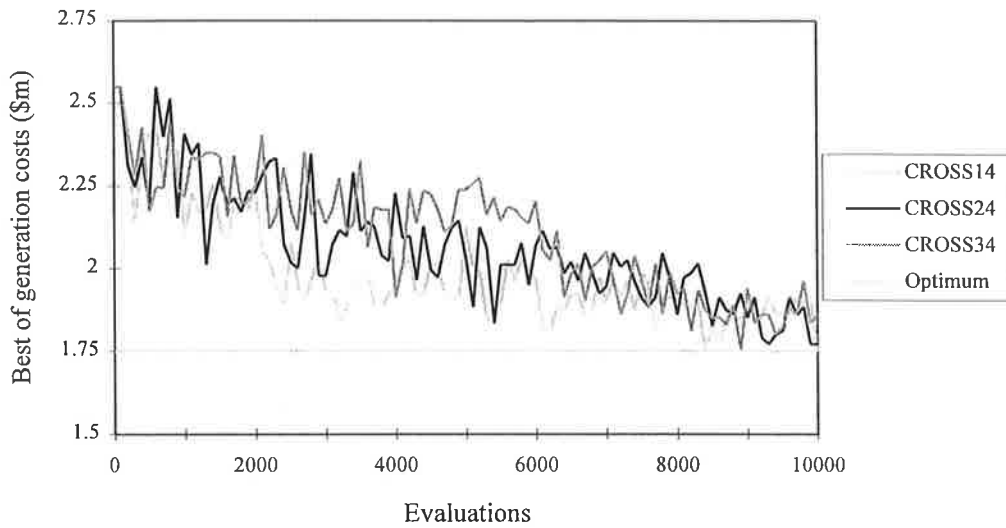


Figure 6.55 Best generation costs for GA runs CROSS14 (two-point crossover), CROSS24 (four-point crossover) and CROSS34 (uniform crossover)

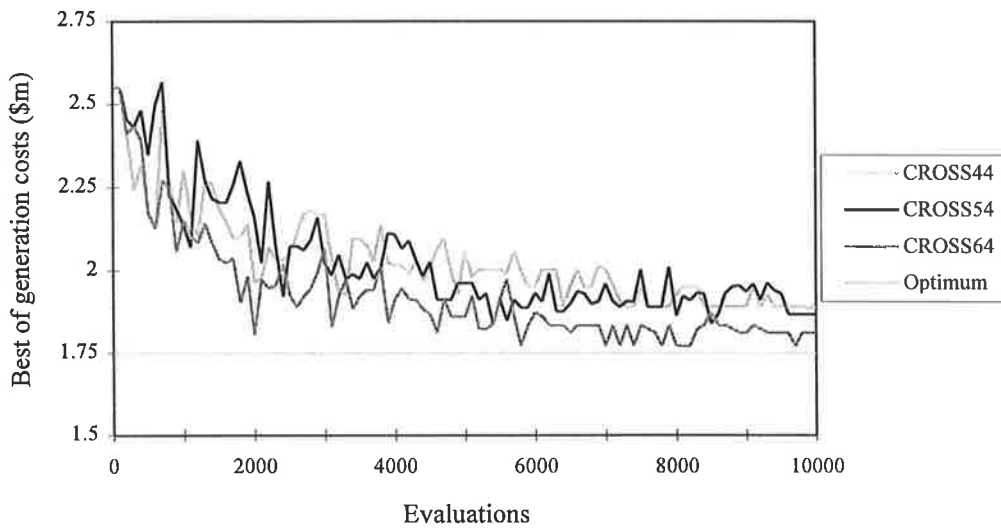


Figure 6.56 Best generation costs for GA runs CROSS44 (one-point crossover at substring boundaries), CROSS54 (two-point crossover at substring boundaries) and CROSS64 (uniform crossover at substring boundaries)

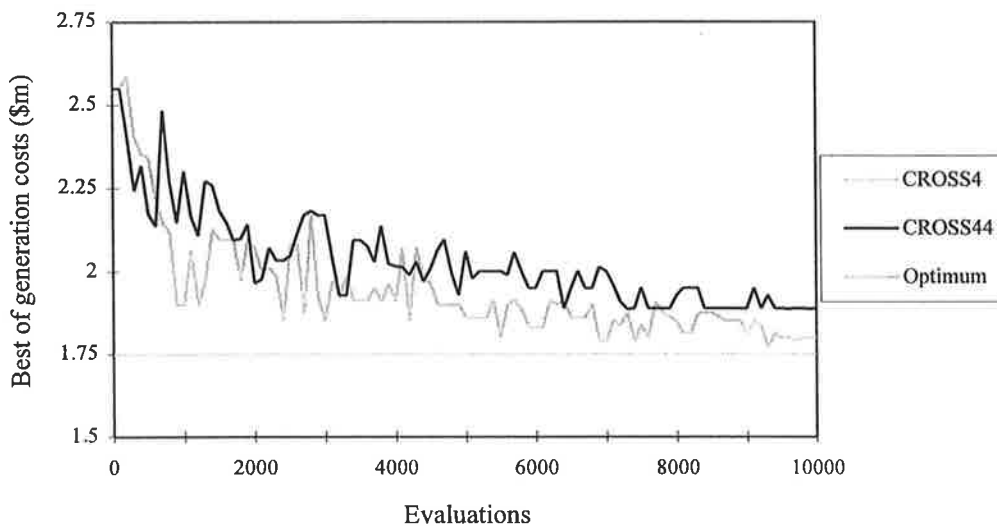


Figure 6.57 Best generation costs for GA runs CROSS4 (one-point crossover) and CROSS44 (one-point crossover at decision-variable substring boundaries)

6 Improvements to the simple GA for pipe network optimisation

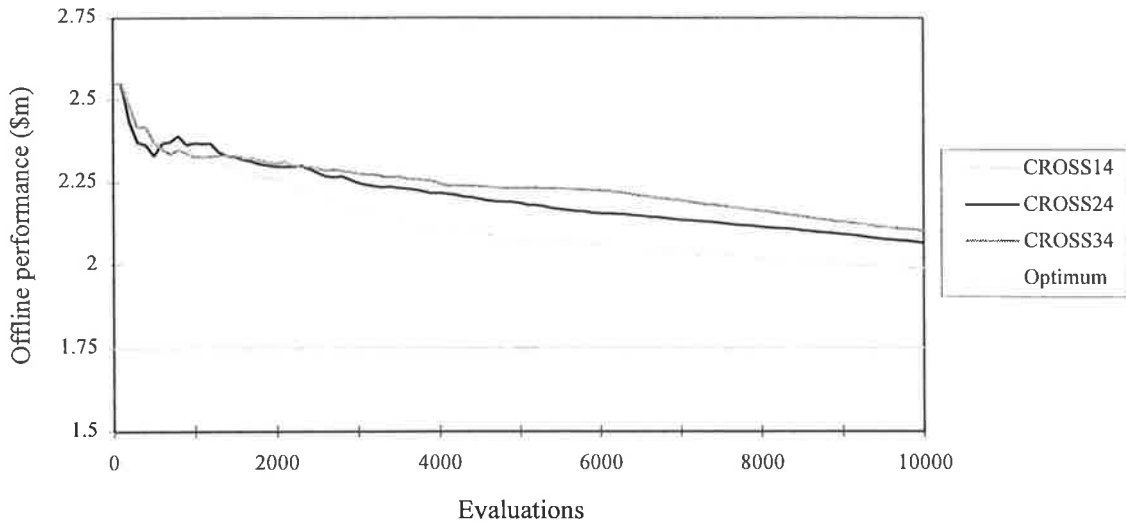


Figure 6.58 Offline performance (running average of best solution costs) for GA runs CROSS14 (two-point crossover), CROSS24 (four-point crossover) and CROSS34 (uniform crossover)

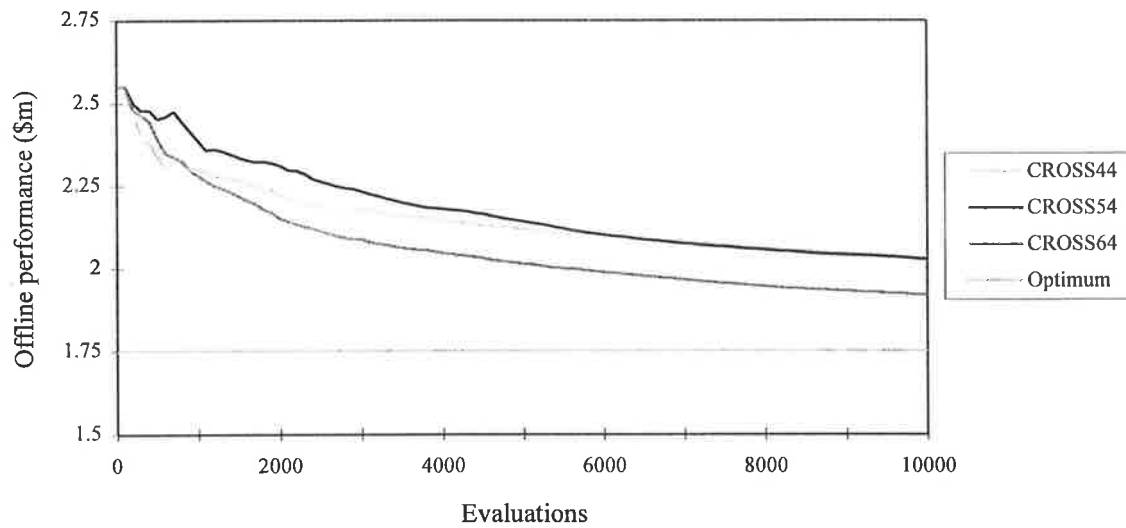


Figure 6.59 Offline performance for GA runs CROSS44 (one-point crossover at substring boundaries), CROSS54 (two-point crossover at substring boundaries) and CROSS64 (uniform crossover at substring boundaries)

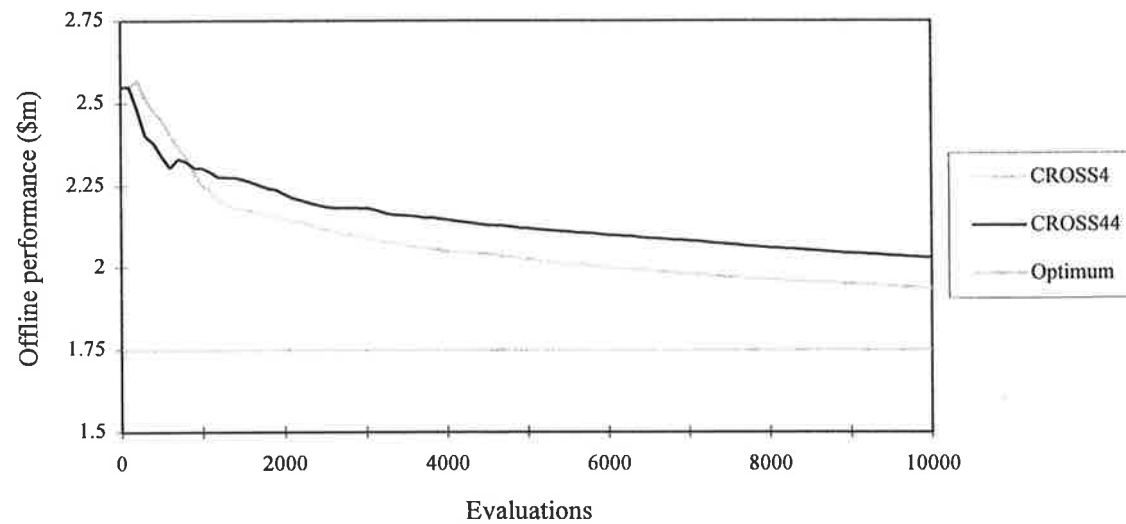


Figure 6.60 Offline performance for GA runs CROSS4 (one-point crossover) and CROSS44 (one-point crossover at decision-variable substring boundaries)

Average generation costs and offline performance

As we might expect, the more disruptive crossover mechanisms such as multi-point and uniform crossovers at bit position linkages display inferior online performance (Figures 6.61 and 6.64). Of the other crossover mechanisms, one-point crossovers at bit positions and uniform crossovers at substring boundaries show good online performance and low average generation costs in addition to determining good solutions.

In terms of lowest average generation costs and ultimate online performance, the most effective combination of GA parameters is $N=100$, $p_c=0.5$ and $p_m=0.01$ (third GA parameter set) and the least effective combination is $N=100$, $p_c=1.0$ and $p_m=0.01$ (fourth GA parameter set) with a higher probability of crossover, and this is despite the latter GA parameter set being the most successful in reaching the best solutions.

6.6.2 Recommendations for crossover mechanisms

The less disruptive crossover mechanisms (for example, one-point crossover) are best suited to the genetic algorithm search for the optimisation of the Gessler network expansions problem for crossover points at any bit position linkage in the string. In contrast, the more disruptive crossover mechanisms (uniform crossover) are most effective for crossover points selected only at the linkages between decision-variable substrings. The exploration offered by the one-point crossover and two-point crossover operators at only the substring boundaries is insufficient for this problem. The uniform crossover at substring boundaries (applied to GA runs CROSS61-CROSS65) and one-point crossover at any bit position (GA runs CROSS1-CROSS5) have demonstrated a similar level of performance for the solution space that has been constructed and the other conditions that have applied here.

The results for various crossover mechanisms are influenced by the form of the coded string and the amount of selection pressure applied. The more selection pressure, the more disruptive the crossover mechanism that may be applied. For example, fitness scaling (Section 6.4) could be used to increase the selection pressure. The more disruptive crossover mechanisms may be better suited to longer coded strings. DeJong (1985) suggested that the number of crossover points could be increased as a function of the length of the strings. The relationships between decision-variable substrings in the string are not as easily defined if the system is complex. The findings in this Section with respect to crossover mechanisms may not be as applicable to very complex systems, and further tests would be required.

6 Improvements to the simple GA for pipe network optimisation

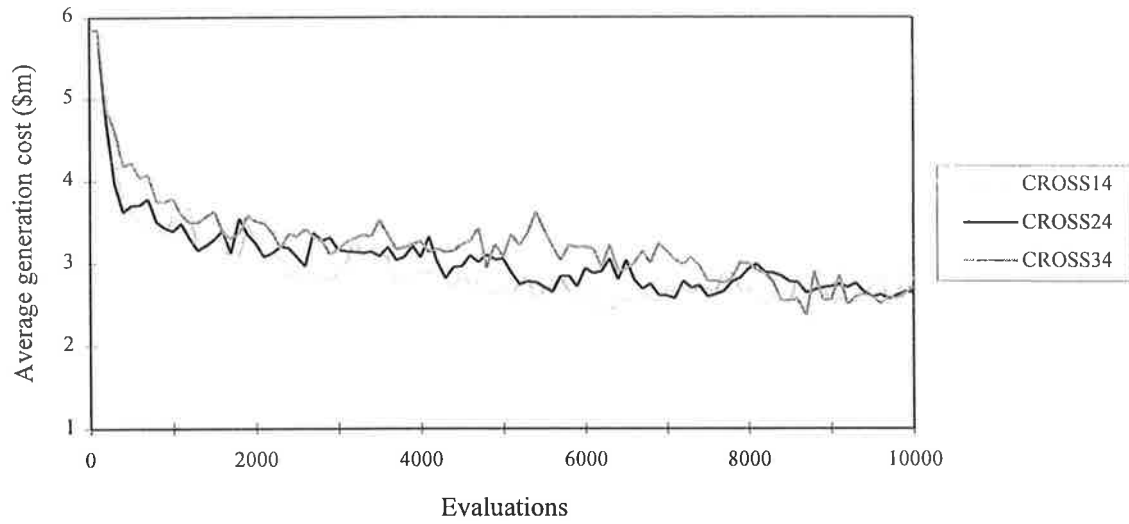


Figure 6.61 Average generation costs for GA runs CROSS14 (two-point crossover), CROSS24 (four-point crossover) and CROSS34 (uniform crossover)

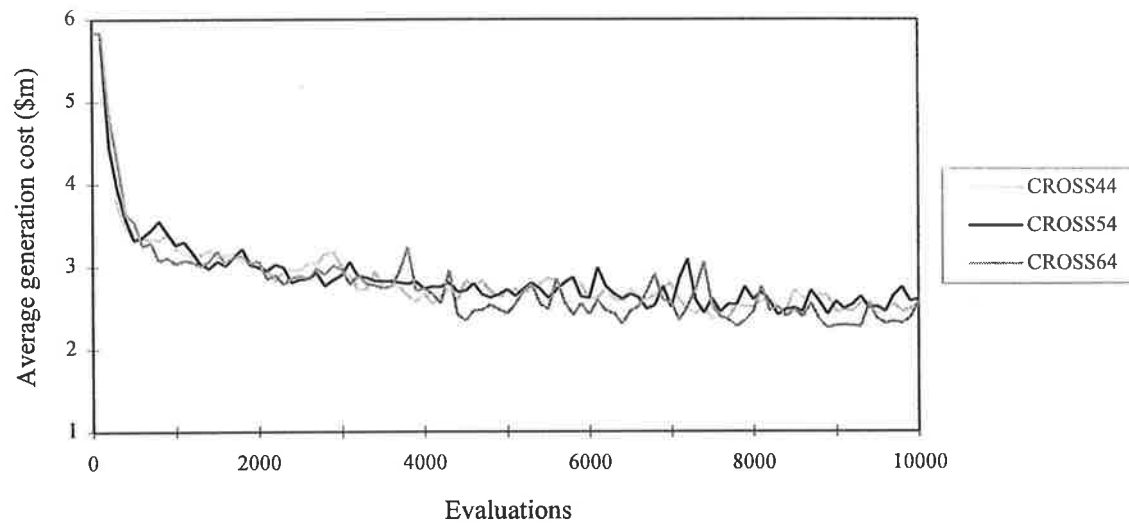


Figure 6.62 Average generation costs for GA runs CROSS44 (one-point crossover at substring boundaries), CROSS54 (two-point crossover at substring boundaries) and CROSS64 (uniform crossover at boundaries)

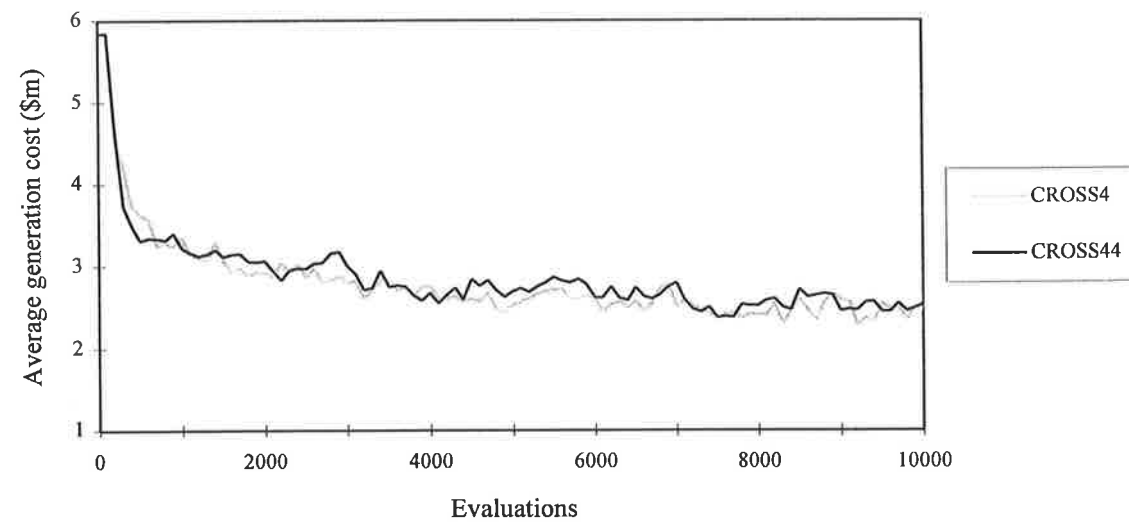


Figure 6.63 Average generation costs for GA runs CROSS4 (one-point crossover) and CROSS44 (one-point crossover at decision-variable substring boundaries)

6 Improvements to the simple GA for pipe network optimisation

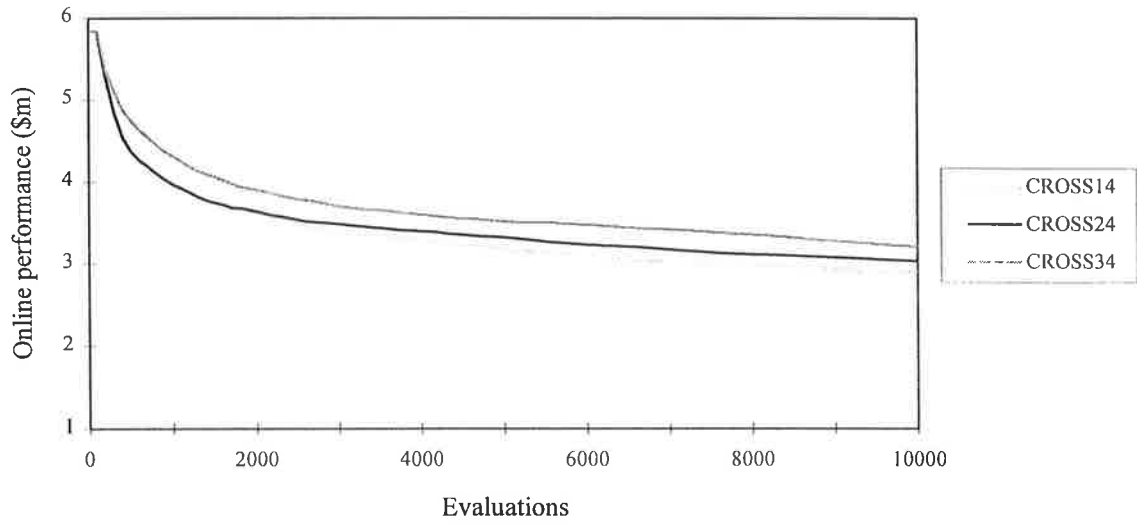


Figure 6.64 Online performance (running average of all solution costs) for GA runs CROSS14 (two-point crossover), CROSS24 (four-point crossover) and CROSS34 (uniform crossover)

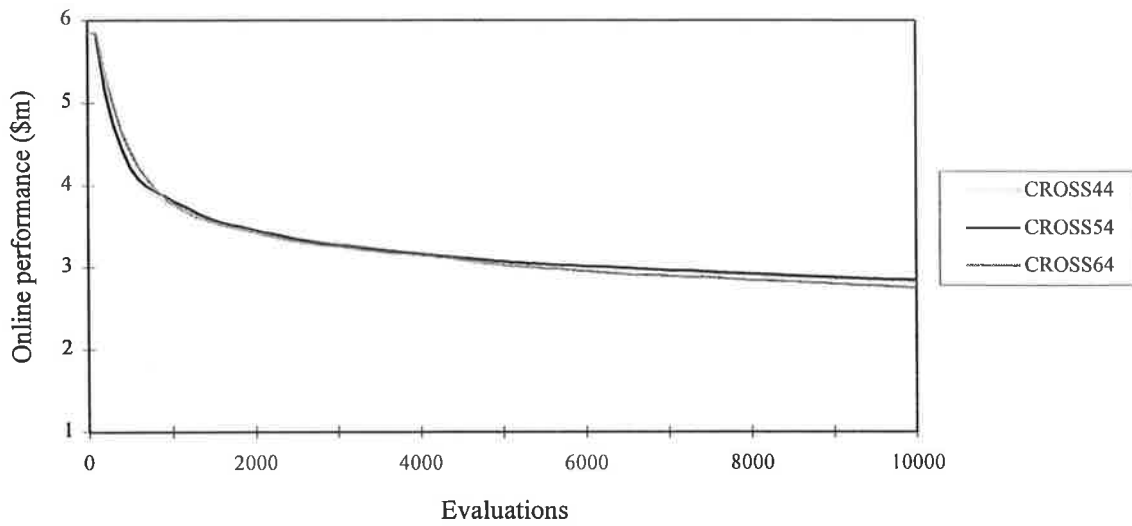


Figure 6.65 Online performance for GA runs CROSS44 (one-point crossover at substring boundaries), CROSS54 (two-point crossover at substring boundaries) and CROSS64 (uniform crossover at substring boundaries)

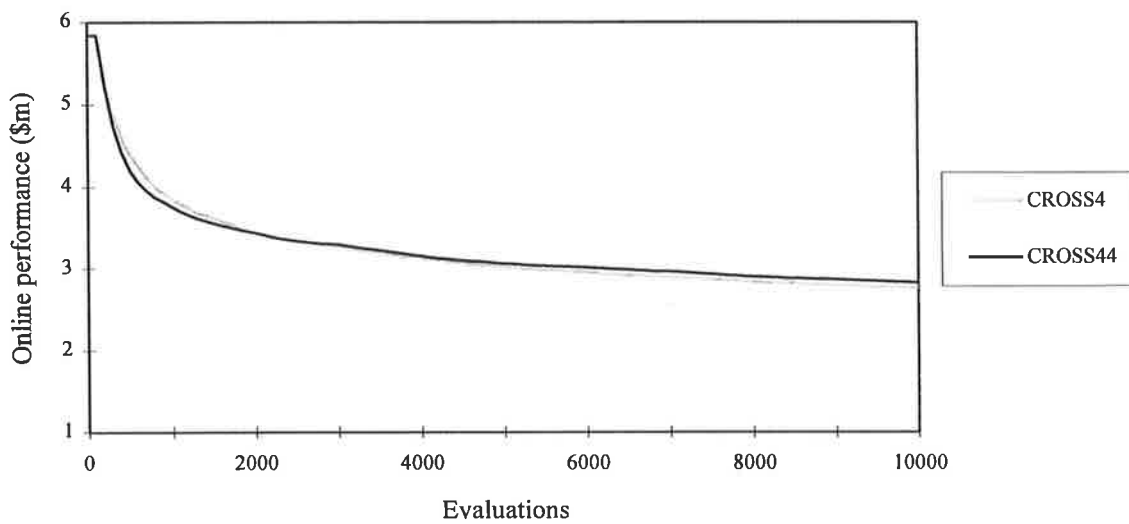


Figure 6.66 Online performance for GA runs CROSS4 (one-point crossover) and CROSS44 (one-point crossover at decision-variable substring boundaries)

6.7 A Creeping Mutation Operator

Mutation is a genetic algorithm mechanism which provides occasional random variations of the genetic code in offspring strings to prevent the loss of potentially useful genetic traits. The traditional bit-wise mutation operator randomly selects bits in offspring strings (generated in crossover) and inverts the bit value (from 1 to 0 or from 0 to 1). A creeping mutation operator is developed here and incorporated within the GA formulation to complement the random bit-wise mutations.

The creeping mutations are also known as adjacency mutations (Dandy, Simpson and Murphy, 1996a), decision-variable-wise mutations or phenotypic mutations (Goldberg, 1990). The creeping mutation mechanism presented in this study is similar to 'creep' described by Davis and Coombs (1987) and Davis (1989).

Davis and Coombs (1987) and Coombs and Davis (1987) studied the application of the GA search to the design of packet switching communication networks. They used coded strings that were lists of link speeds, such that each link speed in the string was coded as a single 'letter' (discussed in Section 6.5). The use of such coding was principally to admit the introduction of an advanced GA operator called 'creep'. Creep altered the speed of a link upward or downward one or multiple steps in the list of allowable link speeds. The process of choosing link speeds of a packet switching communications network is an optimisation problem with similar characteristics to the design of the pipe sizes of the water supply pipe network. Like the pipe network optimisation solution space, the domain searched by Davis and Coombs is very noisy and there is a high degree of parameter interaction. They suggest further consideration be given to creep and representations that support creep in GA approaches to spaces with contiguous optimal parameter values.

Davis (1989) studied a technique for determining effective GA parameter settings based on observed performance as the GA run progresses. He developed a GA consisting of five operators including 'guaranteed-mutation', 'guaranteed-big-creep' and 'guaranteed-little-creep'. An application of an operator on a parent string to produce a child string constituted a reproduction event. The 'guaranteed' feature checks the child string is not identical to the parent string. The 'guaranteed-mutation' operator replaced the value of a coded substring on the parent string, with 10% probability, with a randomly selected value within the list of allowable values. The 'guaranteed-big-creep' operator replaced the value of a coded substring on the parent string, with 20% probability, with a value that is 1, 2 or 3 units above or below the original value. The magnitude and direction of the big-creep is selected randomly. The 'guaranteed-little-creep' operated with 10% probability and the modified value of the coded substring on the child string was only 1 unit above or below the original value. The direction

of the little-creep movement was selected randomly. The creeping should not extend past the limits of the list of allowable values.

Goldberg (1990) recognised the potential of a decision-variable-wise mutation operator in his discussion of real-coded GAs. Goldberg suggested the use of both bit-wise (traditional mutation operator) and decision-variable-wise mutation or phenotypic mutations in multi-parameter binary-coded GAs to overcome such problems as Hamming Cliffs.

6.7.1 A creeping mutation operator for binary-coded substrings

The creeping mutation operator presented in this research is applied to selected binary-coded decision-variable substrings of offspring coded strings which have been generated for the new population by crossover. The chosen coded substring is mutated to a neighbouring substring representing an adjacent design parameter (pipe size), either up or down the list of design parameters (and within the limits of allowable parameter values).

Candidate substrings of offspring strings to be subjected to creep are selected randomly with some specified probability of creeping mutation, p_a . For example, the GA model runs designated CREEP21-CREEP25 in the following experimental analysis use a value of $p_a=0.125$, which implies about 1 in 8 decision-variable substrings are disrupted by creep in offspring strings formed by crossover. Since the coded strings symbolising Gessler network designs consist of 8 decision-variable substrings for the 8 pipe sizes to be selected, we would expect one substring to undergo a creeping mutation for every new string created by crossover.

The creeping mutation operator allows for the adjustment of the chances of creeping up or down the list of design parameters. The creeping mutation will generate a modified substring representing an adjacent design parameter down the parameter list (decreasing pipe size) with some specified probability of creeping down, p_d . For example, a value of $p_d = 0.6$, implies there is a 60% chance of creeping from the current pipe size to the next smaller pipe size rather than the next larger pipe size. The probability of creeping up the list of allowable pipe sizes is $p_u=1-p_d$. The direction of the creep may be unbiased ($p_d=0.5$) or biased in either direction. The chosen decision-variable substring remains intact if the substring is at the upper or lower limit of allowable substrings and the creep would like to adjust the substring beyond this limit.

Figure 6.67 demonstrates the very simple action of a creeping mutation. The 24-bit coded string in Figure 6.67 is constructed of 8 concatenated 3-bit substrings. The string symbolises a solution to the Gessler network expansions problem. The decision-variable substrings of binary codes represent new pipe sizes. The last substring position is associated with the new pipe [14]. The binary number 011 in this position decodes to a 12" pipe diameter for pipe [14]

(by observing the mapping in Table 6.23). If the last substring was chosen to be subjected to creep, it may be mutated down to 010 (10" pipe) or up to 100 (14" pipe) to form one of the modified strings as shown:

Old coded string:	011-110-000-000-001-111-010- 011	
New coded strings:	011-110-000-000-001-111-010- 010	(creeping down)
	011-110-000-000-001-111-010- 100	(creeping up)

Figure 6.67 The action of a creeping mutation

The creeping mutations contrast with the traditional bit-wise mutations that may or may not produce an adjacent substring. Bit-wise mutations provide useful variations, however, alone they are not likely to overcome the Hamming Cliffs of binary code, in which substrings such as 011 and 100 represent adjacent design parameters. Bit-wise mutations with low probability and creeping mutations were used simultaneously in the following GA model runs. Like bit-wise mutations, it becomes clear in this analysis that creeping mutations are most effective when used with low probability.

In addition to possessing some reliability that accounts for global exploration of the solution space, an effective optimisation technique must exercise refinement in the vicinity of relatively good solutions. The subtle creeping mutations (with low probability) may provide the fine adjustments that are necessary. The purpose of the creeping mutations is to promote hill-climbing and local exploration. We expect the creeping mutations to be most prominent once the selection method has identified relatively good regions of the solution space.

6.7.2 GA model runs to measure the effectiveness of creeping mutations

A series of GA model runs were conducted which experimented with various combinations of the probability of creep, p_a and the probability of creeping down, p_d . The decision-variable substrings were coded in binary codes. The GA runs CREEP1-CREEP5 are the traditional GA runs using $p_a=0.0$ (no creep). The GA runs CREEP11-CREEP15 ($p_a=0.0625$ and $p_d=0.5$), CREEP21-CREEP25 ($p_a=0.125$ and $p_d=0.5$) and CREEP31-CREEP35 ($p_a=0.25$ and $p_d=0.5$) use increasing probabilities of creeping mutation for an equal chance of creeping up or down. The GA runs CREEP41-CREEP45 ($p_a=0.125$ and $p_d=0.25$) consider a high probability of creeping up and GA runs CREEP51-CREEP55 ($p_a=0.125$ and $p_d=0.75$) consider the effect of a high probability of creeping down. The performance of the GA model runs CREEP1-CREEP55 is summarised in Tables 6.37-6.42.

Table 6.37 Search results for genetic algorithm model runs CREEP1-CREEP5

No creeping mutations ($p_a=0.0$). GA runs CREEP1-CREEP5 equivalent to GA runs PEN11-PEN15.					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	CREEP1 $N=50$, $p_c=0.75$	CREEP2 $p_c=0.75$	CREEP3 $p_c=0.5$	CREEP4	CREEP5 $p_m=0.005$
Number of generations required	266	133	197	100	100
Lowest solution cost (\$m) (after - generations)	1.7503 [‡] (59)	1.7503 [‡] (96)	1.7503 [‡] (59)	1.7725 (93)	1.8807 (82)
Lowest cost GA design generated (Table 5.4)	2	2	1	3	-
Lowest average generation cost (\$m) (after - generations)	2.045 (212)	2.139 (73)	2.038 (176)	2.271 (92)	2.274 (86)
Ultimate offline performance (\$m)	1.963	1.896	1.916	1.935	2.026
Ultimate online performance (\$m)	2.560	2.557	2.475	2.766	2.694

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.38 Search results for GA model runs CREEP11-CREEP15

Probability of creeping mutation $p_a=0.0625$ with probability of creeping down $p_d=0.5$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	CREEP11 $N=50$, $p_c=0.75$	CREEP12 $p_c=0.75$	CREEP13 $p_c=0.5$	CREEP14	CREEP15 $p_m=0.005$
Number of generations required	261	131	195	100	100
Lowest solution cost (\$m) (after - generations)	1.8506 (74)	1.8232 (48)	1.7503 [‡] (177)	1.7503 [‡] (51)	1.7503 [‡] (99)
Lowest cost GA design generated (Table 5.4)	44	16	1	1	1
Lowest average generation cost (\$m) (after - generations)	2.441 (92)	2.452 (63)	2.236 (79)	2.625 (53)	2.547 (37)
Ultimate offline performance (\$m)	2.137	2.003	1.897	2.012	2.032
Ultimate online performance (\$m)	2.930	2.912	2.583	3.092	2.931
Total no. of strings affected by creep	3,984	3,997	3,915	3,928	3,928
Total no. of <i>good</i> creeps	1,597	1,539	1,340	1,536	1,559
Total no. of <i>bad</i> creeps	2,059	2,148	2,234	2,089	2,109
Average % cost decrease due to <i>good</i> creeps	4.28%	4.30%	4.45%	4.86%	4.10%
Average % cost increase due to <i>bad</i> creeps	11.69%	11.75%	15.44%	11.94%	11.75%

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

Table 6.39 Search results for GA model runs CREEP21-CREEP25

Probability of creeping mutation $p_a=0.125$ with probability of creeping down $p_d=0.5$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	CREEP21 $N=50$, $p_c=0.75$	CREEP22 $p_c=0.75$	CREEP23 $p_c=0.5$	CREEP24	CREEP25 $p_m=0.005$
Number of generations required	266	133	198	100	100
Lowest solution cost (\$m) (after - generations)	1.8115 (207)	1.8383 [†] (104)	1.8010 (157)	1.8417 (96)	1.8232 (89)
Lowest cost GA design generated (Table 5.4)	13	-	7	37	18
Lowest average generation cost (\$m) (after - generations)	2.328 (208)	2.674 (123)	2.376 (191)	2.888 (91)	2.785 (81)
Ultimate offline performance (\$m)	2.070	2.022	2.023	2.130	2.138
Ultimate online performance (\$m)	2.931	3.006	2.779	3.316	3.209
Total no. of strings affected by creep	6,536	6,518	6,581	6,487	6,487
Total no. of <i>good</i> creeps	2,525	2,581	2,473	2,666	2,688
Total no. of <i>bad</i> creeps	3,601	3,507	3,726	3,453	3,380
Average % cost decrease due to <i>good</i> creeps	4.85%	5.0%	4.69%	4.96%	4.73%
Average % cost increase due to <i>bad</i> creeps	15.25%	15.43%	12.44%	12.91%	11.86%

[†] Infeasible design. The solution cost includes the penalty cost. GA run CREEP22 determined a best cost feasible solution for \$1.8417 million (Solution 36 in Table 5.4) after 44 generations.

The optimum solution is achieved in all the GA runs CREEP51-CREEP55 with the high probability of creeping down (Table 6.42). About 7 or 8 from 10 creeping mutations will be down the design parameter list using $p_d=0.75$. The success of the bias towards creeping down is perhaps because the objective function is to be minimised. We do not disregard the possibility of creeping up since we expect to approach the best regions of the solution space from both the feasible and infeasible regions.

Figures 6.68-6.79 show some relevant comparisons of the variations of best generation costs, offline performance, average generation costs and online performance for the GA model runs CREEP4, CREEP14, . . . , CREEP54.

Table 6.40 Search results for GA model runs CREEP31-CREEP35

Probability of creeping mutation $p_a=0.25$ with probability of creeping down $p_d=0.5$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	CREEP31 $N=50$, $p_c=0.75$	CREEP32 $p_c=0.75$	CREEP33 $p_c=0.5$	CREEP34	CREEP35 $p_m=0.005$
Number of generations required	268	133	200	100	100
Lowest solution cost (\$m) (after - generations)	1.8807 (120)	1.8383 [†] (70)	1.7503 [‡] (82)	1.8010 (48)	1.8417 (17)
Lowest cost GA design generated (Table 5.4)	-	-	2	7	35
Lowest average generation cost (\$m) (after - generations)	2.674 (119)	2.820 (47)	2.522 (110)	3.048 (51)	2.957 (94)
Ultimate offline performance (\$m)	2.160	2.074	1.967	2.159	2.182
Ultimate online performance (\$m)	3.181	3.195	3.015	3.414	3.343
Total no. of strings affected by creep	9,001	8,937	8,892	8,963	8,963
Total no. of <i>good</i> creeps	3,664	3,627	3,293	3,830	3,937
Total no. of <i>bad</i> creeps	5,016	4,933	5,161	4,812	4,704
Average % cost decrease due to <i>good</i> creeps	6.20%	5.95%	6.1%	6.12%	5.55%
Average % cost increase due to <i>bad</i> creeps	15.95%	16.8%	19.9%	15.34%	13.3%

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

[†] Infeasible design. The solution cost includes the penalty cost. GA run CREEP32 determined a best cost feasible solution for \$1.8417 million (Solution 34 in Table 5.4) after 71 generations.

The plot of best of generation costs in Figure 6.70 demonstrates the success of the GA run CREEP54. The poor online performance of creeping down more often in GA run CREEP54 may be attributed to stepping inside the infeasible region more often (Figure 6.79).

The plot of average generation costs in Figures 6.74-6.76 show that using creep will generally lead to higher average generation costs, due to the disruptions caused by the increased exploration by the creeping mutations.

6.7.3 Recommendations for creeping mutations

The results of these experiments indicate the use of low probabilities of creeping mutation with an increased chance of creeping down are appropriate. Further testing of these parameters with more complicated pipe network optimisation problems is required. We have not investigated the combination of creeping mutations and substrings of Gray codes. Creeping mutations are actually particular bit-wise mutations when creeping mutations are applied to a list of substrings of Gray codes.

Table 6.41 Search results for GA model runs CREEP41-CREEP45

Probability of creeping mutation $p_a=0.125$ with probability of creeping down $p_d=0.25$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	CREEP41 $N=50$, $p_c=0.75$	CREEP42 $p_c=0.75$	CREEP43 $p_c=0.5$	CREEP44	CREEP45 $p_m=0.005$
Number of generations required	266	133	198	100	100
Lowest solution cost (\$m) (after - generations)	1.8612 (260)	1.8807 (67)	1.8300 (145)	1.9114 (66)	1.9251 (70)
Lowest cost GA design generated (Table 5.4)	>50	>50	21	>50	>50
Lowest average generation cost (\$m) (after - generations)	2.456 (259)	2.620 (113)	2.396 (125)	2.994 (74)	2.927 (63)
Ultimate offline performance (\$m)	2.236	2.148	2.061	2.289	2.257
Ultimate online performance (\$m)	3.034	3.026	2.839	3.288	3.256
Total no. of strings affected by creep	6,536	6,518	6,581	6,487	6,487
Total no. of <i>good</i> creeps	1,841	1,739	1,698	1,814	1,745
Total no. of <i>bad</i> creeps	4,198	4,162	4,353	4,104	4,141
Average % cost decrease due to <i>good</i> creeps	5.4%	5.07%	6.46%	5.23%	4.74%
Average % cost increase due to <i>bad</i> creeps	5.96%	6.31%	7.26%	5.28%	5.25%

Further study is recommended to observe the relationship between penalty functions and the preferred direction of creep (up or down). There is likely to be some relationship between the bias in the direction of creep and the severity of the penalties. It may be appropriate to tend to creep up when penalties are light so as to approach the best regions of the solution space from the infeasible region (below) and similarly, it may be preferred to creep down when penalties are strict so as to approach the optimum from the feasible region (above).

Finally, there may be better ways to determine the best direction to creep. For example, the chances of creeping down may be increased if the current population of solutions is typically feasible or if the coded string in which the chosen substring belongs is currently feasible or more specifically, according to the hydraulic performance (for example, based on headloss per unit length) of the pipe size represented by the substring in the network solution.

Table 6.42 Search results for GA model runs CREEP51-CREEP55

Probability of creeping mutation $p_a=0.125$ with probability of creeping down $p_d=0.75$					
GA RUNS Unless specified otherwise GA parameters $N=100$, $p_c=1.0$ and $p_m=0.01$	CREEP51 $N=50$, $p_c=0.75$	CREEP52 $p_c=0.75$	CREEP53 $p_c=0.5$	CREEP54	CREEP55 $p_m=0.005$
Number of generations required	266	133	198	100	100
Lowest solution cost (\$m) (after - generations)	1.7503 [‡] (186)	1.7503 [‡] (93)	1.7503 [‡] (59)	1.7503 [‡] (58)	1.7503 [‡] (74)
Lowest cost GA design generated (Table 5.4)	1	2	2	1	1
Lowest average generation cost (\$m) (after - generations)	2.389 (206)	2.680 (109)	2.314 (155)	2.763 (53)	2.864 (94)
Ultimate offline performance (\$m)	1.980	1.980	1.901	1.987	1.992
Ultimate online performance (\$m)	3.136	3.135	2.816	3.420	3.354
Total no. of strings affected by creep	6,536	6,518	6,581	6,487	6,487
Total no. of <i>good</i> creeps	2,596	2,931	2,523	2,909	2,697
Total no. of <i>bad</i> creeps	3,464	3,154	3,396	3,117	3,322
Average % cost decrease due to <i>good</i> creeps	4.7%	4.28%	4.55%	4.8%	4.93%
Average % cost increase due to <i>bad</i> creeps	34.1%	25.9%	36.17%	30.3%	31.7%

[‡] Global optimum solution (verified by complete enumeration in Chapter 5)

6 Improvements to the simple GA for pipe network optimisation

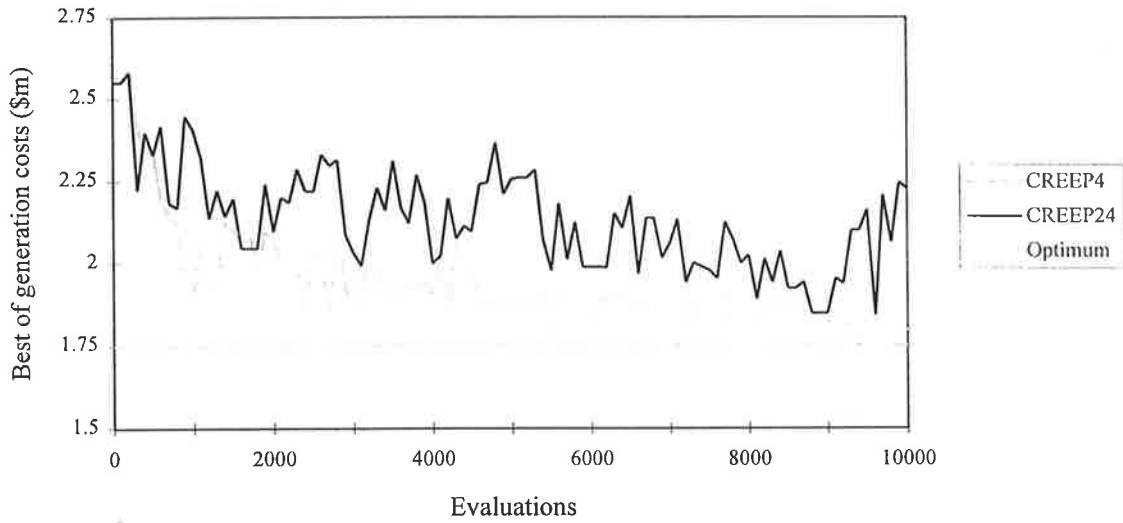


Figure 6.68 Best generation costs for GA runs CREEP4 (No creeping mutation, $p_a=0.0$) and CREEP24 (probability of creep, $p_a=0.125$, and probability of creeping down, $p_d=0.5$)

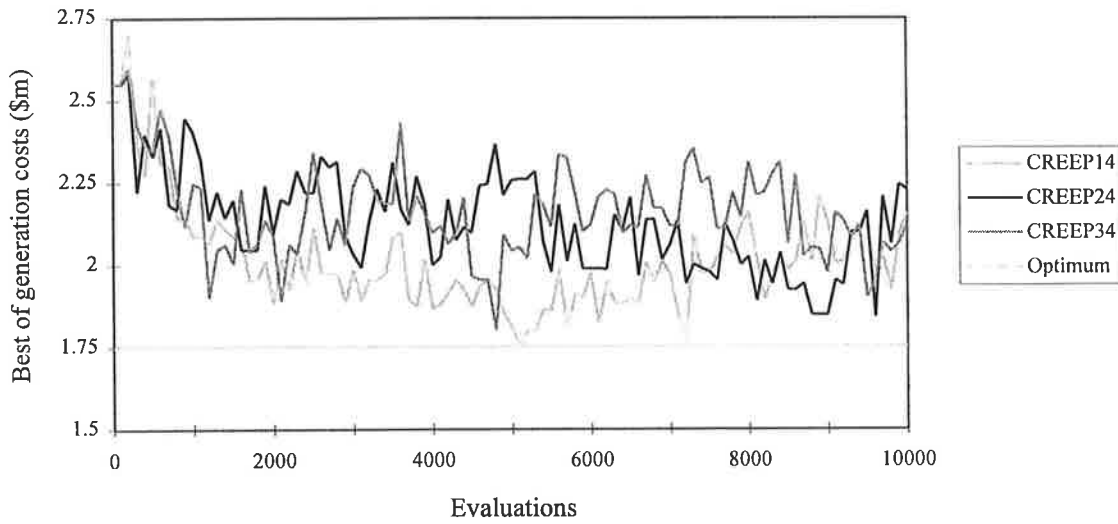


Figure 6.69 Best generation costs for GA runs CREEP14 ($p_a=0.0625$, $p_d=0.5$), CREEP24 ($p_a=0.125$, $p_d=0.5$) and CREEP34 ($p_a=0.25$, $p_d=0.5$)

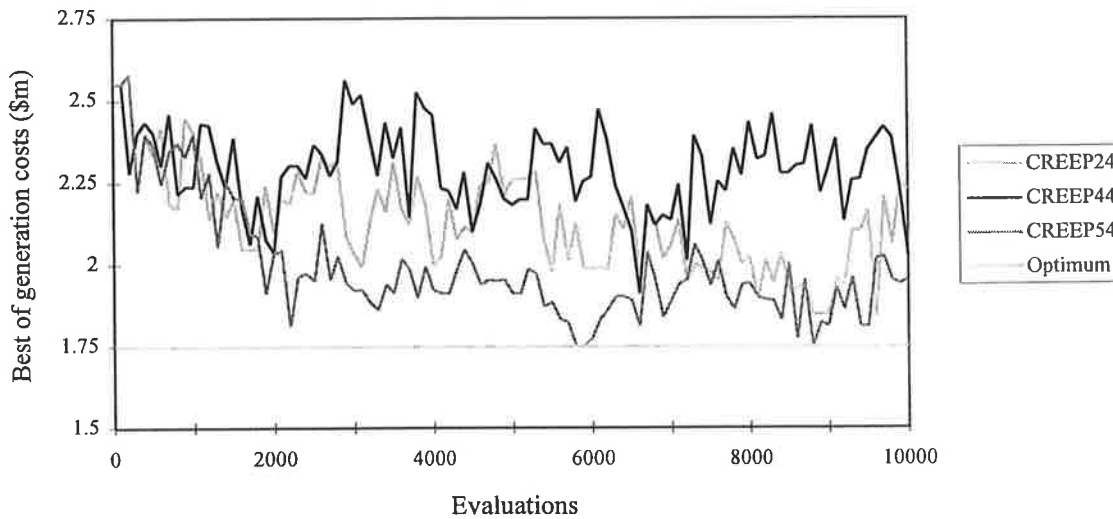


Figure 6.70 Best generation costs for GA runs CREEP24 ($p_a=0.125$, $p_d=0.5$), CREEP44 ($p_a=0.125$, $p_d=0.25$) and CREEP54 ($p_a=0.125$, $p_d=0.75$)

6 Improvements to the simple GA for pipe network optimisation

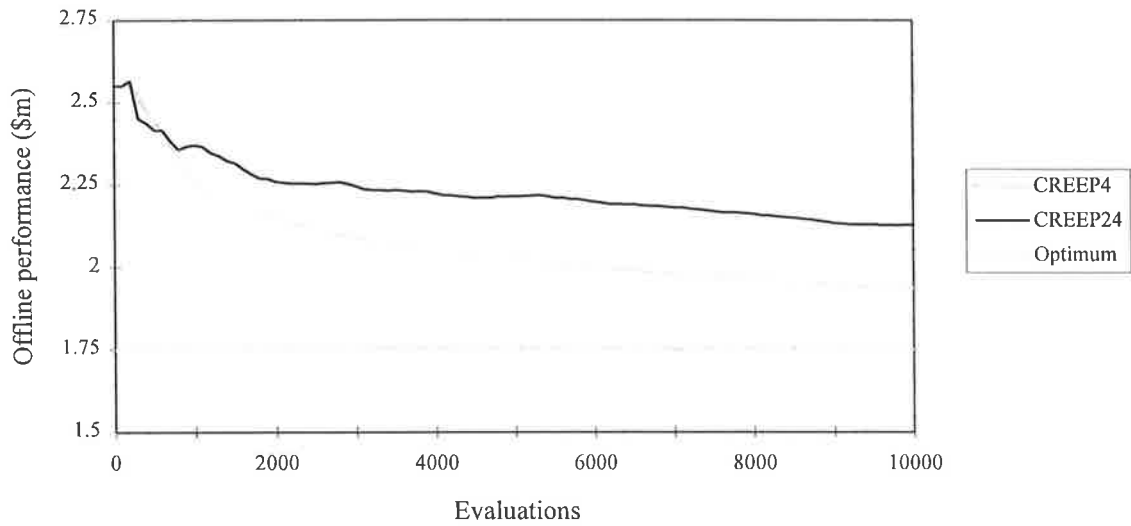


Figure 6.71 Offline performance (running average of best cost solutions) for GA runs CREEP4 (No creeping mutation, $p_a=0.0$) and CREEP24 (probability of creep, $p_a=0.125$, and probability of creeping down, $p_d=0.5$)

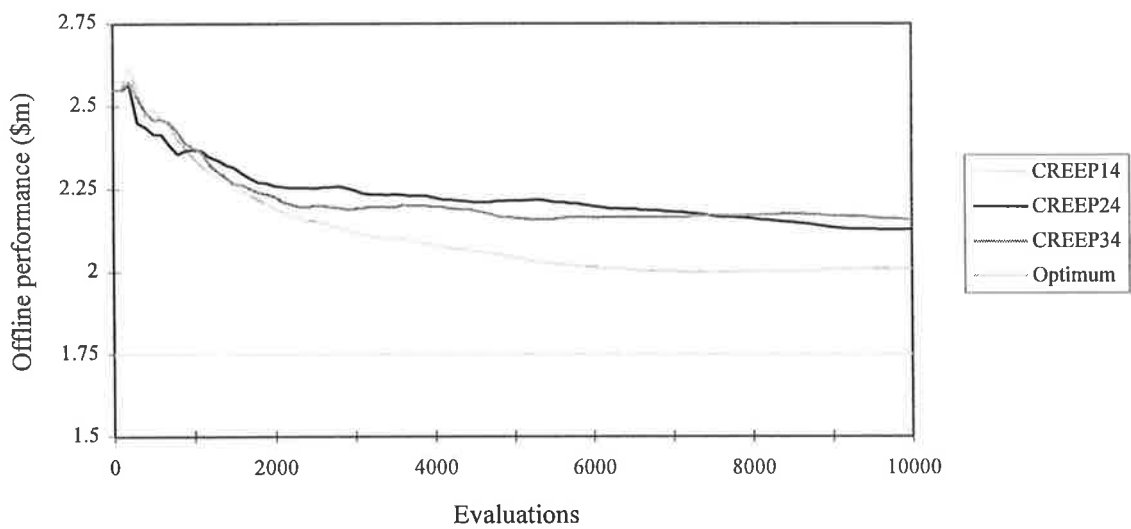


Figure 6.72 Offline performance for GA runs CREEP14 ($p_a=0.0625$, $p_d=0.5$), CREEP24 ($p_a=0.125$, $p_d=0.5$) and CREEP34 ($p_a=0.25$, $p_d=0.5$)

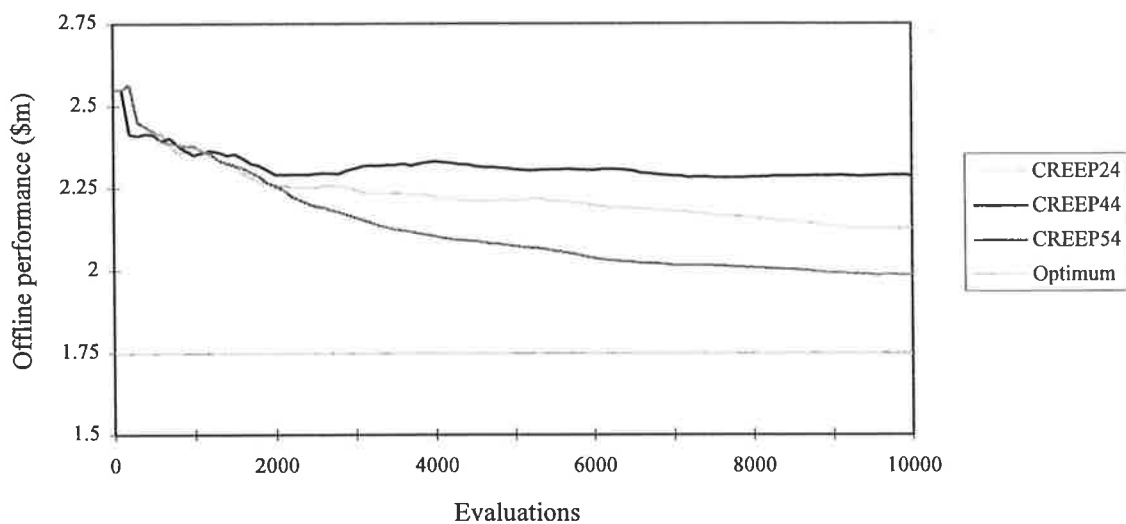


Figure 6.73 Offline performance for GA runs CREEP24 ($p_a=0.125$, $p_d=0.5$), CREEP44 ($p_a=0.125$, $p_d=0.25$) and CREEP54 ($p_a=0.125$, $p_d=0.75$)

6 Improvements to the simple GA for pipe network optimisation

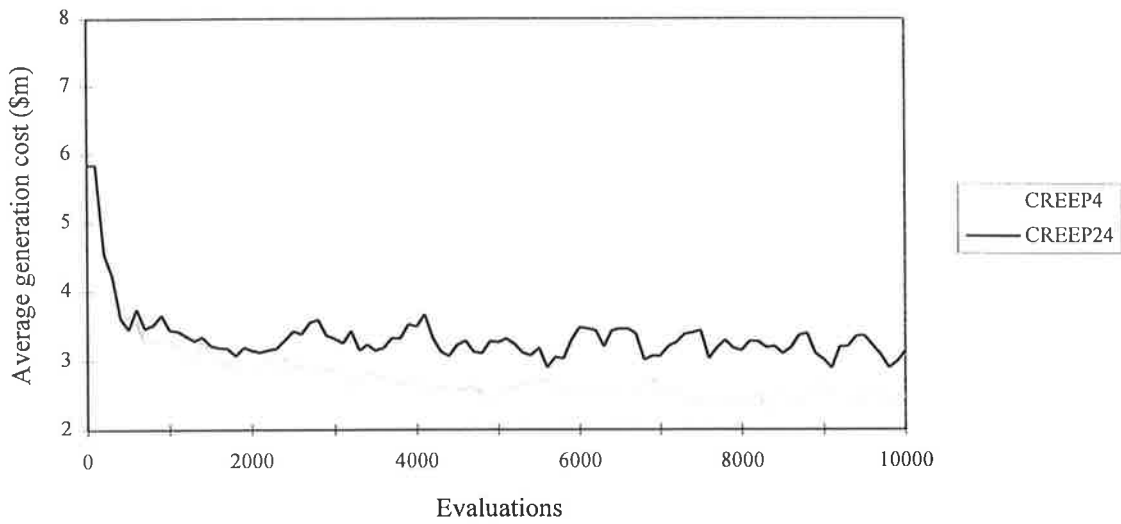


Figure 6.74 Average generation costs for GA runs CREEP4 (No creeping mutation, $p_a=0.0$) and CREEP24 (probability of creep, $p_a=0.125$, and probability of creeping down, $p_d=0.5$)

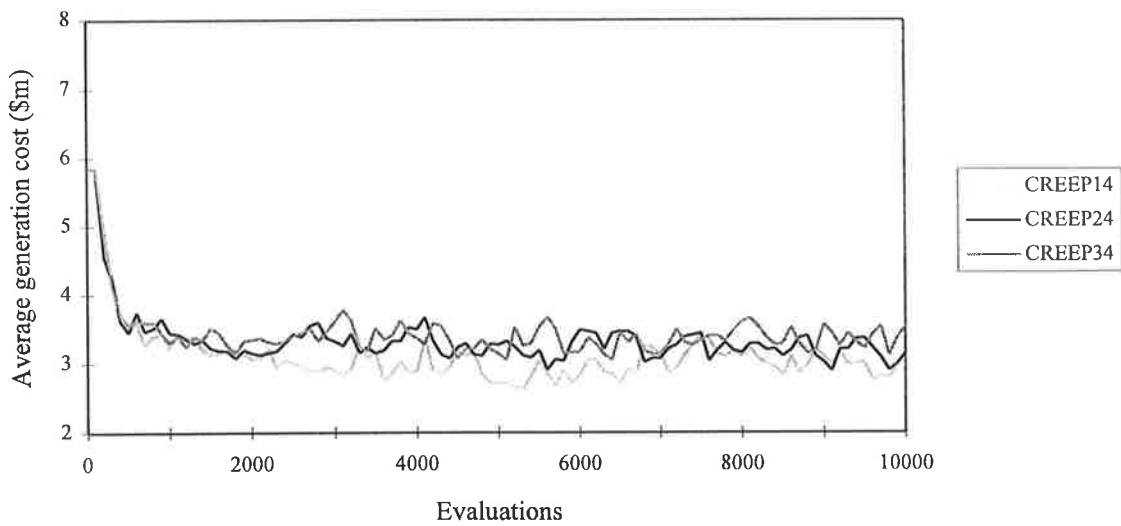


Figure 6.75 Average generation costs for GA runs CREEP14 ($p_a=0.0625$, $p_d=0.5$), CREEP24 ($p_a=0.125$, $p_d=0.5$) and CREEP34 ($p_a=0.25$, $p_d=0.5$)

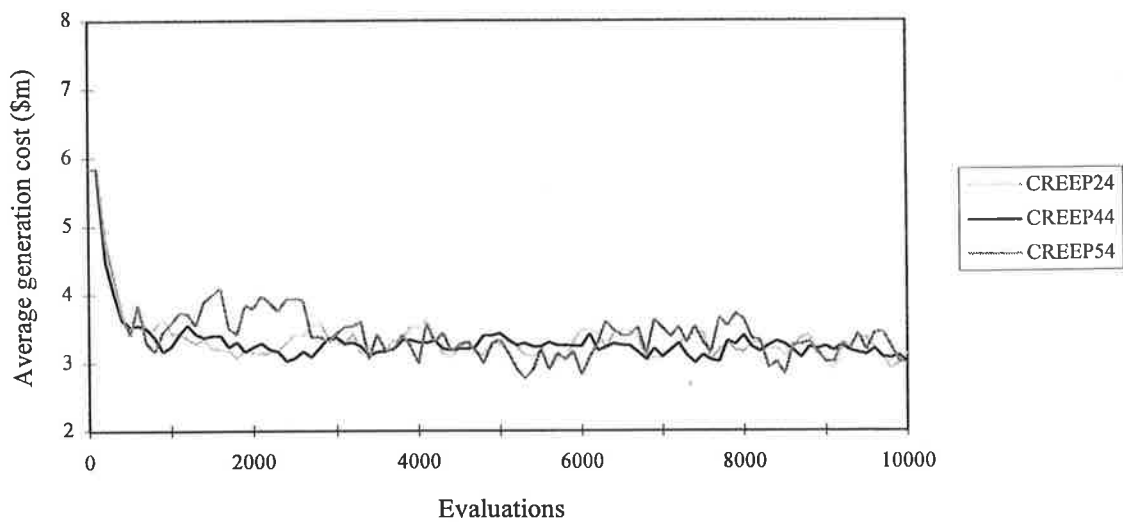


Figure 6.76 Average generation costs for GA runs CREEP24 ($p_a=0.125$, $p_d=0.5$), CREEP44 ($p_a=0.125$, $p_d=0.25$) and CREEP54 ($p_a=0.125$, $p_d=0.75$)

6 Improvements to the simple GA for pipe network optimisation

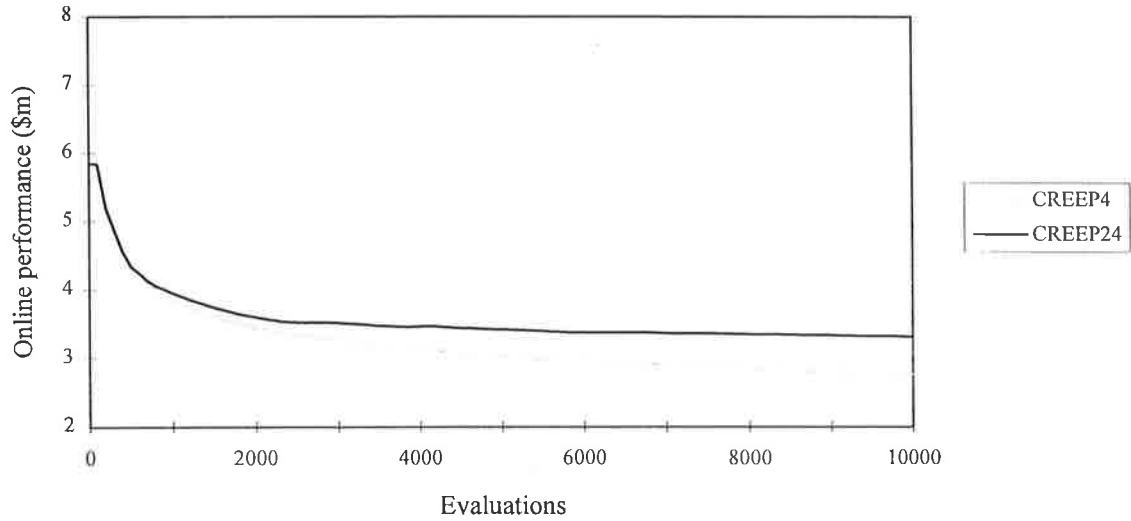


Figure 6.77 Online performance (running average of all solution costs) for GA runs CREEP4 (No creeping mutation, $p_a=0.0$) and CREEP24 (probability of creep, $p_a=0.125$, and probability of creeping down, $p_d=0.5$)

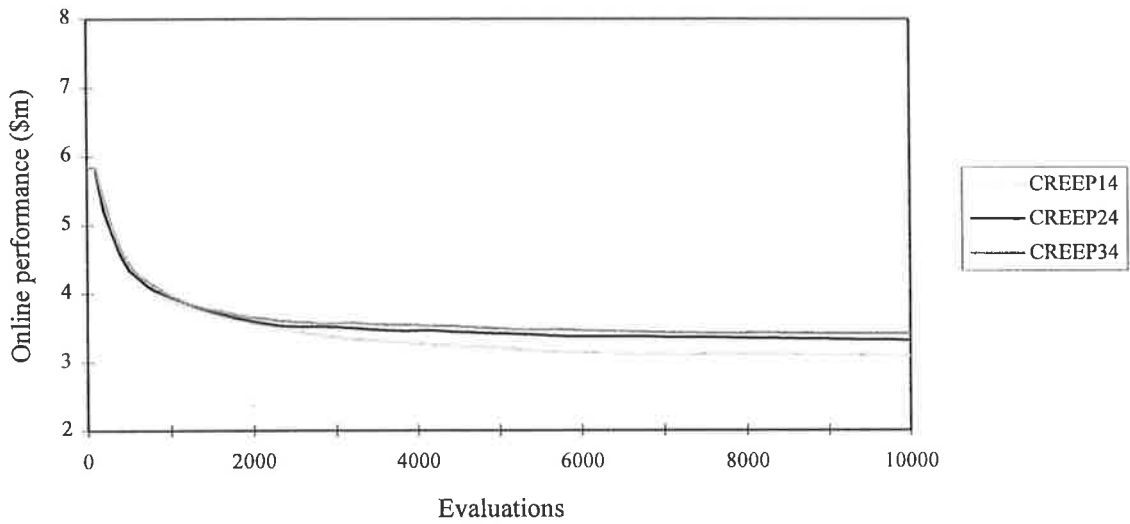


Figure 6.78 Online performance for GA runs CREEP14 ($p_a=0.0625$, $p_d=0.5$), CREEP24 ($p_a=0.125$, $p_d=0.5$) and CREEP34 ($p_a=0.25$, $p_d=0.5$)

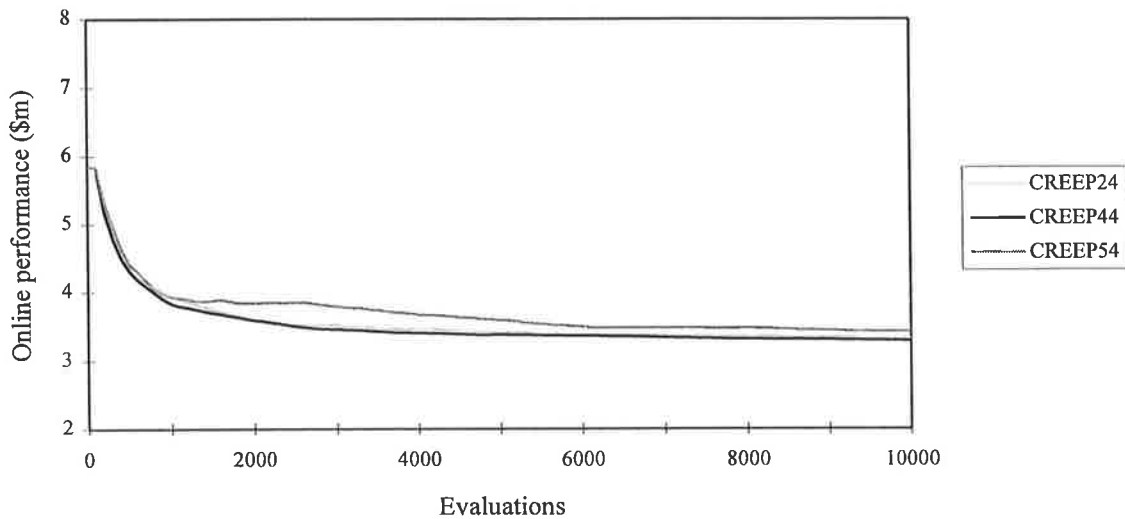


Figure 6.79 Online performance for GA runs CREEP24 ($p_a=0.125$, $p_d=0.5$), CREEP44 ($p_a=0.125$, $p_d=0.25$) and CREEP54 ($p_a=0.125$, $p_d=0.75$)

6.8 Conclusions

The experimental analyses in this chapter has provided an improved understanding of some of the issues facing the GA user when selecting a genetic algorithm formulation for pipe network optimisation.

A penalty function which computes penalty costs based on the maximum violations of the pressure constraints for each loading condition was found to be the most suitable. An appropriate penalty multiplier ($k=\$50,000/\text{psi}$) was established for the Gessler problem, although this parameter is problem-dependent. There may be value in varying the penalty multiplier as the GA run proceeds.

A form of power law fitness scaling has been used. The fitness scaling exponent n is increased in steps during the GA model run, adjusting raw fitness values in order to maintain appropriate levels of competition at different stages of the GA run. The use of power scaled fitnesses with a low value of the exponent n early in the GA run allows the population the freedom of a wider exploration of the solution space. The scaled fitnesses with higher values of n later on in the GA run forces the population to focus on the best regions of the solution space. The tournament selection method was found to be very effective and very efficient, however, larger population sizes are recommended.

A number of past theoretical and empirical results suggest a Gray code representation may be effective in eliminating biases introduced by the binary code. There have been indications in the analysis conducted in this chapter, that the Gray code may be more appropriate for the pipe network optimisation problem, however binary codes have also performed effectively. There is clear evidence that the arrangement of decision variable substrings within the coded string is a significant issue to be confronted by the GA user for the application to pipe network optimisation. Decision variable substrings which are related in the hydraulics and economics of the solution should be positioned nearby in the string. The performance of an integer coding scheme was inferior to the coding schemes based on the binary alphabet when applied to an identical solution space, however redundant binary codes have increased the size of the solution space from 4 million possible solutions to 16 million solutions. Redundant codes can be eliminated using an integer coding scheme. In addition, the performance of the integer coding scheme may be improved if it is used in association with suitable operators.

The traditional GA one-point crossover operator provided a sufficient amount of exploration for the length of coded string constructed for the Gessler problem. More crossover points may be required as the length of the coded string increases. The more disruptive crossover operators

6 Improvements to the simple GA for pipe network optimisation

such as uniform crossover were more effective if the crossover points were selected only at the boundaries of the decision variable substrings.

The creeping mutation operator was found to be a very effective tool for exploring the immediate vicinity of the current population in the solution space.

The modifications to the traditional GA model have been considered systematically and separately in the application of the Gessler pipe network optimisation problem in this chapter. In the next chapter, an improved genetic algorithm is formulated based on the findings in this chapter. The improved GA combines elements of the GA model formulation such as the increased exploitation offered by fitness scaling and the subtle exploration powers of creeping mutations. The improved GA model is applied to pipe network optimisation problems of increased size and complexity in the following chapters.

7 Larger Problems with Known Optimal Solutions

7.1 The Original Gessler Problem

The two-reservoir Gessler (1985) network expansion problem introduced in Chapter 5 is a relatively simple pipe network optimisation. The Gessler problem was chosen as a trial optimisation problem for the genetic algorithm (GA) application as it is feasible to enumerate every possible pipe network solution in order to positively identify the global optimum solution(s). As a pipe network optimisation problem increases in size, it soon becomes impossible to perform an exhaustive enumeration.

The set of possible network solutions to the Gessler problem may be represented by the set of coded strings of 24 binary bits. The 24-bit strings are formed by eight 3-bit substrings. The eight substring positions in the coded string correspond to the eight pipes to be sized. The unique structure of a 3-bit coded substring represents the selected pipe size (from the list of eight possible new or equivalent pipe sizes) for the pipe associated with the substring.

The exhaustive enumeration described in Section 5.2.2 identified two optimal solutions, each having a cost of \$1.7503 million, given by solutions 1 and 2 in Table 5.4. The corresponding optimal 24-bit coded strings may be represented using Gray codes as shown below. The representations of the alternative optimal coded strings differ in only two bit positions (i.e., a Hamming distance of 2). The solution space searched by the GA consists of $2^{24} = 16,777,216$ possible network solutions.

Solution 1	000-101-000-010-001-001-000-011
	leave-dup.14"-leave-12"-8"-8"-6"-10"
Solution 2	000-101-000-010-001-011-000-001
	leave-dup.14"-leave-12"-8"-10"-6"-8"

7.2 Simultaneous Optimisation of Two Gessler Problems

In engineering practice, the GA will be required to optimise much larger pipe network designs. An expanded solution space with known global optimal solutions is considered by simultaneously optimising two independent 14-pipe Gessler networks as an equivalent 28-pipe network. The GA searches for the optimal 48-bit coded string formed by two adjacent 24-bit coded strings as shown in Figure 7.1.

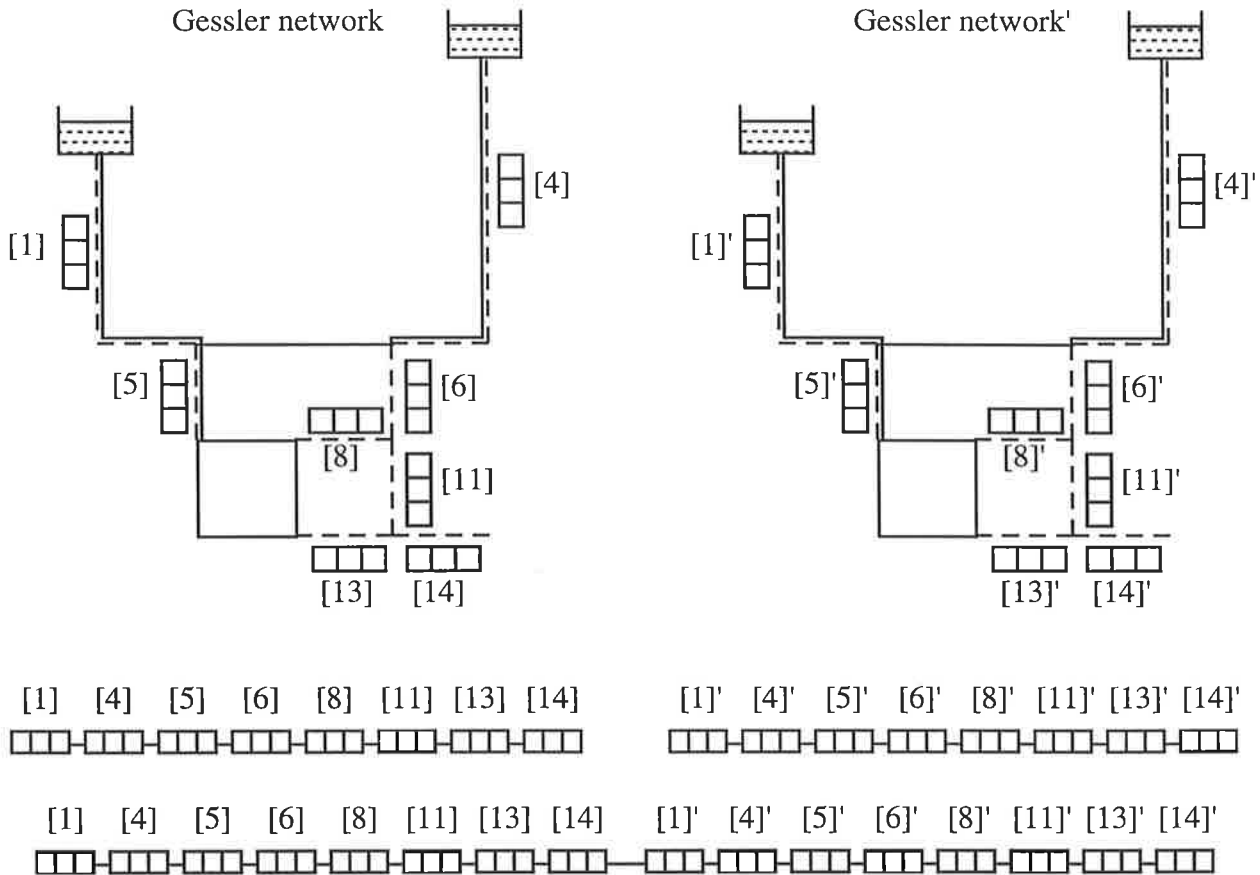


Figure 7.1 Formation of a 48-bit string (from two 24-bit strings) representing trial solutions in the GA search for two independent Gessler network designs

The cost of the global optimal solutions for the equivalent 28-pipe network expansion problem is \$3.5006 million. The four possible combinations of the optimal solutions 1 and 2 to the original Gessler problem form four optimal solutions to this larger problem. The four optimal 48-bit coded string solutions may be represented using Gray codes as shown below. The expanded solution space searched by the GA consists of $2^{48} = 2.815 \times 10^{14}$ possible network solutions.

- Solution 1,1 000-101-000-010-001-001-000-011--000-101-000-010-001-001-000-011
- Solution 1,2 000-101-000-010-001-001-000-011--000-101-000-010-001-011-000-001
- Solution 2,1 000-101-000-010-001-011-000-001--000-101-000-010-001-001-000-011
- Solution 2,2 000-101-000-010-001-011-000-001--000-101-000-010-001-011-000-001

The GA is not aware of the relationship between the first 24 bits and the second 24 bits of the coded string. There are no GA operators used in the analyses in this chapter that can shift bits transversely to other positions on the string to assist the search. The GA considers the 48-bit coded string as a solution to a pipe network optimisation problem that requires the sizing of 16

pipes (with eight possible new or equivalent pipe sizes). The 48-bit string is separated into the two component 24-bit strings to both compute the pipe costs and test the hydraulic performance of the pipe network solutions. The total cost of the 48-bit string is the sum of the pipe costs and penalty costs (if any) of the component 24-bit strings. The GA itself, is not aware of the decoding and evaluation procedures; only of the strings of code and their associated fitnesses.

In pipe network designs, the suitability of a decision regarding the location, sizing or operation of a system component is dependent to some extent on the other decisions in the design. In the case of the solution of multiple Gessler problems, there is no relationship between the component 24-bit strings of the subproblems. It is difficult to say whether this lack of interaction across the component 24-bit substrings assists the GA search or impedes it in the recognition and propagation of short, highly fit string similarities (Schema Theorem of Holland, 1975).

The exhaustive enumeration of the Gessler network expansion problem identified 50 feasible solutions (listed in Table 5.4) within \$101,900 (5.82%) of the cost of the optimal solutions. The 50 solutions represent only 0.0003% of the total number of possible solutions for the Gessler problem. By comparison, for two Gessler networks, there are 2500 feasible network solutions within \$203,800 (5.82%) of the cost of the optimal solutions. The 2500 feasible network solutions represent only 0.000,000,000,9% of the total number of possible solutions. The 2500 solutions are the 50^2 combinations of the 50 solutions in Table 5.4.

7.2.1 The *improved* genetic algorithm approach

In the previous chapter, possible modifications to the traditional GA formulation were tested for the optimisation of the original Gessler network expansion problem. Based on the results of the experiments in Chapter 6, an *improved* genetic algorithm for pipe network optimisation is proposed for the optimisation of two Gessler problems. In this study, five improved GA runs of the double Gessler problem were conducted (designated D1, D2,..., D5) using the parameter sets given in Table 7.1. The improved GA model incorporates the following features:

- Penalty costs (for infeasible solutions) are based on the maximum violations of the pressure constraints for each loading condition (Eq. 6.1). A pressure violation penalty multiplier of $k=\$50,000/\text{psi}$ is used.
- Proportionate (roulette-wheel) selection is adopted using the inverse fitness function (Eq. 6.4) and the fitness scaling exponent, n is allowed to increase in steps as the GA run progresses according to Table 7.2.
- The 48-bit coded strings are formed by 3-bit substrings of Gray codes.

7 Larger problems with known optimal solutions

- One-point crossover is used and the crossover point may occur at any point in the 48-bit coded string. A high crossover rate $p_c=0.5-1.0$ is used (Table 7.1).
- Both random bit-wise mutations and creeping mutations are employed. The probability of a bit-wise mutation is low ($p_m=0.001-0.005$). The creeping mutation rate, $p_a=0.125$, is relatively high and on average two 3-bit substrings per 48-bit coded string will be subjected to a creeping mutation. The chance of creeping down is greater than the chance of creeping up ($p_d=0.6$).
- An elitist model is introduced (described in Section 7.2.2).

Population sizes of $N=100$ to 200 are considered (Table 7.1). The GA runs were allowed a maximum of 100,000 new solution evaluations which represents only $3.55 \times 10^{-8}\%$ of the total number of possible solutions. By comparison, the 10,000 solution evaluations for the original Gessler problem (Chapters 5 and 6) represented 0.06% of the 16,777,216 possible solutions.

The 100,000 new coded strings to be evaluated are formed by 1,000 generations of a population of $N=100$ strings (for GA run D1). The GA runs could be terminated if an optimal solution was determined before the specified maximum number of evaluations.

Table 7.1 Parameter sets D1-D5 for the optimisation of two Gessler problems

GA Parameters	D1	D2	D3	D4	D5
Population size, N	100	200	100	100	100
Maximum number of generations	1000	500	1000	1000	1000
Maximum number of evaluations	100,000	100,000	100,000	100,000	100,000
Probability of crossover, p_c	1.0	1.0	0.5	1.0	1.0
Probability of bit-wise mutation, p_m	0.005	0.005	0.005	0.001	0.005
Probability of creeping mutation, p_a	0.125	0.125	0.125	0.125	0.125
Probability of creeping down, p_d	0.6	0.6	0.6	0.6	0.6
Elite population size, N'	10	10	10	10	10
Probability of an elite mate, p_e	0.04	0.02	0.04	0.04	0.04
Penalty factor, k (\$m/psi)	0.05	0.05	0.05	0.05	0.05
Random number seed	100	100	100	100	200

Table 7.2 Variation of fitness scaling exponent, n for the GA runs D1-D5

Value of n	Evaluation Number Interval
$n = 1$	evaluations $\leq 25,000$
$n = 2$	$25,000 < \text{evaluations} \leq 50,000$
$n = 3$	$50,000 < \text{evaluations} \leq 75,000$
$n = 4$	$75,000 < \text{evaluations} \leq 100,000$

7.2.2 Elitism

The simple or traditional GA is driven by three operators including (1) roulette-wheel selection, (2) one-point crossover and (3) bit-wise complement mutations. In his study of the application of GAs to function optimisation, DeJong (1975) experimented with the traditional GA formulation and various other GA models including elitist models. DeJong's elitist model (Goldberg, 1989) preserves the best coded string solution identified by the GA up to the current generation, by including it in the new population of N coded strings as the $(N+1)$ th member, if it is not already a member of this new population. The new populations are generated by the traditional GA operators. DeJong (1975) found the elitist model improved the performance of the GA search in solution spaces with unimodal surfaces but degraded the performance for multimodal surfaces.

A form of elitism is introduced to the improved GA search developed in this thesis for pipe network optimisation. A small elite population of N' coded string solutions identified by the GA up to the current generation is maintained in parallel with the working population of N members. In the formation of the new working population from the old working population, elite population members are selected to mate with working population members with some low probability of an elite mate, p_e . The elite mates are chosen randomly from the elite population.

A coded string solution generated for the new working population by selection, crossover and mutation replaces a coded string in the elite population, if it survives a tournament with the randomly selected elite population member and it is not already a member of the elite population.

A schematic of the elitist model is shown in Figure 7.2. In the improved GA runs D1-D5 applied to the double Gessler problem, a small elite population size of $N'=10$ members is maintained and a low probability of an elite mate, $p_e=0.04$ is used (Table 7.1). A value of $p_e=0.04$ implies that about 4 of the 100 parent strings selected for mating are randomly picked from the elite population.

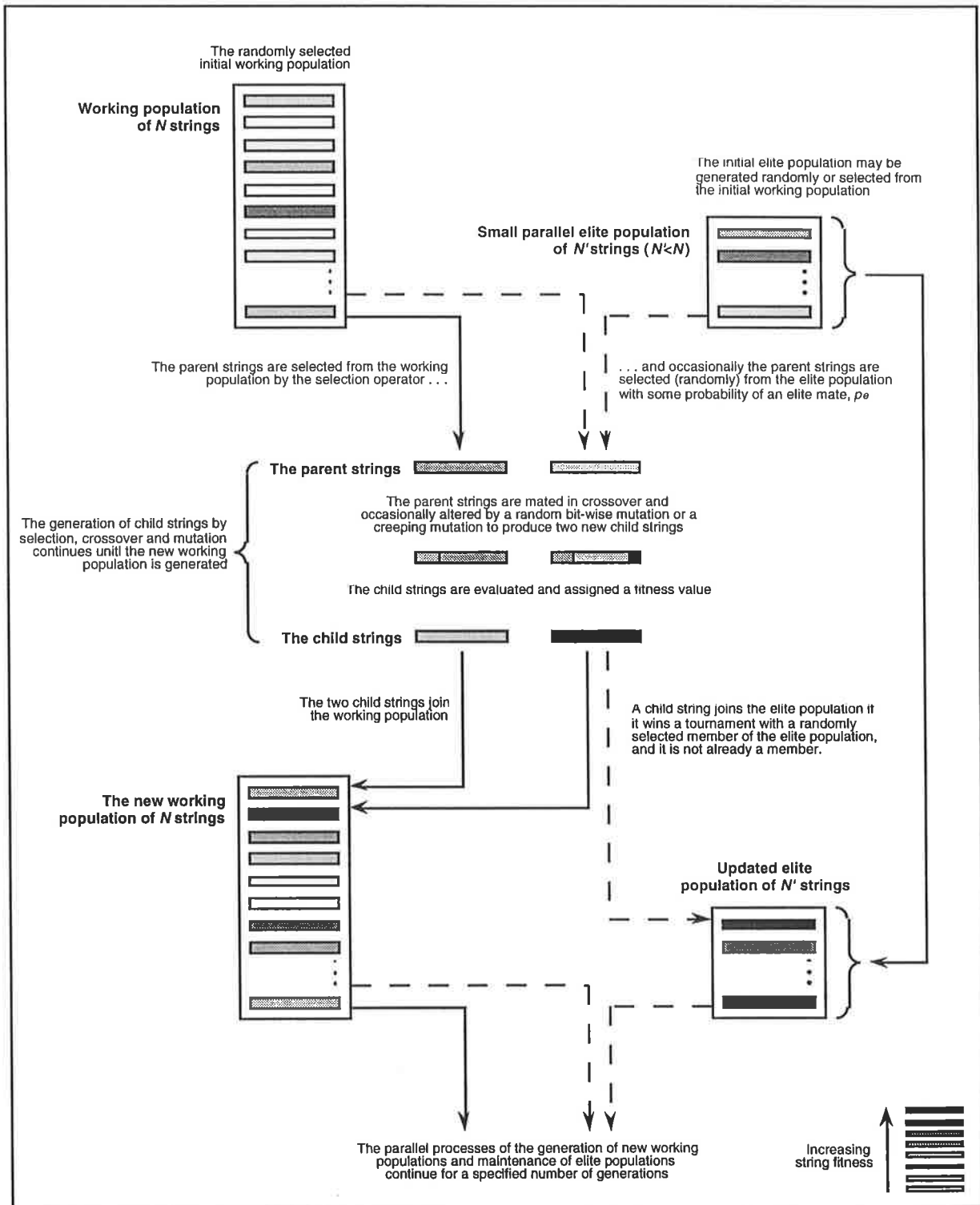


Figure 7.2 The elitist model

7.2.3 Performance of the improved GA (with elitism) applied to two Gessler problems

The improved GA formulation was applied to the search for the optimum designs of two Gessler pipe networks simultaneously in the five GA runs designated D1-D5. The GA runs utilised 116 minutes of CPU computer time on a SUN SPARCstation-1+ to complete the 100,000 evaluations for the double Gessler problem. The GA runs utilised only 14 minutes on a SUN SPARCstation-10. The 100,000 evaluations are equivalent to 600,000 hydraulic analyses since three demand patterns for two Gessler pipe networks are analysed to evaluate each 48-bit coded string. If the computation time to perform the genetic algorithm processes is negligible compared to the time to perform the hydraulic analyses, the time to perform one hydraulic analysis of the 14-pipe (and 4-loop) Gessler pipe network is about 0.0014 seconds.

The minimum cost solutions determined by GA runs D1-D5 are summarised in Table 7.3 and are optimum solutions in every case. A maximum of 79,600 evaluations were required.

Table 7.3 Improved GA results for the optimisation of two Gessler problems

GA run	Lowest cost solution (\$million)	Evaluation number	Solution combination (see Table 5.4)
D1	3.5006*	31,000	2,1
D2	3.5006*	79,600	2,1
D3	3.5006*	43,100	2,1
D4	3.5006*	38,000	2,1
D5	3.5006*	52,100	2,1
Average number of solution evaluations to determine the optimum solution = 49,000			

* global optimum solution

The occurrences of the individual solutions 1 and 2 in the five GA solutions are equal, however there is a dominance of the combined solution 2,1. The optimal solution combination 2,1 is first identified by GA run D1 after 31,000 solution evaluations, however, the optimal solutions are identified repeatedly within the maximum allowed 100,000 evaluations as shown by the plot of best solutions for each generation in Figure 7.3. The convergence of the average generation costs for GA run D1 is shown in Figure 7.4.

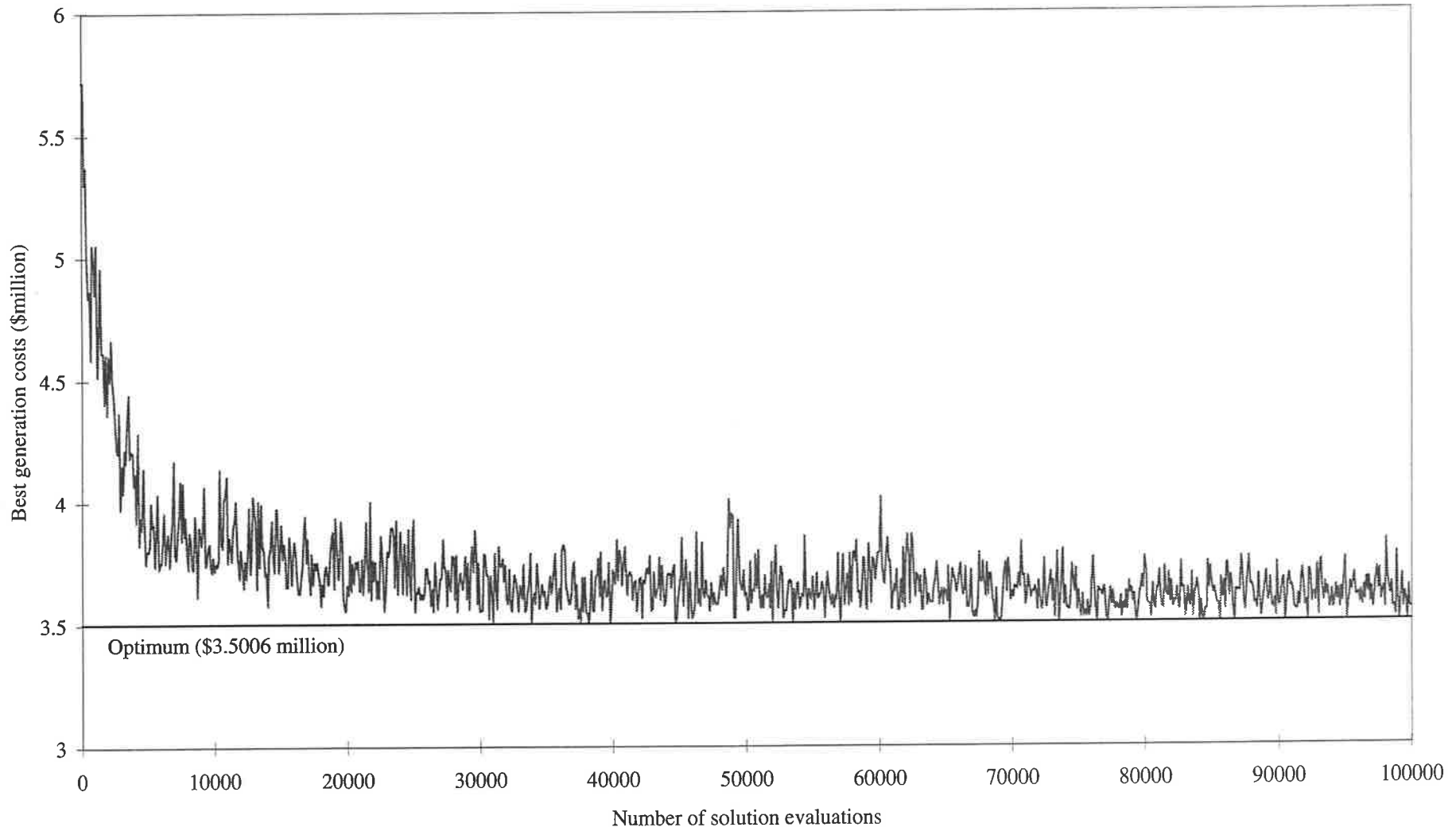


Figure 7.3 Best generation costs for the GA run D1 - two Gessler problems

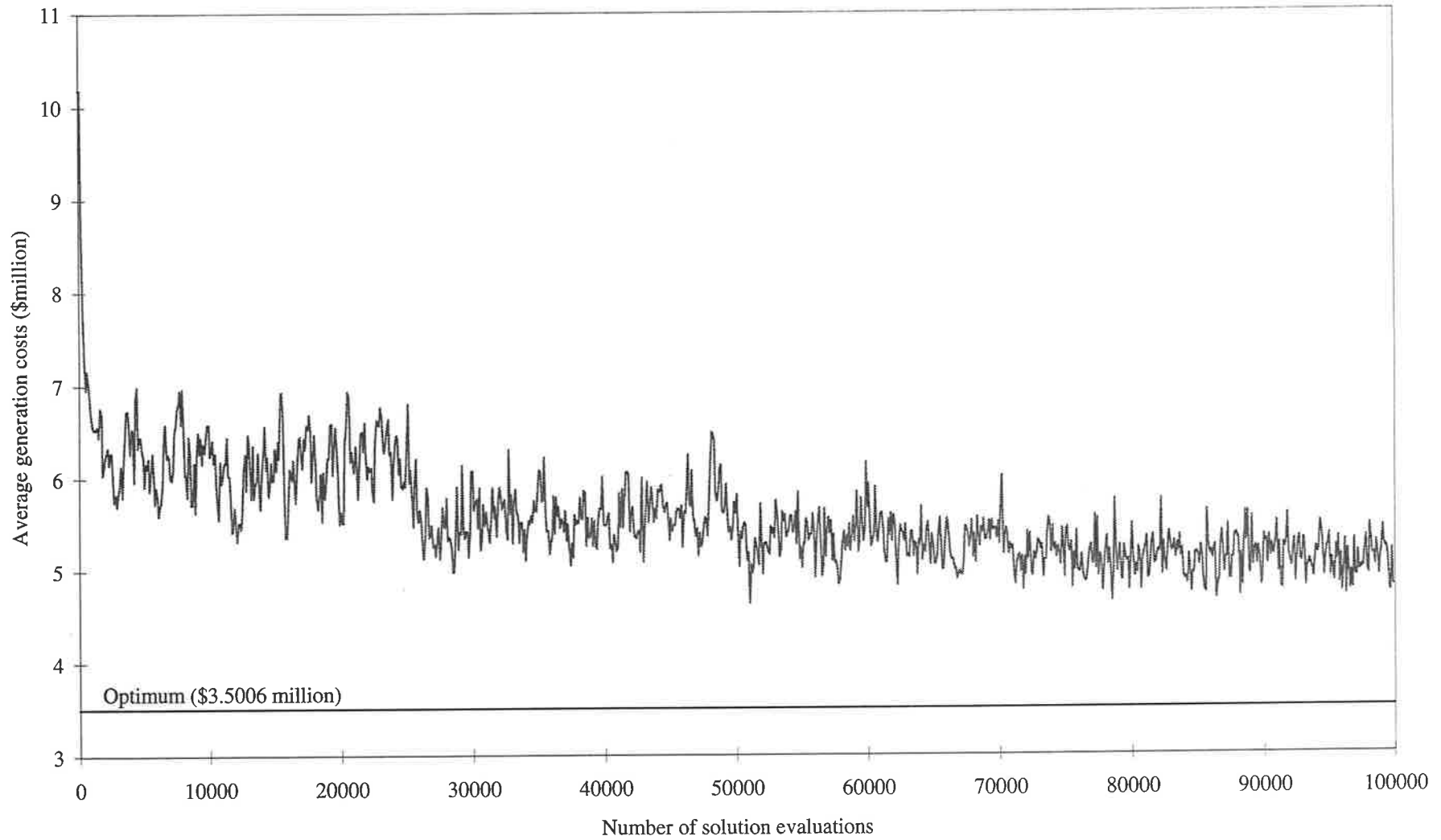


Figure 7.4 Average generation costs for the GA run D1 - two Gessler problems

7 Larger problems with known optimal solutions

The optimal solution is actually identified 29 times within the maximum 100,000 evaluations for GA run D1. The 29 occurrences of the optimal solution combinations are recorded in Table 7.4. The optimal strings are close to one another in the solution space. They are separated by a maximum Hamming distance of only 4 bits for a string of length 48 binary bits. The optimum solution 2,1 is found after 31,000 evaluations (310 generations) for GA run D1, and all of the four optimal solutions are identified in less than 38,200 evaluations (382 generations).

Table 7.4 Occurrences of optimal solutions for GA run D1

Solution	Hamming distance from solution 2,1	Number of occurrences	Number of generations performed to find the optimum solution
1,1	2 bits	3	382; 991; 997
1,2	4 bits	2	339; 571*
2,1	-	19	310; 398; 447; 520; 535; 571*; 653; 687; 764; 772; 794; 841; 844(2)†; 856; 867; 905; 922; 951
2,2	2 bits	5	376; 690; 691; 735; 843

* both solutions 1,2 and 2,1 determined after 571 generations

† solution 2,1 determined twice after 844 generations

7.3 Simultaneous Optimisation of Three Gessler Problems

This study can be extended to consider an even larger pipe network optimisation with known global optimum solutions by simultaneously considering three 14-pipe Gessler networks as an equivalent 42-pipe network. The GA searches for the optimal 72-bit coded string formed by three adjacent 24-bit coded strings. The optimum solution for the expanded solution space has a cost of \$5.2510 million. There are eight combinations of solutions 1 and 2 which are optimal pipe network designs. The solution space searched by the GA consists of $2^{72} = 4.72 \times 10^{21}$ possible solutions. There are 125,000 feasible solutions within \$305,700 (5.82%) of the cost of the optimal solution. The 125,000 solutions (representing only $2.65 \times 10^{-15}\%$ of all solutions) are the 50^3 combinations of the 50 solutions in Table 5.4.

The improved GA formulation is applied to the simultaneous optimisation of three Gessler networks. The GA parameter sets used in the previous optimisation (Table 7.1) are employed again in this more difficult optimisation, except the five GA model runs designated T1-T5 are allowed a maximum of 200,000 evaluations to search the solution space. The parameter sets T1-T5 are given in Table 7.5. The variation of the fitness scaling exponent, n for the GA runs of 200,000 evaluations (maximum) is given in Table 7.6. The GA runs of 200,000 evaluations utilised 350 minutes of CPU computer time on a SUN SPARCstation-1+ station (42 minutes on

7 Larger problems with known optimal solutions

a SUN SPARCstation-10). The 200,000 evaluations required 1.8 million hydraulic analyses (200,000 solutions of three networks for three demand patterns).

Table 7.5 Parameter sets T1-T5 for the optimisation of three Gessler problems

GA Parameters	T1	T2	T3	T4	T5
Population size, N	100	200	100	100	100
Maximum number of generations	2000	1000	2000	2000	2000
Maximum number of evaluations	200,000	200,000	200,000	200,000	200,000
Probability of crossover, p_c	1.0	1.0	0.5	1.0	1.0
Probability of bit-wise mutation, p_m	0.005	0.005	0.005	0.001	0.005
Probability of creeping mutation, p_a	0.125	0.125	0.125	0.125	0.125
Probability of creeping down, p_d	0.6	0.6	0.6	0.6	0.6
Elite population size, N'	10	10	10	10	10
Probability of an elite mate, p_e	0.04	0.02	0.04	0.04	0.04
Penalty factor, k (\$m/psi)	0.05	0.05	0.05	0.05	0.05
Random number seed	100	100	100	100	200

Table 7.6 Variation of fitness scaling exponent, n for the GA runs T1-T5

Value of n	Evaluation Number Interval
$n = 1$	evaluations $\leq 50,000$
$n = 2$	$50,000 < \text{evaluations} \leq 100,000$
$n = 3$	$100,000 < \text{evaluations} \leq 150,000$
$n = 4$	$150,000 < \text{evaluations} \leq 200,000$

The 200,000 evaluations represent only $4.24 \times 10^{-15}\%$ of the total number of possible solutions. To place these numbers in some perspective, consider the extent of the pipe network solution space can be represented by the total land area of Australia (which is approximately 7,682,300 square kilometres). Each solution to the optimisation of three Gessler problems would then occupy about 0.0016 mm^2 . The GA search of 200,000 evaluations for the optimal 72-bit coded string solution is the equivalent of investigating about 3.25 square centimetres of the total land area of Australia.

7 Larger problems with known optimal solutions

The lowest cost network solutions identified by the GA runs T1-T5 are presented in Table 7.7. The optimum solution for \$5.2510 million is determined in three out of five cases after about 180,000 evaluations. The solutions determined by the other two GA runs are near-optimal solutions. The occurrences of the individual solutions 1 and 2 are about equal.

Table 7.7 Improved GA results for the optimisation of three Gessler problems

GA run	Lowest cost solution (\$million)	Evaluation number	Solution combination (see Table 5.4)
T1	5.2510*	182,000	1,1,2
T2	5.3238	156,800†	15,2,2
T3	5.2510*	181,400	1,2,1
T4	5.2731	148,000†	3,1,2
T5	5.2510*	184,400	2,2,1

* global optimum solution

† a maximum of 200,000 evaluations were performed

The variation of best generation costs for the GA run T1 (shown in Figure 7.5) is very irregular, although there is a general downward approach towards the optimal solution. The corresponding convergence of average generation costs is shown in Figure 7.6.

The effect of the increases in the value of the fitness function exponent, n on average generation cost is clear for the GA run T1 in Figure 7.6. The transformation from $n=1$ to $n=2$ after 50,000 evaluations obviously reduces the average generation cost of the population. The changes from $n=2$ to $n=3$ after 100,000 evaluations and $n=3$ to $n=4$ after 150,000 evaluations are not as distinct, however, a reduction in average generation cost may be noticed.

The improved GA is found to be reasonably capable of handling pipe network design problems of this size (string length $l=72$ bits). Often, more than one GA run are performed to experiment with the GA parameters, particularly with the penalty factor k . In this case the most appropriate penalty factor was established in Section 6.3 of the previous chapter.

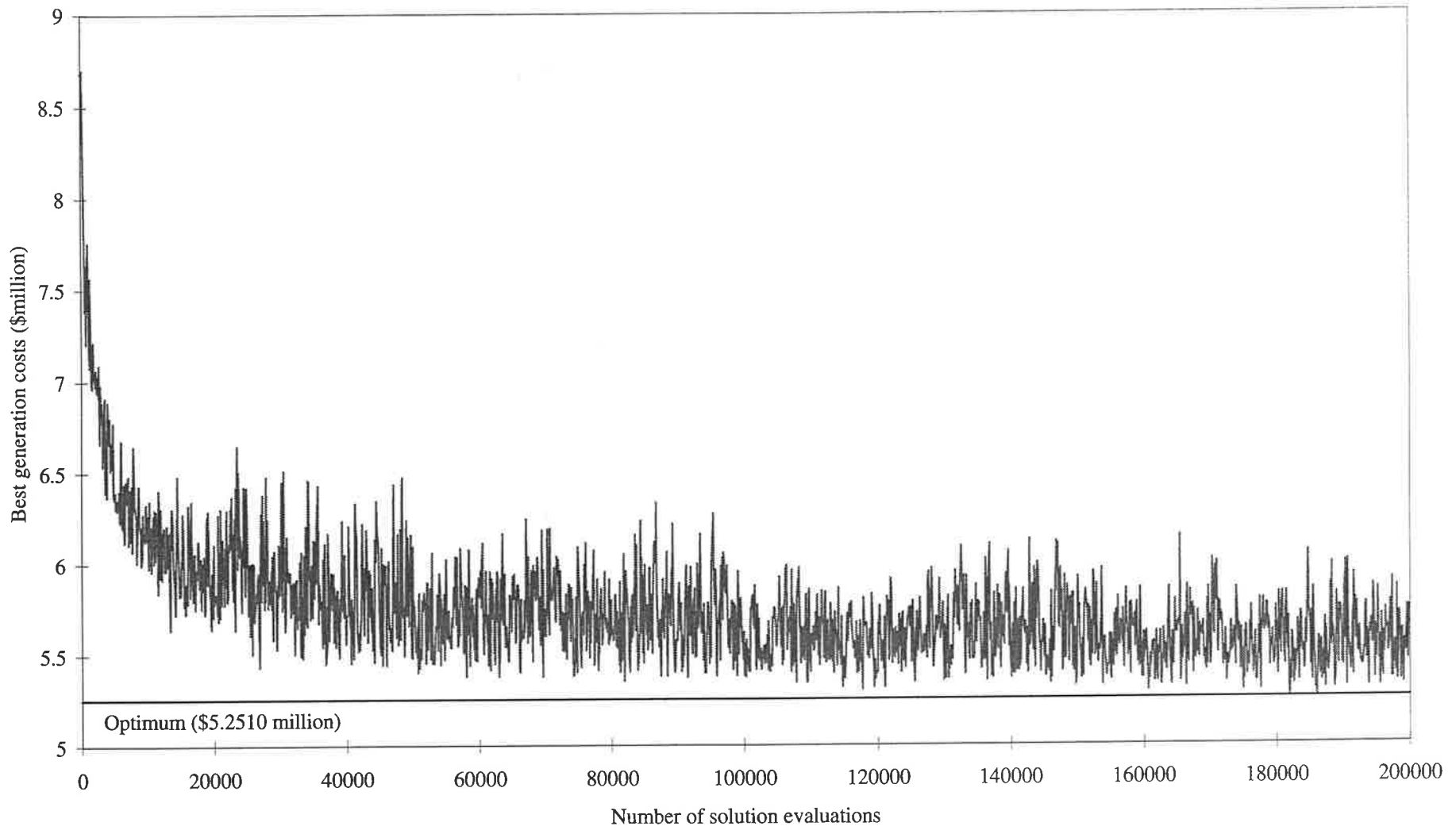


Figure 7.5 Best generation costs for the GA run T1 - three Gessler problems

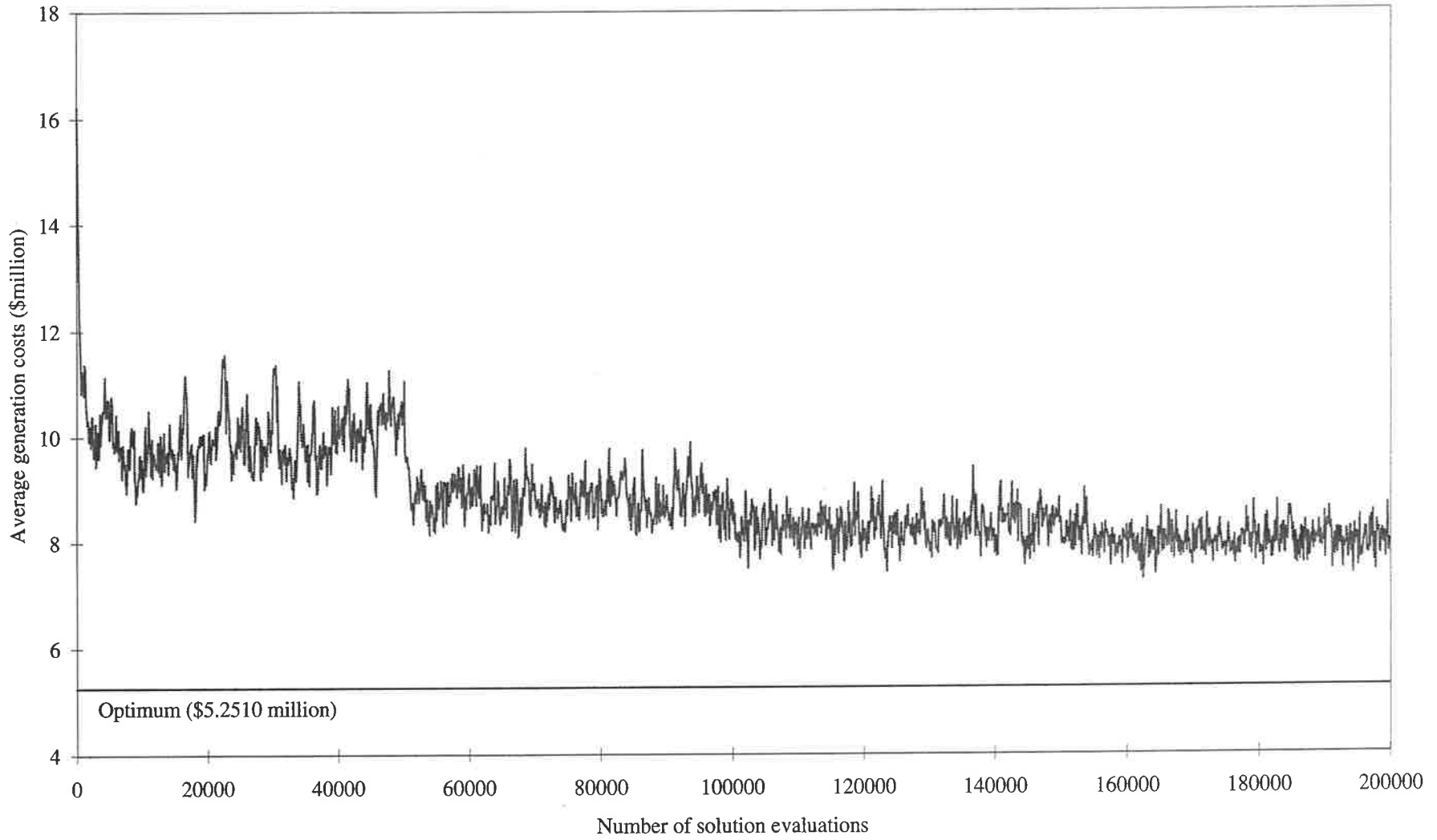


Figure 7.6 Average generation costs for the GA run T1 - three Gessler problems

7.4 Simultaneous Optimisation of Five Gessler Problems

Consider the simultaneous optimisation of five 14-pipe Gessler networks as an equivalent 70-pipe network. The GA searches for the optimal 120-bit coded string formed by five adjacent 24-bit coded strings. The optimum solution for the solution space has a cost of \$8.7516 million. There are 32 combinations of solutions 1 and 2 which are optimal pipe network designs. The solution space searched by the GA consists of $2^{120} = 1.329 \times 10^{36}$ possible solutions. Table 7.8 compares the size of the solution spaces and the GA runs for the simultaneous optimisation of two, three and five Gessler problems.

Table 7.8 A comparison of the various expanded solution spaces

The solution space and genetic algorithm run comparisons	Two Gessler Problems	Three Gessler Problems	Five Gessler Problems
Length of coded strings	48 binary bits	72 bits	120 bits
Total number of solutions in the solution space	2^{48} (2.815×10^{14})	2^{72} (4.722×10^{21})	2^{120} (1.329×10^{36})
Global optimum solution cost	\$ 3.5006 million	\$ 5.2510 million	\$ 8.7516 million
Number of global optima	4	8	32
Maximum number of string evaluations for the GA runs	100,000	200,000	400,000
Maximum fraction of solution space searched by the GA	$3.55 \times 10^{-8}\%$	$4.24 \times 10^{-15}\%$	$3.01 \times 10^{-29}\%$
Maximum number of hydraulic analyses	600,000	1,800,000	6,000,000
Total computational time (SUN SPARCstation-1+)	116 minutes	350 minutes	-
Total computational time (SUN SPARCstation-10)	14 minutes	42 minutes	140 minutes
Estimated time for one hydraulic analysis (SUN SPARCstation-10)	0.0014 secs	0.0014 secs	0.0014 secs

The improved GA formulation (described in Section 7.2.1) using substrings of Gray codes, proportionate selection with power law fitness scaling, creeping mutations and the elitist strategy is applied to the simultaneous optimisation of five Gessler networks. In order to measure the effectiveness of the elitist model, the improved GA without elitism is also applied

to the optimisation of five Gessler networks. Finally, to further demonstrate the capabilities of the improved GA formulation for pipe network optimisation, the traditional GA formulation (using substrings of binary codes and without fitness scaling, creeping mutations or elitism) is applied to the optimisation of five Gessler problems.

7.4.1 The GA parameter sets F1-F5

The three GA formulations (the improved GA with elitism, the improved GA without elitism and the traditional GA) used the GA parameter sets designated F1-F5 (with a few exceptions) for the optimisation of five Gessler networks as given in Table 7.9. The size of the solution space and the length of the binary string has increased significantly, and for this reason population sizes are increased and probabilities of bit-wise mutations and creeping mutations (for the improved GA runs) are decreased by comparison to the parameter sets D1-D5 used for two Gessler problems (Table 7.1) and T1-T5 used for three Gessler problems (Table 7.5).

Table 7.9 Parameter sets F1-F5 for the optimisation of five Gessler problems

GA Parameters	F1	F2	F3	F4	F5
Population size, N	200	400	200	200	200
Maximum number of generations	2000	1000	2000	2000	2000
Maximum number of evaluations	400,000	400,000	400,000	400,000	400,000
Probability of crossover, p_c	1.0	1.0	0.5	1.0	1.0
Probability of bit-wise mutation, p_m	0.0025	0.0025	0.0025	0.0025	0.0025
† Probability of creeping mutation, p_a	0.015	0.015	0.015	0.03	0.015
† Probability of creeping down, p_d	0.6	0.6	0.6	0.6	0.6
*† Elite population size, N'	20	20	20	20	20
*† Probability of an elite mate, p_e	0.03	0.015	0.03	0.03	0.03
Penalty factor, k (\$m/psi)	0.05	0.05	0.05	0.05	0.05
Random number seed	100	100	100	100	200

* not applicable to the improved GA runs without elitism

† not applicable to the traditional GA runs

Goldberg (1985) presented a theoretical basis for selecting initial population size. Goldberg found the optimal population size to be a function of the length of the binary strings and recommended a population size of $N=2,240$ strings for a string length of $l=50$ bits and $N=10,200$ for $l=60$ bits. A population size of $N>10,000$ is not really practical, however, the population size used for the optimisation of three Gessler problems, $N=100$ ($N=200$ for GA run T2) may have been inadequate for the string length of $l=72$ bits. The population size used

7 Larger problems with known optimal solutions

for the optimisation of five Gessler problems ($l=120$ bits) is increased to $N=200$ ($N=400$ for GA run F2). The populations of 200 strings are allowed to evolve over a maximum of 2,000 generations (a total of 400,000 solution evaluations). The GA runs are terminated if the optimal solution is determined before the generation of the 2,000 new populations.

The elite population size is increased to $N'=20$ members for the improved GA runs (with elitism). The probability of an elite mate is decreased to $p_e=0.3$ ($p_e=0.015$ for GA run F2). Since the population size has increased to $N=200$ members ($N=400$ for GA run F2), we expect about 6 elite mates to be selected as parent strings per generation of 200 (or 400) child strings.

The probability of crossover is maintained at $p_c=1.0$. As the string length gets longer, one-point crossover may not provide adequate mixing of population individuals. Although it was not considered in this study of multiple Gessler problems, multiple-point crossover may be more effective under these circumstances (Section 6.6). The probability of bit-wise mutation is decreased from $p_m=0.005$ (1 bit inverted for every 200 bits crossed over) to $p_m=0.0025$ (1 bit inverted for every 400 bits crossed over). The probability of creeping mutation (for improved GA runs) is significantly decreased from $p_a=0.125$ (1 substring mutated to an adjacent substring in the decision variable substring list for every 8 substrings crossed over) to $p_a=0.015$ (1 substring mutated for every 67 substrings crossed over). The binary strings of 120 bits are composed of 40 substrings of 3 binary bits. The strings are expected to be subjected to 1 crossover and an average of 0.3 random bit-wise mutations and 0.6 creeping mutations (improved GA runs) before forming the new population.

The GA runs designated F1-F5 were allowed a maximum of 400,000 solution evaluations. Each solution evaluation required 15 hydraulic analyses (three demand patterns for five pipe network designs per evaluation). Thus, a total of 6 million hydraulic analyses are performed for 400,000 evaluations. The approximation in Section 7.2 that the computation time to perform the genetic algorithm processes is negligible and the computation time therefore to perform one hydraulic analysis of the 14-pipe Gessler pipe network is 0.0014 seconds would indicate the GA runs of 6 million hydraulic analyses would require approximately 8,400 seconds. The GA runs required 8365 seconds (140 minutes) of CPU computer time on the SUN SPARCstation-10 to complete the 400,000 evaluations. The maximum 400,000 evaluations represents only $3.01 \times 10^{-29}\%$ of the total number of possible solutions.

The variation of the fitness scaling exponent, n throughout the improved GA runs of 400,000 solution evaluations (maximum) is given in Table 7.10.

Table 7.10 Variation of fitness scaling exponent, n for the GA runs F1-F5

Value of n	Evaluation Number Interval
$n = 1$	evaluations $\leq 100,000$
$n = 2$	$100,000 < \text{evaluations} \leq 200,000$
$n = 3$	$200,000 < \text{evaluations} \leq 300,000$
$n = 4$	$300,000 < \text{evaluations} \leq 400,000$

7.4.2 Performance of the improved GA (with elitism) applied to five Gessler problems

The performance of the improved GA (with the elitism strategy) is excellent and the optimum solution is identified by four of the five GA runs as shown in Table 7.11. The variations of the best generation costs and average generation costs for GA run F1 are given in Figures 7.7 and 7.8 respectively. The optimum solution was determined after completing 261,800 of the allowed 400,000 evaluations. The GA run F4 (higher creeping mutation, $p_a=0.03$) was the only GA run to perform the maximum 400,000 evaluations without identifying the optimum solution. The GA run F4 has probably become stuck on a local peak. The best solution obtained for \$8.8493 million is composed of four optimal solutions to the Gessler problem and solution 39 from Table 5.4. Solution 39 is represented by the coded string as shown. Solution 39 is a Hamming distance of 7 bits from the optimal solution 1 and 9 bits from the optimal solution 2.

Solution 39 011-111-011-011-001-001-001-011
 clean-dup.12"-clean-10"-8"-8"-8"-10"

Table 7.11 Results of the optimisation of five Gessler problems using the improved GA (with elitism)

GA run (with elitism)	Lowest cost solution (\$million)	Evaluation number	Solution combination (see Table 5.4)
F1	8.7516*	261,800	1,1,1,2,1
F2	8.7516*	298,800	2,2,2,1,1
F3	8.7516*	269,400	1,1,1,2,1
F4	8.8493	392,600 [†]	1,39,1,1,1
F5	8.7516*	141,200	1,2,2,1,1

* global optimum solution

[†] a maximum of 400,000 evaluations were performed

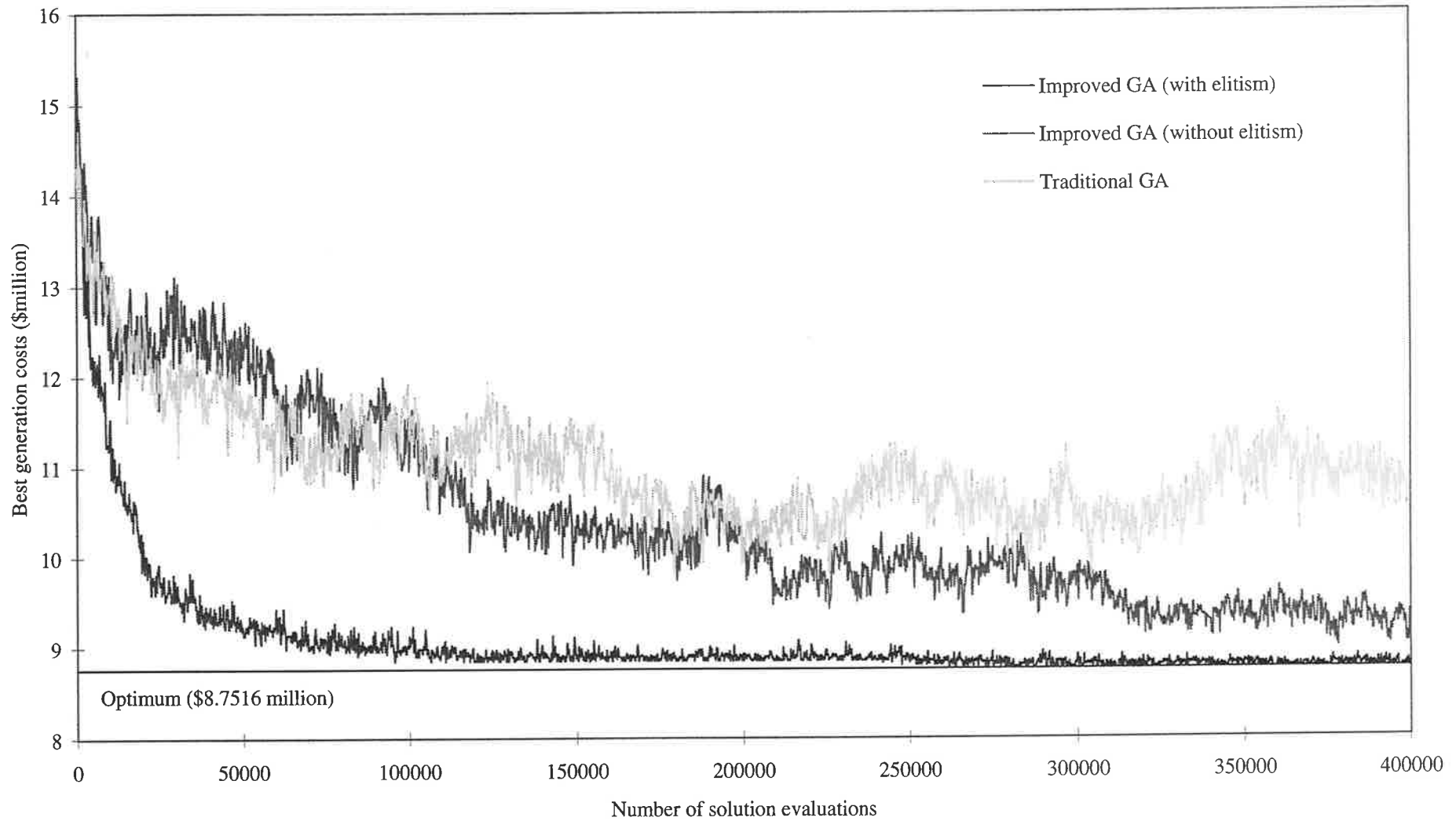


Figure 7.7 Best generation costs for GA runs F1 (improved GA with elitism), F1' (improved GA without elitism) and F1'' (traditional GA) - five Gessler problems

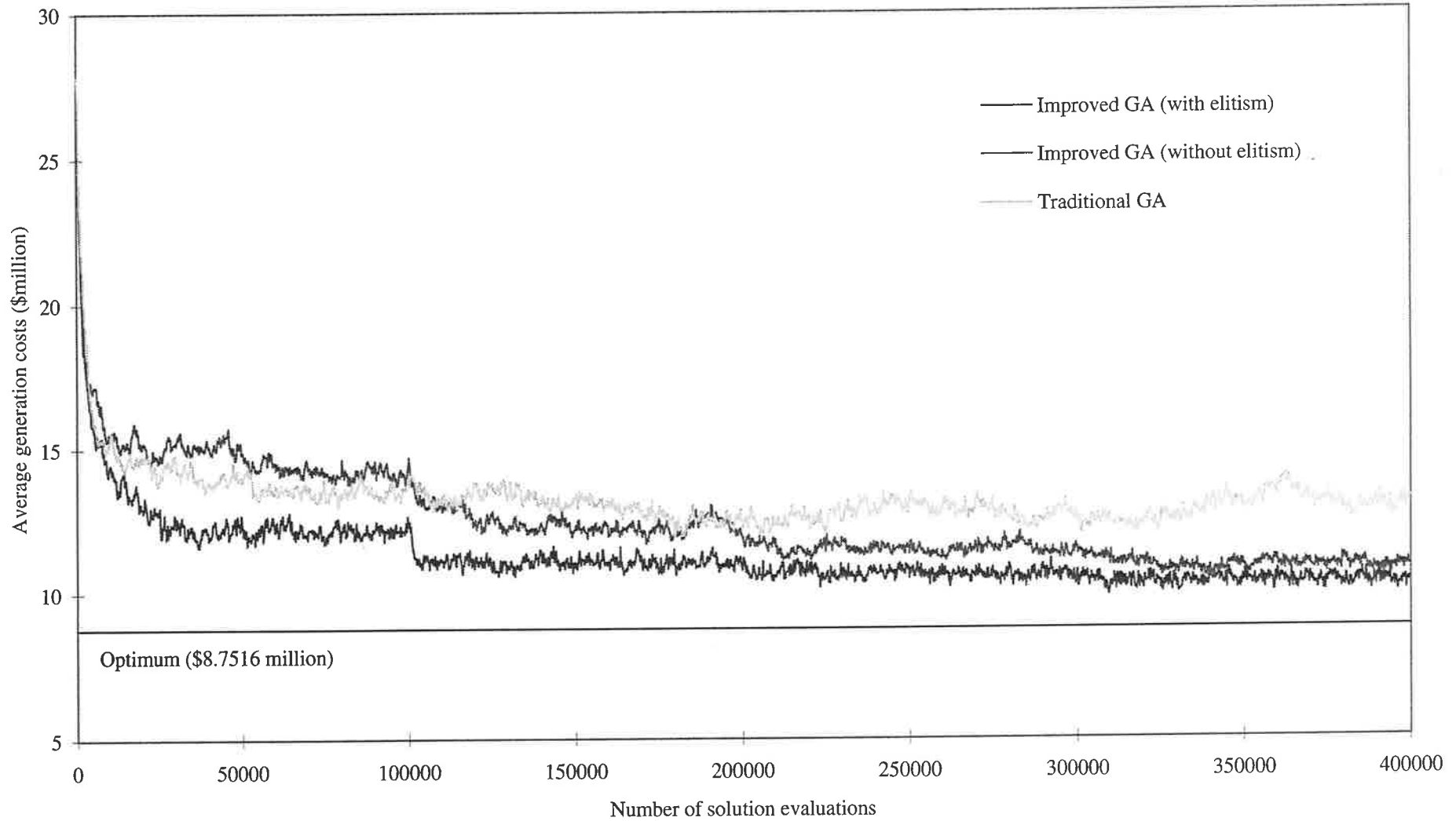


Figure 7.8 Average generation costs for GA runs F1 (improved GA with elitism), F1' (improved GA without elitism) and F1'' (traditional GA) - five Gessler problems

7.4.3 Performance of the improved GA (without elitism) applied to five Gessler problems

The improved GA (without the elitism strategy) does not determine the optimal solution with any of the five GA runs as shown in Table 7.12. The elitism is turned off by using a probability of an elite mate $p_e=0.0$. The elite population is maintained in any case so that the best N' solutions are saved and reported at the completion of the GA run. The maximum 400,000 evaluations are performed for all of the runs. The improved GA run F3' (without elitism) is very close to finding the optimum solution (0.0025% difference), although the GA run F4' is a relatively long way from the optimal solution (0.05% difference). Figures 7.7 and 7.8 compare the variations of the best generation costs and the average generation costs for the improved GA (with and without elitism) and the traditional GA for the GA parameter set F1.

Table 7.12 Results of the optimisation of five Gessler problems using the improved GA (without elitism)

GA run (no elitism)	Lowest cost solution (\$million)	Evaluation number	Solution combination (see Table 5.4)
F1'	8.9845	378,200 [†]	12,4,1,4,15 [#]
F2'	8.9807	367,600 [†]	1,4,2,-,3 ^{#*}
F3'	8.7738	381,200 [†]	3,1,2,2,2
F4'	9.1935	359,200 [†]	1,11,11,-,- [*]
F5'	8.9443	347,200 [†]	8,7,8,2,5

[†] a maximum of 400,000 evaluations were performed

^{*} some solutions do not appear in Table 5.4

[#] infeasible solution 1 from Table 5.29

7.4.4 Performance of the traditional GA applied to five Gessler problems

The traditional GA model represents solutions by coded strings of binary bits in binary codes (as opposed to Gray codes), and uses the original fitness function without the power fitness scaling, and there are no creeping mutations or a parallel elite population. The performance of the traditional GA summarised in Table 7.13 is inferior to that of the improved GA within 400,000 evaluations. The global optimum is not reached by any of the GA runs and the maximum 400,000 evaluations are performed for all of the GA runs. The traditional GA run F3'' is closest to the optimal solution (0.08% difference). The comparisons of the plots of best generation costs and average generation costs for GA model runs F1, F1' and F1'' in Figures

7.7 and 7.8 respectively show that the traditional GA actually performs better than the improved GA (without elitism) for the first 100,000 evaluations. The fitness scaling exponent, n is increased after 100,000 evaluations for the improved GA and the effect of this increase is significant.

Table 7.13 Results of the optimisation of five Gessler problems using the traditional GA

GA run (traditional)	Lowest cost solution (\$million)
F1''	9.7431
F2''	9.7889
F3''	9.4587
F4''	9.7431
F5''	10.0326

7.5 Conclusions

In this chapter, large pipe network optimisation solution spaces with known global optimal solutions have been manufactured by considering the simultaneous optimisation of two, three and five Gessler problems. The coded strings representing solutions to the multiple Gessler problem are formed by placing the component 24-bit strings representing solutions to the subproblems side by side. The coded string for the simultaneous solution of five independent Gessler problems has a string length of $l=120$ binary bits and represents a vast solution space of 1.329×10^{36} (2^{120}) possible solutions. Of course, the global optimum solution is not usually known for solution spaces of this size.

An *improved* genetic algorithm has been developed for this application to multiple Gessler problems. The improved GA formulation (based on the results of the experimental analyses of changes to the traditional GA formulation in Chapter 6) incorporates coded solutions represented by strings of binary bits in Gray codes, power law fitness scaling and creeping (decision-variable-wise or adjacency) mutations. In addition, an elitist concept has been introduced to the improved GA formulation.

In the elitist strategy, a small (N' members) elite population of the best solutions previously determined by the GA is maintained in parallel to the working population through the generations. The members of the elite population are selected as parent strings with some low probability of an elite mate, p_e .

7 Larger problems with known optimal solutions

The improved GA model is very effective and quite efficient in searching the vast, complex solution spaces for the multiple Gessler problems. The power of the elitist strategy is remarkable. The effectiveness of the elitist model is demonstrated by the application of the improved GA model (with and without elitism) to the optimisation of five Gessler problems. The traditional GA formulation is also applied to the optimisation of five Gessler problems. The improved GA (with elitism) clearly demonstrates superior performance to the improved GA (without elitism) and in turn, the improved GA (without elitism) is superior to the traditional GA for this study. The global optimum is identified in 4 out of 5 occasions by the improved GA (with elitism) after relatively few solution evaluations (less than 400,000) in the vast solution space for the simultaneous solution of five Gessler problems. By comparison, the improved GA (without elitism) and the traditional GA are unable to locate the global optimum solution.

The maintenance of an elite population of the previous best solutions allows for relentless *exploitation* of past results by selection and extensive *exploration* by crossover and mutation mechanisms, without moving away from the best regions of the solution space previously identified. Further research is required to determine the most appropriate values of elitist model parameters of elite population size and probability of an elite mate, and to investigate other issues such as the method of selection of elite mates from the elite population and the method of updating the elite population.

The objective of the elitist model is to keep the genetic algorithm search on the right track. It is hoped, that by the same principle, the elitist model does not lead the genetic algorithm search down the wrong track. It may be necessary to vary the probability of an elite mate as the genetic algorithm run proceeds to avoid the elite population being dominated by solutions from better than average, but not optimal, regions of the solution space. If the elite population is not improving, it may be necessary to suppress the selection of elite mates (reduce the probability of an elite mate to zero) for a number of generations to allow the genetic algorithm a chance to explore other regions of the solution space without the influence of the elite population. The elite population may be updated as usual during this time. The initial results for the improved genetic algorithm formulation including the elitist model presented in this chapter are quite promising.

8 An Improved Genetic Algorithm Formulation Applied to the New York Tunnels Problem

The benchmark New York City water supply tunnels optimisation problem has been comprehensively studied since the problem first appeared (Schaake and Lai, 1969) in the early days of the consideration of the pipe network optimisation problem. A number of researchers have reported the results of their studies of the New York tunnels problem in the pipe network optimisation literature. Nonlinear programming, linear programming and enumeration pipe network optimisation models have been applied to the New York tunnels problem.

The purpose of this chapter is to demonstrate the new genetic algorithm (GA) formulation developed for pipe network optimisation in this thesis by applying it to the New York tunnels network optimisation problem. The GA uses variable power scaling of the fitness function. The exponent in the fitness function is increased in magnitude as the GA computer run proceeds in order to stretch the range of fitness values (Goldberg, 1989). A creeping mutation operator complements the more commonly used random bit-wise mutation operator. Gray coding is used in the improved GA instead of binary coding to represent the decision variables as coded substrings. A subtle form of the elitism strategy which demonstrated success when applied to the optimisation of multiple Gessler networks in Chapter 7 is employed again here.

Results are presented comparing the *simple* traditional GA formulation and the improved GA formulation for the New York tunnels problem. The results indicate that the improved GA performs significantly better than the traditional GA. In addition, the solutions obtained by the improved GA are compared with the solutions obtained by traditional optimisation methods such as linear programming and nonlinear programming methods and a partial enumeration algorithm. Application of the improved GA to the New York tunnels problem provides the lowest cost, feasible, discrete pipe size solutions yet presented in the literature.

8.1 The New York Tunnels Problem

In 1969, Schaake and Lai developed an optimisation technique to determine the most economical design for the proposed additions to the primary water distribution system of New York City. The existing system (as defined in 1969) was composed of a network of 21 deep rock tunnels of large diameter (up to 204 inches). Part of the proposed expansions included the construction of *duplicate* gravity tunnels parallel to the existing tunnels to enable the system to meet the increasing water demands. The New York City primary water supply tunnel system (as considered by Schaake and Lai, 1969) is shown in Figure 8.1.

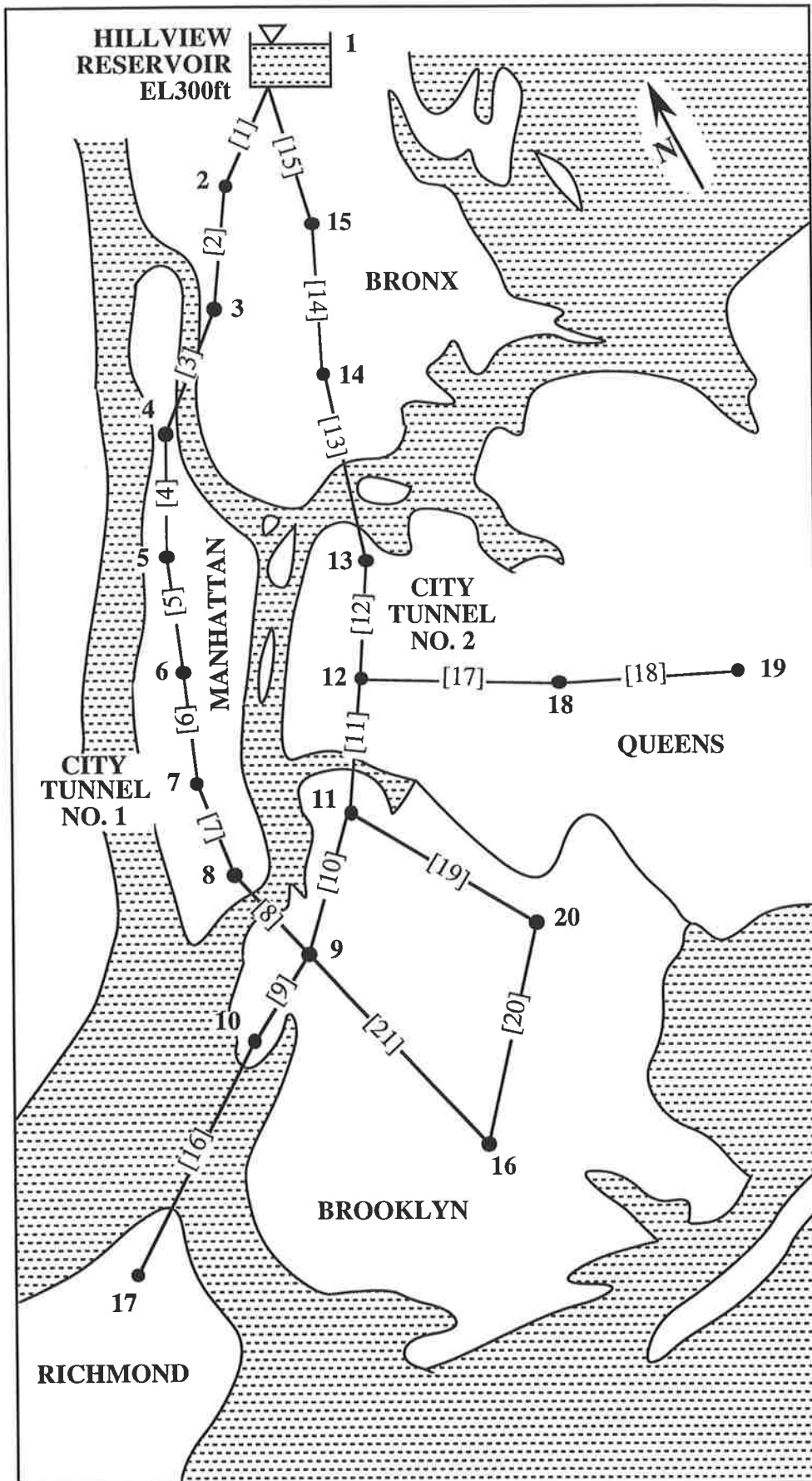


Figure 8.1 New York City water supply tunnels network in 1969

The primary tunnel system consisted of City Tunnels No.1 and No.2. City Tunnel No.1 extended from Hillview Reservoir to node 16 in Brooklyn by way of Manhattan. City Tunnel No.2 extended between Hillview Reservoir and Richmond downtake by way of Queens. City Tunnel No.1 was constructed in about 1920 and City Tunnel No.2 was constructed in about 1940 (de Neufville et al., 1971). The age of the City Tunnels and possible population increases and the consequent increased water demands indicated the need to consider possible expansions to the existing network.

A single demand pattern was considered for designing the improved tunnel system and a corresponding minimum hydraulic grade line (HGL) profile was specified for the nodes as given in Table 8.1. A hydraulic simulation of the projected demands applied to the existing tunnel system shows that nodes 16, 17, 18, 19 and 20 fall significantly below the minimum HGL profile (Bhave, 1985). Nodes 1 to 15 have acceptable hydraulic grade line elevations. The tunnel system is a gravity flow system that draws water (2017.5 cfs) from the Hillview Reservoir at node 1. The lengths and diameters of the 21 existing tunnels are tabulated in Table 8.2. A Hazen-Williams roughness coefficient $C = 100$ is assumed for all new and existing tunnels. Imperial units were used in this study to allow for comparisons with previous studies.

Table 8.1 Nodal data for the New York City water supply tunnels

Node	Demand (cfs)	Minimum HGL (ft)
1	Reservoir	300.0
2	92.4	255.0
3	92.4	255.0
4	88.2	255.0
5	88.2	255.0
6	88.2	255.0
7	88.2	255.0
8	88.2	255.0
9	170.0	255.0
10	1.0	255.0
11	170.0	255.0
12	117.1	255.0
13	117.1	255.0
14	92.4	255.0
15	92.4	255.0
16	170.0	260.0
17	57.5	272.8
18	117.1	255.0
19	117.1	255.0
20	170.0	255.0

Table 8.2 Existing tunnel data for the New York City water supply tunnels

Tunnel	Start node	End node	Length (ft)	Existing diameter (in)
[1]	1	2	11600	180
[2]	2	3	19800	180
[3]	3	4	7300	180
[4]	4	5	8300	180
[5]	5	6	8600	180
[6]	6	7	19100	180
[7]	7	8	9600	132
[8]	8	9	12500	132
[9]	9	10	9600	180
[10]	11	9	11200	204
[11]	12	11	14500	204
[12]	13	12	12200	204
[13]	14	13	24100	204
[14]	15	14	21100	204
[15]	1	15	15500	204
[16]	10	17	26400	72
[17]	12	18	31200	72
[18]	18	19	24000	60
[19]	11	20	14400	60
[20]	20	16	38400	60
[21]	9	16	26400	72

Hazen-Williams roughness $C = 100$ for all pipes

The genetic algorithm technique considers a set of design variable choices that are 15 discrete tunnel sizes and the alternative of *not duplicating* the existing tunnel. The available tunnel sizes considered for the New York tunnels additions are presented in Table 8.3. The tunnel cost function given by Eq. 8.1 relating tunnel construction cost to new tunnel diameter used in the original work is used in this study.

$$TC_i = 1.1 D_i^{1.24} L_i \quad (8.1)$$

in which TC_i = construction cost of tunnel i (1969 US dollars)
 D_i = diameter of tunnel i (in)
 L_i = length of tunnel i (ft)

Table 8.3 Available tunnel sizes and construction costs for New York tunnels duplications and the corresponding coded substrings

Diameter (inches)	Unit tunnel cost (\$/ft)	Corresponding coded substrings	
		Binary code	Gray codes
0 (do nothing)	0	0000	0000
36	93.5	0001	0001
48	134.0	0010	0011
60	176.0	0011	0010
72	221.0	0100	0110
84	267.0	0101	0111
96	316.0	0110	0101
108	365.0	0111	0100
120	417.0	1000	1100
132	469.0	1001	1101
144	522.0	1010	1111
156	577.0	1011	1110
168	632.0	1100	1010
180	689.0	1101	1011
192	746.0	1110	1001
204	804.0	1111	1000

8.2 The Genetic Algorithm Optimisation Approach

The improved GA formulation for pipe network optimisation developed in the preceding chapters has proven to be a successful GA model for this application. The main elements of the improved GA formulation are outlined in Table 8.4.

The GA approach to the optimisation of the expansions to the New York tunnels can be separated into three parts:

- 1) Firstly, the formulation of the GA is established for the New York tunnels optimisation problem including a coded string structure, a decoding scheme, a fitness function, the GA operators and appropriate GA parameters (Section 8.3).
- 2) In the second part of the GA study, the improved GA formulation is applied to the New York problem for different penalty multipliers to establish the most appropriate penalties for violations of the node minimum HGL constraints (Section 8.4).
- 3) Finally, nine sets of GA parameters (population sizes, probabilities of crossover and bit-wise mutation) are investigated. The performance of the GA search is evaluated for the traditional three-operator GA, the improved GA formulation (for pipe network optimisation) and various intermediate GA formulations to measure the effectiveness of the various proposed modifications to the simple GA formulation (Section 8.5).

The differences between the traditional GA and the improved GA formulation developed in this thesis for pipe network optimisation are summarised in Table 8.4.

Table 8.4 The improved GA compared to the traditional GA formulation

Feature of GA formulation	Traditional GA formulation	Improved GA formulation
Coded strings:	Substrings of binary code	Substrings of Gray codes
Fitness values:	Raw fitness values	Variable power scaling of raw fitness values
GA operators:	(1) Reproduction (2) Crossover (3) Random bit-wise mutations	(1) Reproduction (2) Crossover (3) Random bit-wise mutations (4) Random creeping mutations
Other:		Parallel population of elite strings

8.3 The Improved Genetic Algorithm Formulation

8.3.1 Structure of the coded strings

The trial designs for the New York tunnel network expansions generated by the GA are represented by strings of code using the binary alphabet {1,0}. The coded strings of 84 binary bits, consist of 21 coded substrings of 4 binary bits each as shown in Figure 8.2. The 21 coded substring positions correspond to the 21 existing tunnels that may be duplicated. A substring of 4 binary bits permits representation of the 16 alternative discrete design variable choices considered for the New York problem as demonstrated in Table 8.3. The New York tunnels problem is a difficult problem with a significantly larger search space than the Gessler problem. The solution space considered by the GA consists of the set of 2^{84} or 1.934×10^{25} unique solutions for the coded string length of 84 binary bits.

8.3.2 Binary codes and Gray codes

The chosen mapping between coded substrings and design variable choices associates the artificial genetic code with a pipe network design. The design variable choices are the possible diameters of the new parallel tunnels for the New York primary tunnels system. Traditionally, binary codes have been used to specify the mapping. The improved GA formulation uses Gray codes (Section 6.5). Table 8.3 presents the alternative mappings between design variable choices and decision-variable substrings for both coding schemes.

The tunnels corresponding to the substrings

[1] -[2] -[3] -[4] -[5] -[6] -[7] -[8] -[9] -[10]-[11]-[12]-[13]-[14]-[15]-[16]-[17]-[18]-[19]-[20]-[21]

Substrings in Gray codes

GA (1) \$38.796m	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-1100-0111-0101-0111-0110-0000-0110
GA (2) \$39.062m	0000-0000-0000-0000-0000-0000-1111-0000-0000-0000-0000-0000-0000-0000-0000-0101-0100-0110-0110-0000-0110
GA (3) \$39.166m	0000-0000-0000-0000-0000-0000-1110-0000-0000-0000-0000-0000-0000-0000-0000-0101-0101-0111-0110-0000-0110
GA (4) \$39.221m	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-1100-0111-0100-0110-0110-0000-0110
GA (5) \$39.284m	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0100-0101-0101-0111-0110-0000-0110
GA (6) \$38.524m*	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0101-0101-0101-0111-0110-0000-0110
GA (7) \$36.190m*	0000-0000-0000-0000-0000-0000-0111-0000-0000-0000-0000-0000-0000-0000-0000-0101-0101-0111-0110-0000-0110
GA (8) \$33.626m*	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0101-0101-0111-0110-0000-0110

Substrings in binary codes

GA (1) \$38.796m	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-1000-0101-0110-0101-0100-0000-0100
GA (2) \$39.062m	0000-0000-0000-0000-0000-0000-1010-0000-0000-0000-0000-0000-0000-0000-0000-0110-0111-0100-0100-0000-0100
GA (3) \$39.166m	0000-0000-0000-0000-0000-0000-1011-0000-0000-0000-0000-0000-0000-0000-0000-0110-0110-0101-0100-0000-0100
GA (4) \$39.221m	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-1000-0101-0111-0100-0100-0000-0100
GA (5) \$39.284m	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0111-0110-0110-0101-0100-0000-0100
GA (6) \$38.524m*	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0110-0110-0110-0101-0100-0000-0100
GA (7) \$36.190m*	0000-0000-0000-0000-0000-0000-0101-0000-0000-0000-0000-0000-0000-0000-0000-0110-0110-0101-0100-0000-0100
GA (8) \$33.626m*	0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0000-0110-0110-0101-0100-0000-0100

* Infeasible Designs

Figure 8.2 Coded strings representing the best GA designs

The Gray code representation is such that neighbouring decision-variable substrings differ by only one bit or a Hamming distance of 1. Since similar genetic code represents adjacent design variable choices, the Gray codes ensure trial solutions that are nearby in the solution space are represented by similarly coded strings. By comparison, neighbouring substrings in binary codes may differ by any number of bits. The extreme Hamming distance of 4 for bit strings of length 4 is referred to as a Hamming cliff (between substrings 0111 and 1000). It would be difficult to alternate between these coded substrings during the course of the GA run using bit-wise mutations or one-point crossover alone, even though in binary codes they represent adjacent pipe sizes.

8.3.3 Raw fitness of a coded string

The new coded strings generated by the GA are decoded to trial tunnel network designs by observing the mapping between substrings and design variable choices in Table 8.3. The trial designs are evaluated in terms of hydraulic performance and estimated tunnel construction costs. The strings in a population are then accompanied by appropriate measures of fitness. The raw fitness f_j of a string j is a function of tunnel construction costs TC_i for the tunnels i and a penalty cost PC for unacceptable system performance:

$$f_j = \frac{1}{PC + \sum_{i=1}^{21} TC_i} \quad (8.2)$$

8.3.4 The reproduction operator

The GA reproduction operator adopted is a proportionate selection scheme (with replacement) which chooses parent strings for the next generation according to their fitness with respect to the fellow strings of their generation. The probability of selection, p_j of string j in reproduction is given by Eq. 8.3. The string's fitness differentiates it from stronger and weaker strings in its own generation allowing it an appropriate chance of selection in reproduction. There remains a degree of randomness about the proportionate selection process.

$$p_j = \frac{f_j}{\sum_{n=1}^N f_n} \quad (8.3)$$

8.3.5 Scaled fitness of a coded string

The traditional GA considers the raw fitness of coded strings in reproduction. The improved GA adopts a power law fitness scaling mechanism to adjust the raw fitness of strings in a population to maintain an appropriate level of competition between strings (Goldberg, 1989).

$$f'_j = f_j^n \quad (8.4)$$

The fitness scaling function in Eq. 8.4 is used in this study to adjust the calculated raw fitness f_j of the strings j in a population to the scaled fitness f'_j . The value of the exponent n is allowed to increase in steps as the GA run progresses. The early populations of the GA search are formed by diverse sets of strings, all holding potentially valuable genetic information. The exponent is low in the early generations, while the GA assesses the potential strengths of the assortment of strings. The value of the exponent is increased in steps during the intermediate generations.

As the GA search develops further, the strings in a population are constructed of similar genetic code. The raw fitness values are usually very similar, such that the probabilities of selection of the strings from a population are indiscernible to the proportionate selection operator. A high value of the exponent is used to accentuate the small differences in string raw fitness. The exponent n was allowed to vary throughout the improved GA search:

$$n = \begin{cases} 1 & \text{evaluations performed} \leq 50,000 \\ 2 & 50,000 < \text{evaluations} \leq 100,000 \\ 3 & 100,000 < \text{evaluations} \leq 150,000 \\ 4 & 150,000 < \text{evaluations} \leq 200,000 \end{cases}$$

8.3.6 The penalty function

The pipe network design is subject to a set of system performance constraints that may be included in the GA search by way of a penalty function. The New York tunnels network expansions are subject to a pattern of node demands and the proposed designs are required to achieve a minimum hydraulic grade line (HGL) profile (Table 8.1). *Infeasible* solutions are unacceptable designs which do not achieve the specified system performance requirements.

The penalty function in Eq. 8.5 applies a penalty cost PC to infeasible solutions that is a function of the degree by which the design violates the hydraulic head constraints. The penalty function for pipe network optimisation established in Section 6.3 is a linear function of the

maximum deficit of hydraulic head. The pressure violation penalty cost is the product of the maximum violation and a specified penalty multiplier, k .

$$PC = \begin{cases} k \cdot \left[\max_m (H_m^{min} - H_m) \right] & \text{for demand nodes } m : H_m < H_m^{min} \\ 0 & \text{if for all } m : H_m \geq H_m^{min} \end{cases} \quad (8.5)$$

in which m = nodes where the minimum HGL constraints are not satisfied, H_m^{min} = minimum allowable hydraulic head at node m , H_m = measured hydraulic head at node m and k = penalty multiplier (\$/unit of hydraulic head). The selection of an appropriate penalty multiplier is considered in Section 8.4.

8.3.7 Creeping mutations

A creeping mutation operator introduced in Section 6.7 is used in the improved GA formulation. The creeping mutation operator changes a randomly selected decision-variable substring to an adjacent substring up or down the list of decision-variable substrings with some probability, p_a . For example, the substring 0001 may change up to the substring 0011 (in Gray codes) or change down to the substring 0000. The creeping mutations contrast with the traditional random bit-wise mutations that may or may not produce an adjacent decision variable choice. For example, the substring 0000 may be altered by a bit-wise mutation to 1000, 0100, 0010 or 0001. Although these substrings modified by bit-wise mutations are only a Hamming distance of 1 from the original substring, the substrings 1000 and 0100 represent significantly different designs compared to 0000. Random bit-wise mutations are still important as they add diversity to the gene pool. The creeping mutations and the traditional random bit-wise mutations with low probability were used simultaneously in the improved GA formulation.

A creeping mutation occurs when a substring of a new string for the new generation is selected. This occurs with a specified probability of creep p_a . The creeping mutations could be performed on any string as part of the formation of the new population. A value of $p_a=0.04$ is employed in the improved GA runs which implies about 1 in 25 substrings in the new population will shift to an adjacent substring. Since the coded strings for the New York tunnels GA search consist of 21 substrings, most strings are likely to be subject to a creeping mutation.

The creeping mutation operator allows for different probabilities of creeping up or down the list of decision-variable substrings. Given that a creeping mutation will proceed, a probability of creeping down, $p_d=0.6$ was used in the ensuing improved GA runs, implying there is a 60% chance of creeping down the substring list towards 0000 (to smaller tunnel diameters) and a

40% chance of creeping up. The probability of the direction of creeping mutation is biased in the downward direction with the view of enhancing the chance of moving towards lower cost solutions. The subtle creeping mutations provide fine adjustments in the vicinity of the current solutions.

8.3.8 Elitism

An elitism concept was introduced in Section 7.2.2. A small population of N' elite members is maintained in parallel to the natural working populations of N coded strings. The elite members consist of the best strings determined at any of the earlier generations of the GA. A number of the elite strings are mated with strings from the natural population with some probability of an elite mate, p_e . The natural population mates are selected by the proportionate selection operator. The elite mates are selected randomly from the small parallel elite population.

An elite population size of $N'=10$ members and a probability of an elite mate $p_e=0.01$ is considered for the improved GA applied to the New York tunnels problem. A value of $p_e=0.01$ and a population size of $N=200$ imply approximately $(p_e)(N)=2$ of the 200 strings selected to be parent strings for the new population will be strings from the elite population.

8.3.9 Population size, crossover and random bit-wise mutations

The improved GA runs in Section 8.4 experiment with the penalty function for a set of GA parameters which are expected to perform effectively for a problem of this size. The primary GA parameters are chosen to be population size, $N=200$, probability of crossover, $p_c=1.0$ and probability of mutation, $p_m=0.005$. The GA runs in Section 8.5 consider the performance of the various formulations of the GA model across nine alternative sets of GA parameters. The GA parameter sets in Section 8.5 systematically vary from the primary GA parameter set.

Population sizes of coded strings of $N=100$, $N=200$, $N=500$ and $N=1,000$ were considered for a crossover probability $p_c=1.0$ and a probability of mutation $p_m=0.005$. The GA runs were allowed a maximum of 200,000 objective function evaluations which is only a relatively small fraction of the entire solution space of 1.934×10^{25} possible solutions. The GA runs used approximately 6 minutes of CPU time on a SUN SPARCstation-10 (using SunOS Release 4.1.3) for the 200,000 function evaluations.

A function evaluation is required for every new coded string created in a new generation when an old string is disrupted by crossover or mutation or by another GA operator such as creeping mutation. The expected number of generations for 200,000 evaluations can be computed by considering the expected number of new strings created in a new population. In the following

GA runs, the actual numbers of new string evaluations are counted as the GA run proceeds. The GA run is terminated when the maximum number of 200,000 evaluations are performed. The number of generations required to produce 200,000 new strings are recorded.

Crossover occurs between two selected parent strings with a probability of crossover, p_c . Probabilities of crossover $p_c=0.25$, $p_c=0.5$, $p_c=0.75$ and $p_c=1.0$ are considered. A value of $p_c=0.5$ and a population size of $N=200$ will result in approximately $(p_c)(N)=100$ of the 200 coded strings in the new population being created by crossing over two strings from the old population. The other 100 or so strings pass to the new generation without being crossed over. GA researchers (Goldberg, 1989) suggest good performance of the GA may be obtained using high crossover probabilities ($p_c=0.5$ to 1.0). For a value of $p_c=1.0$, every string selected from the old population is modified by crossover in forming the new population.

GA researchers (Goldberg, 1989) suggest bit-wise mutations should occur with low probability ($p_m=0.001$ to 0.05). Mutation probabilities $p_m=0.001$, $p_m=0.005$ and $p_m=0.01$ are considered in the following GA runs. A value of $p_m=0.005$ implies 1 bit in every 200 bits crossed over is mutated. Since the string length is 84 bits for the New York problem, about 4 bits will be mutated from 10 strings crossed over to form a new population. A high probability of crossover ($p_c=1.0$) and relatively frequent bit-wise mutations ($p_m=0.005$) are employed for the GA runs in Section 8.4.

The random number generator seed is usually held constant for a series of GA runs for a fair comparison. The seed is an arbitrarily chosen integer that initiates a unique sequence of random numbers. The same seed produces the same sequence of random numbers and generates the same starting population of coded strings for a given population size. This is useful for the comparison of the performance of various GA formulations and combinations of GA parameters. The mechanics of the random number generator were described in Section 5.4.1.

8.4 Establishing a Penalty Multiplier (GA Runs NY1-NY10)

Infeasible solutions are expected to play an important role in the GA search. The optimal solution lies close to the boundary between feasible and infeasible solutions. The penalty function approach should allow the GA search to approach the optimum solution from both the feasible and infeasible regions of the solution space. Ideally, the lowest cost solution determined by the GA search will be a feasible solution. Therefore, the penalty multiplier should produce penalty costs such that near-optimal infeasible solutions are just a little more expensive than the optimal (best feasible) solution. Since the optimal solution is not known, some trial and error adjustment of the penalty multiplier is usually necessary. The purpose of GA runs NY1, NY2, ..., NY10 is to establish an appropriate penalty multiplier, k . The penalty multiplier sets the severity of the penalties. The GA runs are summarised as follows:

- NY1-NY5 Improved GA formulation with varying penalty multiplier k =\$5million/ft, \$10million/ft, \$20million/ft, \$30million/ft and \$40million/ft for five identical GA parameter sets with $N=200$, $p_c=1.0$ and $p_m=0.005$ (and with random number $seed=10$)
- NY6-NY10 Improved GA with varying penalty multiplier for five identical GA parameter sets as for GA runs NY1-NY5 (but with $seed=20$)

In this study, the performance of a GA run is evaluated in terms of a number of measures. The best costs and average costs of each generation of the GA run are recorded as the run proceeds. The lowest solution cost generated by the GA run and the number of generations and evaluations to achieve this solution, and the lowest average generation cost achieved during the GA run are recorded. DeJong (1975) used two performance measures, online and offline performance, to evaluate the effectiveness of his GA models applied to various solution spaces. The online performance is described as an average of all function evaluations up to the current time and offline performance as the average of the best function evaluations to the current time (Goldberg, 1989). In this study, ultimate offline performance is the average of all best generation costs at the termination of the GA run and the ultimate online performance is the average of all solution costs (excluding elite population solutions) at the end of the GA run.

Ultimately, the success of the GA run is measured by the quality of the best solution produced. The lowest cost feasible and infeasible GA designs determined by GA runs in this chapter, designated GA(1)-GA(8), were shown in Figure 8.2 and are presented in detail in Section 8.6.

8.4.1 Results of GA runs NY1-NY10

The GA parameters and features of the improved GA formulation and the results of GA runs NY1-NY5 are given in Table 8.5. Table 8.6 shows the results of the GA runs NY6-NY10.

Infeasible designs are more prominent in the GA search when the penalty multiplier is smaller. The GA runs NY1 ($k=\$5$ million/ft) and NY2 ($k=\$10$ million/ft) with the smaller penalty multipliers generate many low cost, but slightly infeasible solutions and the best cost solutions are infeasible. The GA run NY1 determined a least cost (infeasible) solution for \$38.497 million (\$37.130m for tunnel construction costs and a penalty cost of \$1.37m as a result of a hydraulic head deficiency of 0.27ft at the critical node). The GA run NY2 determined a least cost infeasible solution for \$38.845m (\$38.638m for tunnel costs and a penalty cost of \$0.207m indicating a maximum hydraulic head violation of 0.04ft). The GA runs NY6 ($k=\$5$ million/ft) and NY7 ($k=\$10$ million/ft) with a different starting population of strings (different random number seed) reach the same least cost (infeasible) solutions as the corresponding GA runs NY1 and NY2.

The costs of several infeasible solutions determined in GA runs NY1, NY2, NY6 and NY7 are less than the cost of the best known feasible design GA(1) of \$38.796m (Section 8.6). In such cases, it is worthwhile observing the least cost *feasible* designs identified by the GA run. The GA run NY1 found the local optimum design GA(2) for \$39.062 million after 164,800 evaluations (824 generations) and GA run NY2 found design GA(2) after 115,800 evaluations (579 generations). The GA runs NY6 and NY7 also located design GA(2). The GA runs with lower penalty multipliers are useful as they demonstrate the value in some infeasible designs. Some of the most valuable infeasible designs produced are GA(6), GA(7) and GA(8).

The offline performance (average of the best of generation costs) after the completion of 1,000 generations (or 200,000 evaluations) for GA runs NY1-NY5 (in Table 8.5) is slightly less for NY1 as it generates many low cost infeasible solutions, but is not significantly different for the increasing penalty multipliers of GA runs NY2-NY4. As expected, the lowest average generation cost and ultimate online performance (average of all solution costs) increases as the penalty multiplier increases. In general, the lowest average generation cost is achieved in the last quarter of the GA runs NY1-NY5 (after 750 generations when the exponent in the scaled fitness function becomes $n=4$), indicating the GA populations are improving up to and beyond 750 generations (or 150,000 evaluations).

Table 8.5 Improved GA runs NY1-NY5 with varying penalty multiplier

GA SEARCH FORMULATION AND PARAMETERS FOR GA RUNS NY1-NY5					
GA RUNS	NY1	NY2	NY3	NY4	NY5
Population size, N	200	200	200	200	200
Maximum number of evaluations	200,000	200,000	200,000	200,000	200,000
Maximum number of generations	1,000	1,000	1,000	1,000	1,000
Probability of crossover, p_c	1.0	1.0	1.0	1.0	1.0
Probability of random bit-wise mutation, p_m	0.005	0.005	0.005	0.005	0.005
Probability of creeping mutation, p_a	0.04	0.04	0.04	0.04	0.04
Probability of creeping down, p_d	0.6	0.6	0.6	0.6	0.6
Random number generator seed	10	10	10	10	10
Penalty multiplier, k (\$million/ft)	5.0	10.0	20.0	30.0	40.0
Coding scheme	Gray	Gray	Gray	Gray	Gray
Fitness	Scaled	Scaled	Scaled	Scaled	Scaled
Elite population size, N'	10	10	10	10	10
Probability of an elite mate, p_e	0.01	0.01	0.01	0.01	0.01
GA SEARCH RESULTS FOR GA RUNS NY1-NY5					
GA RUNS	NY1	NY2	NY3	NY4	NY5
Lowest solution cost (incl. penalties) (\$million)	38.497 [†]	38.845 [†]	39.682	39.062	39.062
- after - generation	947	915	412	437	345
- after - evaluation	189,400	183,000	82,400	87,400	69,000
Lowest average generation cost (\$million)	47.8	48.8	52.9	53.1	55.6
- after - generation	811	964	754	992	879
- after - evaluation	162,200	192,800	150,800	198,400	175,800
Ultimate offline performance (\$million)	42.33	43.66	43.68	43.46	43.39
Ultimate online performance (\$million)	60.3	66.5	75.0	83.4	91.1
GA designs generated (Table 8.13 and Figure 8.2)	2,3,5,6 [†] ,8 [†]	2,3	-	2,3	2,3
Average no. of infeasible solutions / population	119.6	73.0	52.3	45.5	40.3
Maximum no. of infeasible solutions	162	109	102*	102*	102*
Minimum no. of infeasible solutions	11	25	5	16	8

[†] infeasible designs

* number of infeasible solutions in starting population

Table 8.6 Improved GA runs NY6-NY10 with varying penalty multipliers and new random number generator seed

GA SEARCH FORMULATION AND PARAMETERS# FOR GA RUNS NY6-NY10					
GA RUNS	NY6	NY7	NY8	NY9	NY10
Random number generator seed	20	20	20	20	20
Penalty multiplier, k (\$million/ft)	5.0	10.0	20.0	30.0	40.0
GA SEARCH RESULTS FOR GA RUNS NY6-NY10					
Lowest solution cost (incl. penalties) (\$million)	38.497 [†]	38.845 [†]	38.796	39.062	39.062
- after - generation	897	826	347	431	334
- after - evaluation	179,400	165,200	69,400	86,200	66,800
Lowest average generation cost (\$million)	46.5	48.8	51.6	52.5	55.1
- after - generation	769	790	963	799	935
- after - evaluation	153,800	158,000	192,600	159,800	187,000
Ultimate offline performance (\$million)	42.41	43.42	43.66	43.18	43.56
Ultimate online performance (\$million)	60.9	66.1	75.0	82.5	90.4
GA designs generated (Table 8.13 and Figure 8.2)	2,6 [†] ,7 [†] ,8 [†]	2,3	1,4,5,6 [†]	2,3	2,3
Average no. of infeasible solutions / population	120.5	73.8	44.5	44.8	40.8
Maximum no. of infeasible solutions	170	110	101*	101*	101*
Minimum no. of infeasible solutions	32	27	13	14	9

GA parameter sets identical to corresponding GA runs NY1-NY5, except for seed

[†] infeasible designs

* number of infeasible solutions in starting population

Figure 8.3 shows the best of generation costs for GA runs NY1 (k =\$5million/ft) and NY5 (k =\$40million/ft). Both GA runs converge quickly on the best regions of the solution space, although the GA run NY1 converges slightly faster to lower total costs (including penalties). Figure 8.4 plots the average generation costs for the GA runs NY1, NY3 and NY5. The average generation costs increase and are less stable for higher penalties.

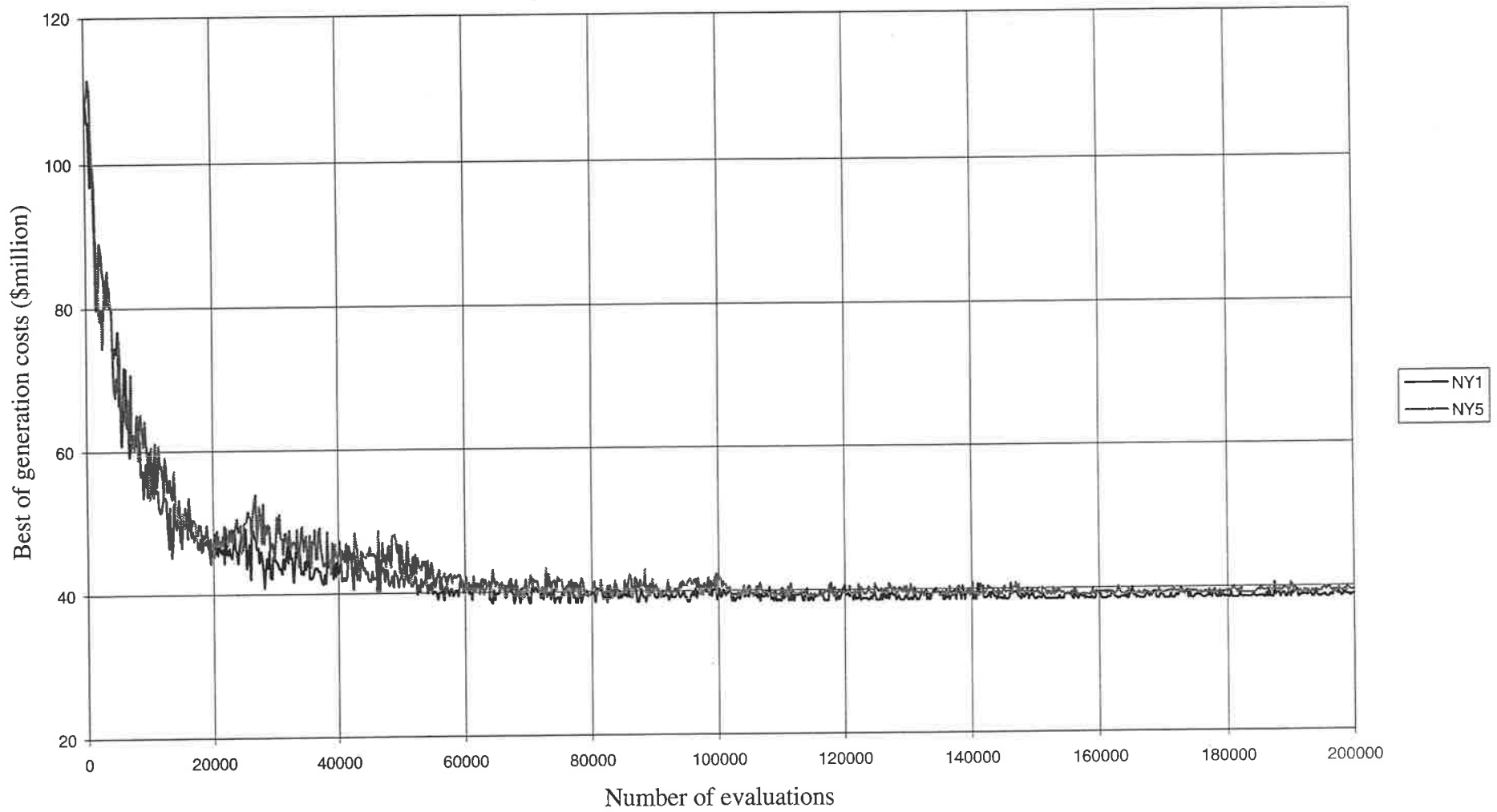


Figure 8.3 Best of generation costs for GA runs NY1 ($k = \$5\text{m/ft}$) and NY5 ($k = \40m/ft)

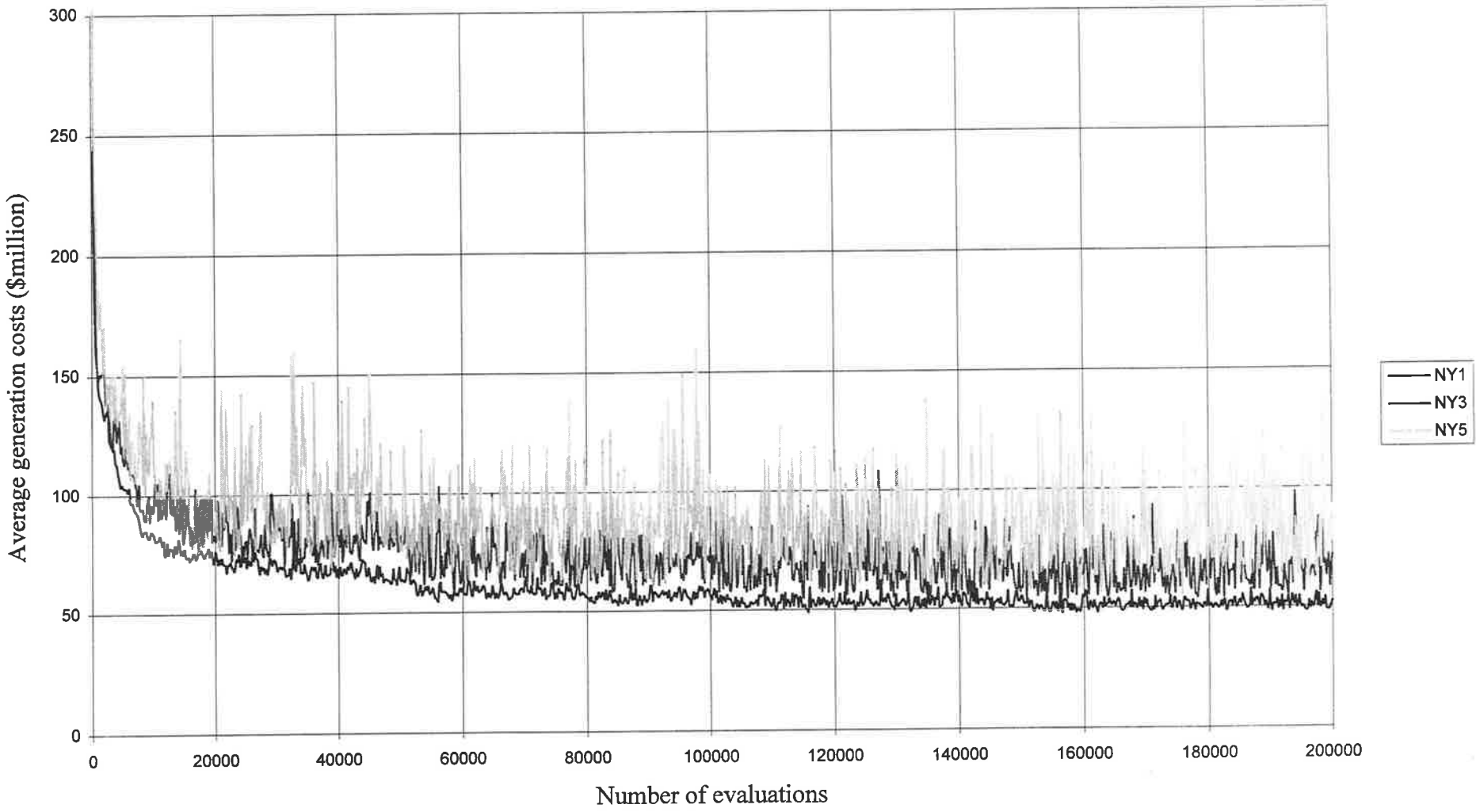


Figure 8.4 Average generation costs for GA runs NY1, NY3 and NY5

The GA run NY1 moves further into the best infeasible regions of the solution space as indicated by the number of infeasible solutions per generation in Figure 8.5. The GA run NY1 has on average 120 and up to 160 infeasible solutions in any population of 200 solutions, compared to an average of 40 infeasible solutions per population for GA run NY5 with the larger penalty multipliers. The high numbers of infeasible solutions present in the GA runs (even for high values of the penalty coefficient) underline the important role of infeasible solutions in the search.

Since the five GA runs NY1-NY5 are initiated with the same random number generator seed, the starting populations of coded strings are identical and each starting population contains 102 infeasible solutions. The starting populations with the different seed for GA runs NY6-NY10 contain 101 infeasible solutions. The randomly generated starting populations of solutions for the New York problem was consistently formed with about an equal share of feasible and infeasible solutions. The size of the feasible and infeasible regions of the solution space may be approximately equal.

Most of the infeasible solutions in the starting population are likely to be very infeasible and the numbers of infeasible solutions in the subsequent early populations suddenly drops from 100 in a population size of 200 to only 10 or 20 (for all values of k). The proportion of infeasible solutions then gradually increases again toward the typical number for the given value of k . Further experimentation may consider some variation of the penalty multiplier value as the GA run proceeds. Initially, the value of the penalty multiplier should be low to avoid the immediate culling of large numbers of infeasible designs, but perhaps should increase as the GA run develops to ensure convergence to feasible regions of solutions.

The performances of GA runs NY6-NY10 are similar to corresponding GA runs NY1-NY5. The significant difference is the improved GA run NY8 (with $k=\$20$ million/ft) achieves a lower cost feasible design, which is the lowest cost design identified by any of the GA runs. The design is GA(1) with a tunnel construction cost of \$38.796 million (presented in Section 8.6).

The design GA(1) is determined by GA run NY8 after only 69,400 solution evaluations of the maximum 200,000. The other GA runs NY6, NY7, NY9 and NY10 all find the second best known feasible solution GA(2) with cost \$39.062 million, although the least cost solutions (including penalty costs) determined by GA runs NY6 ($k=\$5$ million/ft) and NY7 ($k=\$10$ million/ft) are infeasible solutions.

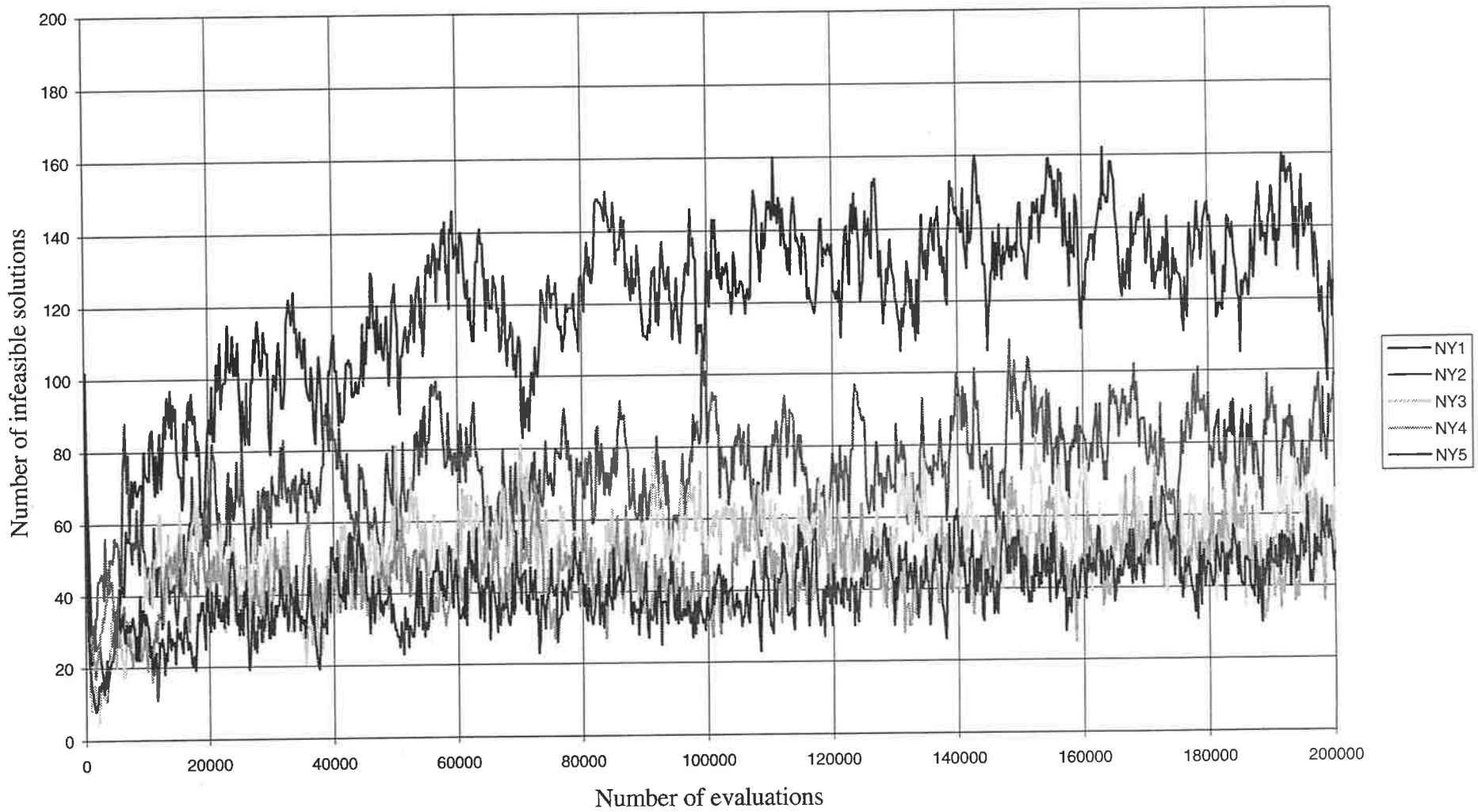


Figure 8.5 Fluctuations of number of infeasible solutions per population for GA runs NY1-NY5

The improved GA runs NY1-NY10 with varying penalty multipliers demonstrate a high level of performance regardless of the severity of the penalties for the penalty multiplier values tested. For GA runs NY1-NY10, the number of infeasible network solutions present in each new generation and the feasibility of the lowest cost solutions identified by the GA search are indications of the suitability of the penalty multiplier. The lower penalties ($k=\$5\text{million/ft}$ and $\$10\text{million/ft}$) produce a number of valuable just-infeasible solutions and the low cost feasible solution GA(2), which is one of two distinct local optima found by the GA search. When larger penalty multipliers ($k=\$30\text{million/ft}$) are applied to infeasible solutions, the local optimal solutions GA(1) and GA(2) are the best cost solutions and near-optimal infeasible solutions are a little more expensive than the feasible optima.

A penalty multiplier of $k=\$30\text{million/ft}$ (GA runs NY4 and NY9) is used for the GA runs in Section 8.5 of this study. For a value of $k=\$30\text{million/ft}$, the lowest cost designs generated are feasible designs, although some infeasible designs are still amongst the best designs and there is a reasonable number of infeasible solutions generated in a population.

8.5 Performance of the Improved GA Formulation (GA Runs NY11-NY69)

The series of GA runs designated NY11-NY19, NY21-NY29, ..., NY61-NY69 are intended to measure the effectiveness of the various features of the improved GA formulation. The GA runs are summarised as follows:

- NY11-NY19 Improved GA incorporating the elitism strategy, Gray codes, fitness scaling and creeping mutations for nine GA parameter sets that are variations of the original GA parameters $N=200$, $p_c=1.0$ and $p_m=0.005$ (with $seed=50$ and $k=\$30\text{million/ft}$)
- NY21-NY29 Improved GA (NY11-NY19) for the nine corresponding GA parameter sets (used for NY11-NY19) but with no elitism strategy ($p_e=0.0$)
- NY31-NY39 Improved GA (NY11-NY19) for the nine corresponding GA parameter sets (NY11-NY19) but using substrings of binary codes (not Gray codes)
- NY41-NY49 Improved GA (NY11-NY19) for the nine corresponding GA parameter sets (NY11-NY19) but with no fitness scaling
- NY51-NY59 Improved GA (NY11-NY19) for the nine corresponding GA parameter sets (NY11-NY19) but with no creeping mutation ($p_a=0.0$)
- NY61-NY69 Traditional GA formulation (with $p_e=0.0$, binary codes, raw fitness values and $p_a=0.0$) for the nine corresponding GA parameter sets (used for NY11-NY19)

The improved GA formulation for pipe network optimisation is applied to the New York tunnels problem for nine alternative GA parameter sets and the GA runs are labelled NY11-NY19. The GA parameter sets are summarised in Table 8.7. The primary GA parameters (used for GA run NY12) are chosen to be $N=200$, $p_c=1.0$ and $p_m=0.005$. The population size, crossover rate and bit-wise mutation rate are systematically varied from the primary GA parameter values to form the nine GA parameter sets. Population size is varied from $N=100$ to 1,000 members, probability of crossover is varied from $p_c=0.25$ to 1.0 and mutation rates of $p_m=0.001$, 0.005 and 0.01 are considered. The random number seed is held constant for all the GA runs. Since, it is more desirable to produce a least cost solution which is feasible, a penalty multiplier value of $k=\$30\text{million/ft}$ is used for the GA runs NY11-NY69 (see Section 8.4). The performance of the improved GA formulation (GA runs NY11-NY19), the traditional GA formulation (GA runs NY61-NY69) and the various intermediate GA formulations are assessed across the nine selected GA parameter sets.

8.5.1 Results of GA runs NY11-NY69

The results for the improved GA runs NY11-NY69 are summarised in Tables 8.7-8.12. The GA run NY12 identifies the solution GA(1) after 459 generations (91,800 evaluations). Of the total of 64 GA runs performed in this GA study of the New York tunnels problem, the best known feasible solution GA(1) for \$38.796m (presented in Section 8.6) was achieved only 3 times in GA runs NY8 (improved GA with $k=\$20\text{million/ft}$), NY12 (improved GA with $k=\$30\text{million/ft}$) and NY37 (improved GA with substrings of binary codes and $p_c=0.75$). By comparison, the competing local optimum solution GA(2) for \$39.062m was achieved on 39 occasions. The improved GA runs NY11-NY19 (with elitism) determined solution GA(1) on one occasion with $N=200$, achieved solution GA(2) six times and twice ($N=500$ and $p_c=0.25$) failed to determine either GA(1) or GA(2).

Table 8.7 Improved GA runs NY11-NY19

GA SEARCH FORMULATION AND PARAMETERS FOR GA RUNS NY11-NY19									
GA RUNS	NY11	NY12	NY13	NY14	NY15	NY16	NY17	NY18	NY19
Population size, N	100	200	500	1,000	200	200	200	200	200
Maximum evaluations	200,000	200,000	200,000	200,000	200,000	200,000	200,000	200,000	200,000
Max. generations required	2,000	1,000	400	200	1,137	1,089	1,044	1,000	1,000
Prob. crossover, p_c	1.0	1.0	1.0	1.0	0.25	0.5	0.75	1.0	1.0
Prob. bit mutation, p_m	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.001	0.01
Prob. creep mutation, p_a	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Prob. creep down, p_d	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
Random number seed	50	50	50	50	50	50	50	50	50
Penalty factor, k (\$m/ft)	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
Coding scheme	Gray	Gray	Gray	Gray	Gray	Gray	Gray	Gray	Gray
Fitness	Scaled	Scaled	Scaled	Scaled	Scaled	Scaled	Scaled	Scaled	Scaled
Elite population size, N'	10	10	10	10	10	10	10	10	10
Prob. elite mate, p_e	0.02	0.01	0.004	0.002	0.01	0.01	0.01	0.01	0.01
GA SEARCH RESULTS FOR GA RUNS NY11-NY19									
GA RUNS	NY11	NY12	NY13	NY14	NY15	NY16	NY17	NY18	NY19
Lowest cost (\$million)	39.062	38.796	39.694	39.062	40.294	39.062	39.062	39.062	39.062
- after - generation	1,005	459	281	178	304	315	617	366	555
- after - evaluation	100,500	91,800	140,500	178,000	53,400	57,800	118,200	73,200	111,000
Lowest avg gen cost	48.8	52.4	60.6	64.5	48.0	49.6	50.5	46.2	62.2
- after - generation	1,534	800	298	160	1,019	796	1,034	904	901
- after - evaluation	153,400	160,000	149,000	160,000	179,300	146,300	198,200	180,800	180,200
Offline performance (\$m)	43.5	43.8	47.0	50.5	44.3	44.3	43.6	42.6	45.4
Online performance (\$m)	81.7	82.7	89.2	98.1	66.3	72.5	77.7	64.5	104.6
GA designs (Table 8.13)	2,3	1,4,5	-	2,3	-	2,3	2,3	2,3	2,3

Table 8.8 Improved GA runs NY21-NY29 (without elitism)

GA SEARCH FORMULATION AND PARAMETERS# FOR GA RUNS NY21-NY29									
GA RUNS‡	NY21 <i>N</i> =100	NY22 <i>N</i> =200	NY23 <i>N</i> =500	NY24 <i>N</i> =1,000	NY25 <i>p_c</i> =0.25	NY26 <i>p_c</i> =0.5	NY27 <i>p_c</i> =0.75	NY28 <i>p_m</i> =0.001	NY29 <i>p_m</i> =0.01
Max. generations required	2,000	1,000	400	200	1,137	1,088	1,044	1,000	1,000
Prob. elite mate, <i>p_e</i>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GA SEARCH RESULTS FOR GA RUNS NY21-NY29									
Lowest cost (\$million)	39.062	39.062	39.062	39.062	39.062	39.062	39.062	39.062	39.725
- after - generation	1,033	773	273	163	875	755	596	414	885
- after - evaluation	103,300	154,600	136,500	163,000	153,800	139,100	114,200	82,800	177,000
Lowest avg gen cost	51.4	55.1	62.0	64.9	47.6	48.1	51.2	47.1	62.3
- after - generation	1,760	735	288	194	894	1,024	906	925	978
- after - evaluation	176,000	147,000	144,000	194,000	157,100	188,400	173,500	185,000	195,600
Offline performance (\$m)	49.1	47.4	51.0	52.4	46.2	45.1	47.8	44.5	52.3
Online performance (\$m)	85.6	86.1	93.9	99.4	67.2	72.3	80.7	66.0	106.3
GA designs (Table 8.13)	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	-

GA parameter sets are identical to the corresponding GA runs NY11-NY19, with no elitism

‡ unless otherwise specified *N*=200, *p_c*=1.0, *p_m*=0.005

The GA runs NY21-NY29 consider the performance of the improved GA formulation without the elitism concept described in Section 8.3.8. The GA runs NY21-NY29 are identical to the corresponding GA runs NY11-NY19 in all other respects. The results of the GA runs NY21-NY29 are given in Table 8.8. The GA runs generate solution GA(2) and GA(3) in eight of the nine GA runs. The optimum solution GA(1) is not identified and the GA run NY29 with the relatively high bit-wise mutation rate *p_m*=0.01 does not achieve GA(1) or GA(2).

The variation of best of generation costs and average generation costs for the improved GA run NY13 with elitism and NY23 without elitism are compared in Figures 8.6 and 8.7 respectively. The decision to compare GA runs NY13 and NY23 with a population size *N*=500 is not particularly favourable to the improved GA run NY13 which fails to determine either solution GA(1) or GA(2) under these circumstances. The improved GA run NY13 with elitism converges faster but after about 100,000 evaluations the improved GA run NY23 without elitism is achieving similarly priced solutions and there is little improvement shown by either GA run beyond this point.

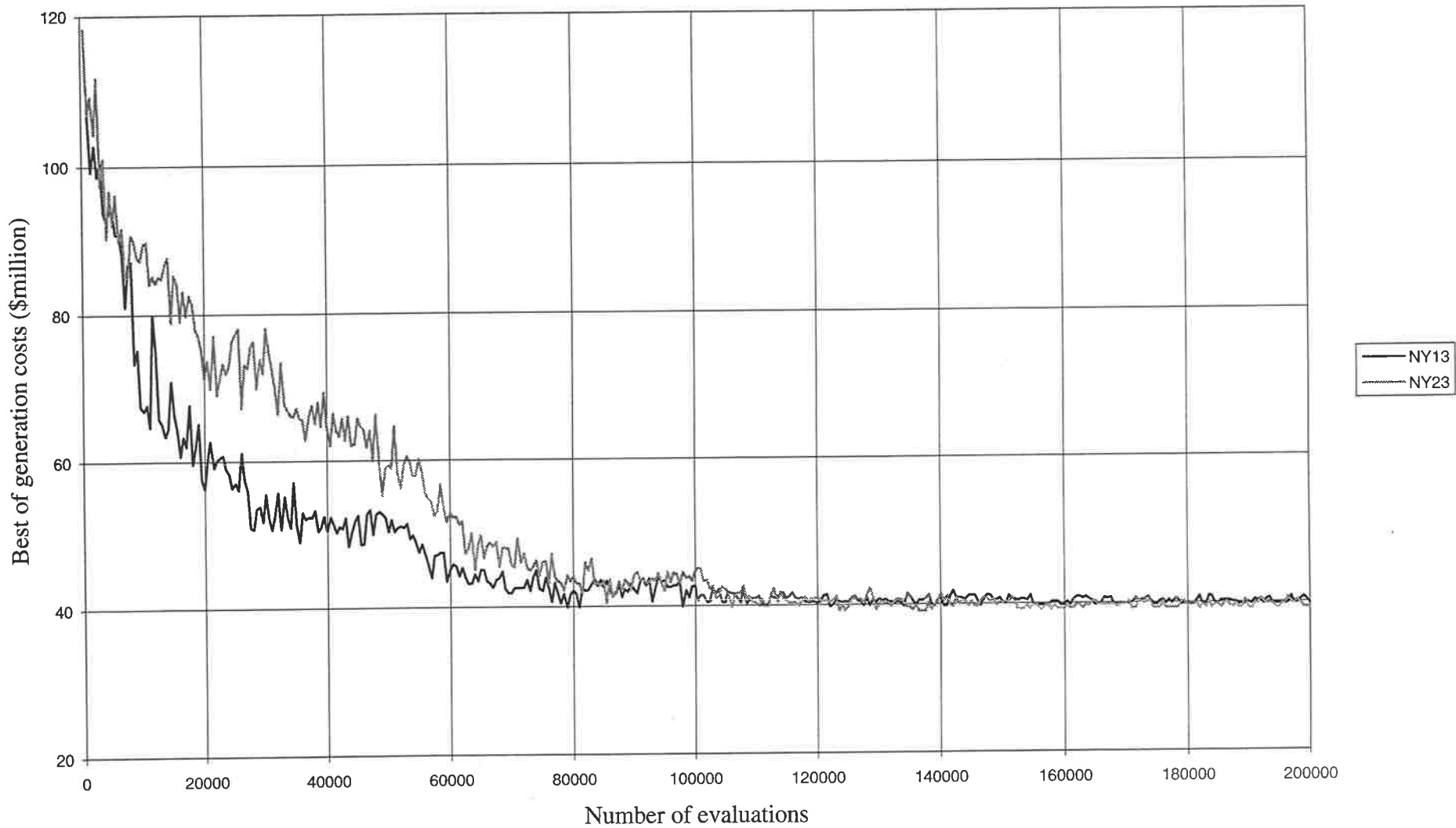


Figure 8.6 Best of generation costs for GA runs NY13 (Improved GA with $N=500$) and GA run NY23 (No elitism)

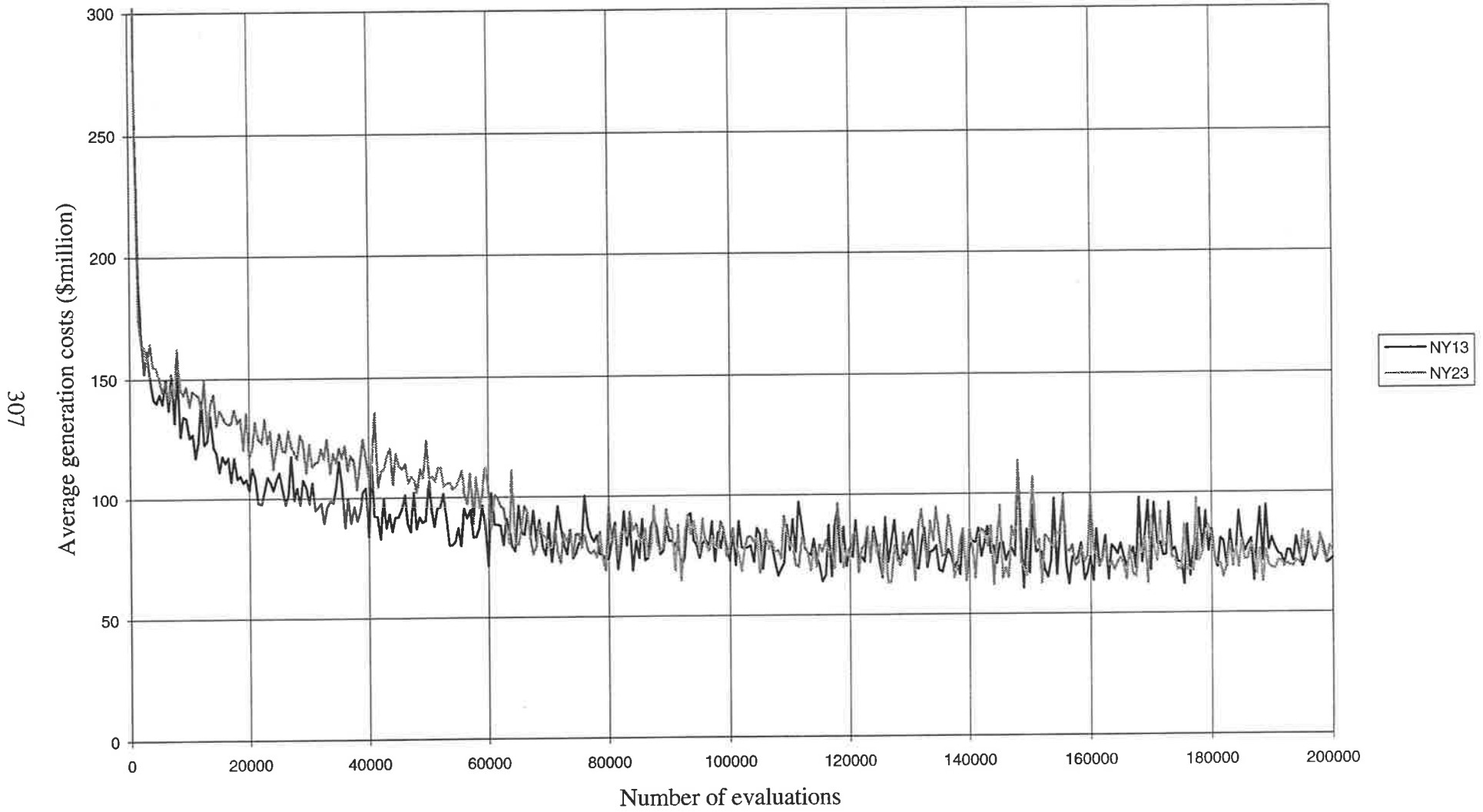


Figure 8.7 Average generation costs for GA runs NY13 (Improved GA with $N=500$) and NY23 (No elitism)

Table 8.9 Improved GA runs NY31-NY39 (with substrings of binary codes)

GA SEARCH FORMULATION AND PARAMETERS# FOR GA RUNS NY31-NY39									
GA RUNS‡	NY31 <i>N</i> =100	NY32 <i>N</i> =200	NY33 <i>N</i> =500	NY34 <i>N</i> =1,000	NY35 <i>p_c</i> =0.25	NY36 <i>p_c</i> =0.5	NY37 <i>p_c</i> =0.75	NY38 <i>p_m</i> =0.001	NY39 <i>p_m</i> =0.01
Max. generations required	2,000	1,000	400	200	1,136	1,091	1,043	1,000	1,000
Coding scheme	Binary	Binary	Binary	Binary	Binary	Binary	Binary	Binary	Binary
GA SEARCH RESULTS FOR GA RUNS NY31-NY39									
Lowest cost (\$million)	39.062	39.062	39.062	39.062	39.062	39.062	38.796	39.682	39.062
- after - generation	1,681	557	291	194	977	301	268	301	759
- after - evaluation	168,100	111,400	145,500	194,000	172,000	55,100	51,400	60,200	151,800
Lowest avg gen cost	50.7	53.9	59.3	68.2	45.7	49.7	52.3	48.9	62.0
- after - generation	1,918	865	368	180	883	318	792	774	833
- after - evaluation	191,800	173,000	184,000	180,000	155,300	58,200	152,000	154,800	166,600
Offline performance (\$m)	43.5	44.4	47.1	51.4	45.3	41.9	42.4	43.3	44.7
Online performance (\$m)	84.6	86.5	92.7	102.4	68.9	71.5	77.0	67.9	106.7
GA designs (Table 8.13)	2,3	2,3	2,3	2	2,3	2,3	1,4,5	-	2

GA parameters are identical to the corresponding GA runs NY11-NY19, with binary codes

‡ unless otherwise specified $N=200$, $p_c=1.0$, $p_m=0.005$

The GA runs NY31-NY39 consider the performance of the improved GA formulation using substrings of binary codes rather than Gray codes. The results of the GA runs are given in Table 8.9. The least cost known feasible solution GA(1) is generated by GA run NY37 with $p_c=0.75$. The local optimum solution GA(2) is determined by seven of the nine GA runs and the GA run NY38 ($p_m=0.01$) does not achieve GA(1) or GA(2).

There is not much difference between the plots of best of generation costs and average generation costs for the improved GA run NY13 using Gray codes and GA run NY23 using binary codes in Figures 8.8 and 8.9. There seems to be little improvement in the GA performance using Gray codes as opposed to binary codes for the New York tunnels problem.

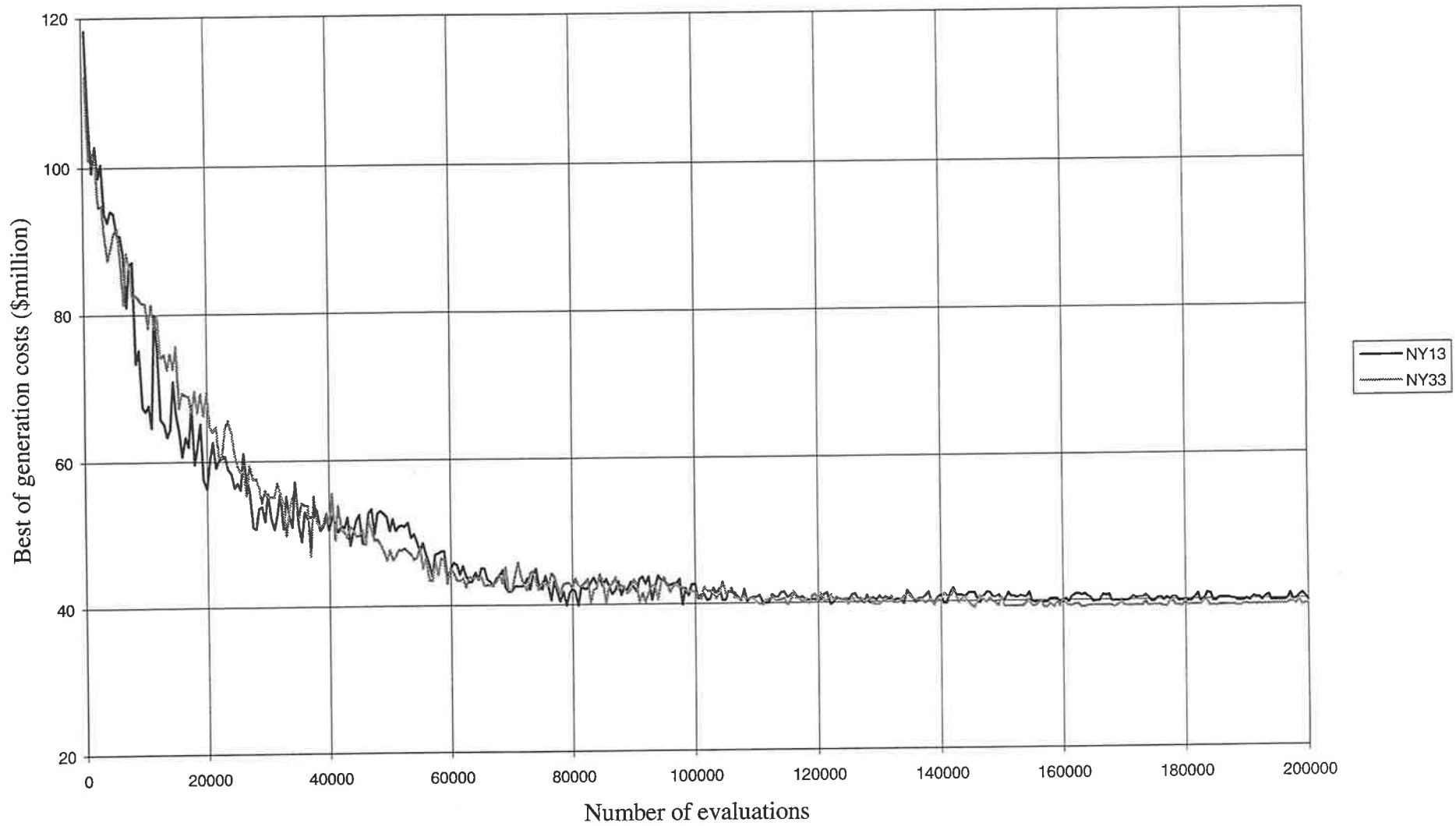


Figure 8.8 Best of generation costs for GA runs NY13 (Improved GA) and NY33 (Binary codes)

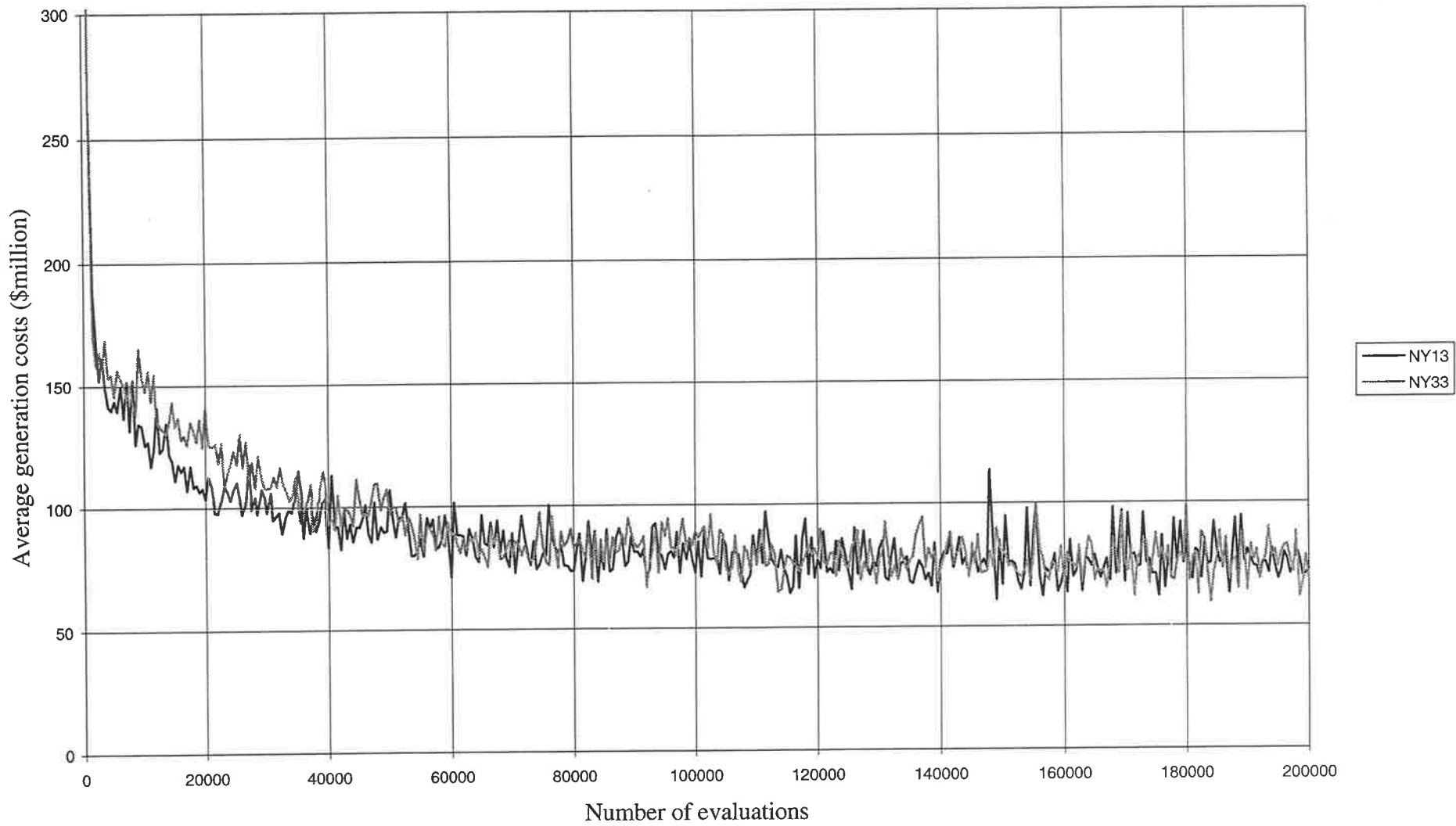


Figure 8.9 Average generation costs for GA runs NY13 (Improved GA) and NY33 (Binary codes)

Table 8.10 Improved GA runs NY41-NY49 (not including fitness scaling)

GA SEARCH FORMULATION AND PARAMETERS# FOR GA RUNS NY41-NY49									
GA RUNS‡	NY41 <i>N</i> =100	NY42 <i>N</i> =200	NY43 <i>N</i> =500	NY44 <i>N</i> =1,000	NY45 <i>p_c</i> =0.25	NY46 <i>p_c</i> =0.5	NY47 <i>p_c</i> =0.75	NY48 <i>p_m</i> =0.001	NY49 <i>p_m</i> =0.01
Max. generations required	2,000	1,000	400	200	1,136	1,087	1,044	1,000	1,000
Fitness	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
GA SEARCH RESULTS FOR GA RUNS NY41-NY49									
Lowest cost (\$million)	39.742	39.284	41.251	41.476	40.294	39.062	39.166	39.062	39.590
- after - generation	1,869	640	392	196	534	555	375	684	885
- after - evaluation	186,900	128,000	196,000	196,000	93,900	102,100	71,900	136,800	177,000
Lowest avg gen cost	58.5	66.2	75.4	80.2	53.1	57.2	60.8	52.8	80.1
- after - generation	1,895	928	278	176	447	593	785	722	895
- after - evaluation	189,500	185,600	139,000	176,000	78,600	109,100	150,400	144,400	179,000
Offline performance (\$m)	45.5	45.9	51.1	54.7	44.5	44.9	44.2	43.4	49.4
Online performance (\$m)	90.7	93.3	98.7	108.9	70.9	77.4	84.9	69.0	118.1
GA designs (Table 8.13)	-	5,6†	-	-	-	2,3	3	2,3	-

GA parameters are identical to corresponding GA runs NY11-NY19, with no fitness scaling

‡ unless otherwise specified $N=200$, $p_c=1.0$, $p_m=0.005$

† infeasible design

The fitness scaling appears to be the most effective feature of the improved GA. The results of the improved GA runs NY41-NY49 without fitness scaling are given in Table 8.10. The solution GA(1) is not identified and the readily generated local optimum solution GA(2) is achieved in only two ($p_c=0.5$ and $p_m=0.001$) of the nine GA runs.

The variations of best generation costs and average generation costs for the improved GA run NY13 with fitness scaling and the corresponding GA run NY43 without fitness scaling in Figures 8.10 and 8.11 demonstrate the significant effect of the increase in the exponent n in the fitness function (Section 8.3.5) after 50,000 evaluations are performed. The GA runs are identical up to this point ($n=1.0$ for less than 50,000 evaluations).

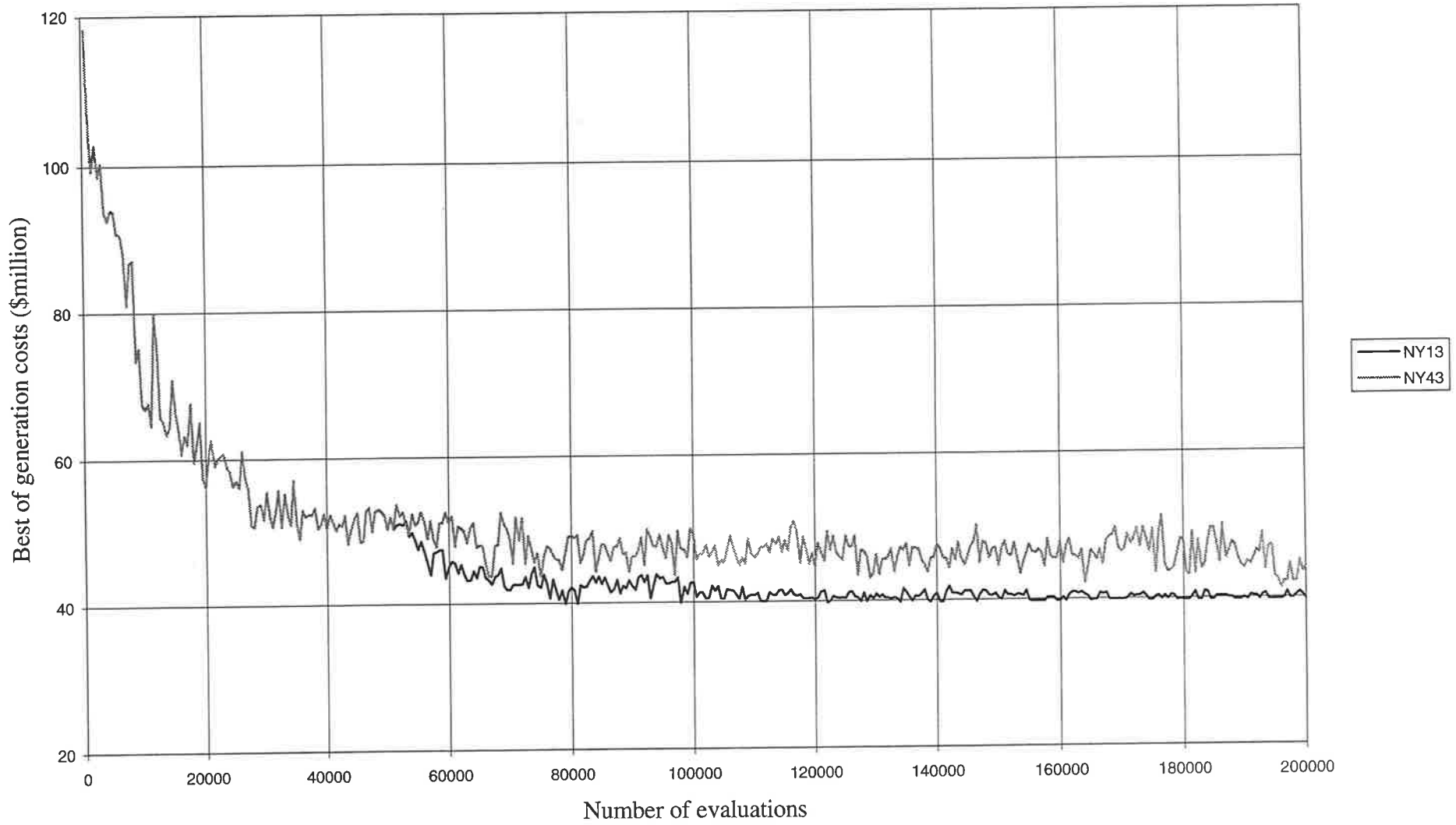


Figure 8.10 Best of generation costs for GA runs NY13 (Improved GA) and NY43 (Raw fitness values)

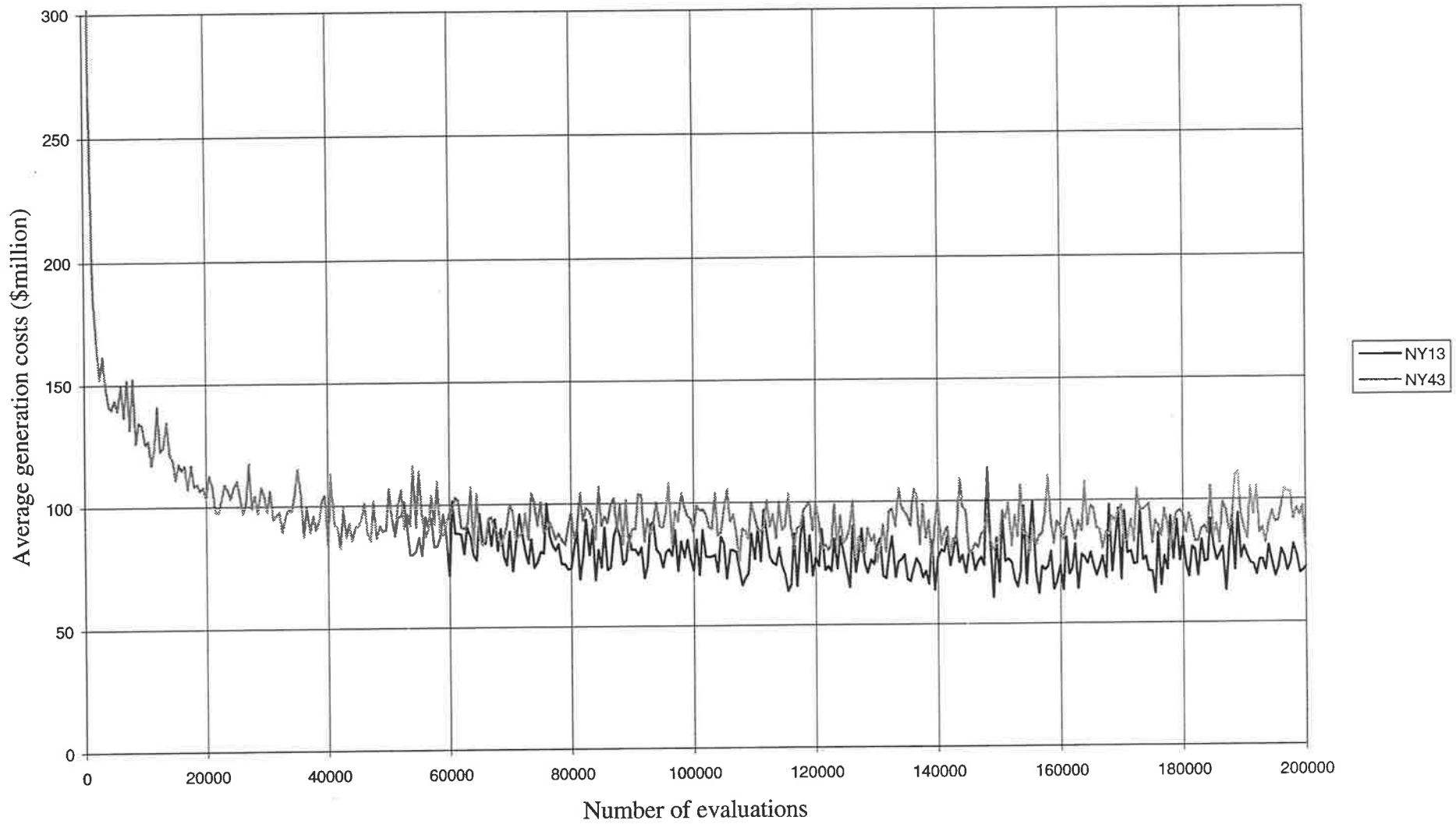


Figure 8.11 Average generation costs for GA runs NY13 (Improved GA) and NY43 (Raw fitness values)

Table 8.11 Improved GA runs NY51-NY59 (not including creeping mutation)

GA SEARCH FORMULATION AND PARAMETERS# FOR GA RUNS NY51-NY59									
GA RUNS‡	NY51 <i>N</i> =100	NY52 <i>N</i> =200	NY53 <i>N</i> =500	NY54 <i>N</i> =1,000	NY55 <i>p_c</i> =0.25	NY56 <i>p_c</i> =0.5	NY57 <i>p_c</i> =0.75	NY58 <i>p_m</i> =0.001	NY59 <i>p_m</i> =0.01
Max. generations required	2,000	1,000	400	200	4,004	1,994	1,338	1,000	1,000
Prob. creep mutation, <i>p_a</i>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GA SEARCH RESULTS FOR GA RUNS NY51-NY59									
Lowest cost (\$million)	39.062	39.062	39.062	39.062	39.062	39.062	39.062	41.242	39.062
- after - generation	1,061	362	276	149	1,942	1,084	680	421	794
- after - evaluation	106,100	72,400	138,000	149,000	97,100	108,800	101,800	84,200	158,800
Lowest avg gen cost	42.9	44.4	47.4	56.9	39.8	41.1	43.0	41.9	55.4
- after - generation	1,701	980	385	136	3,667	1,656	1,926	796	573
- after - evaluation	170,100	196,000	192,500	136,000	183,200	166,200	193,800	159,200	114,600
Offline performance (\$m)	42.5	42.0	45.9	50.8	43.2	41.3	42.6	44.6	45.2
Online performance (\$m)	69.4	70.6	79.4	93.6	51.1	56.4	64.9	54.2	97.2
GA designs (Table 8.13)	2,3	2,3	2,3	2	2,3	2,3	2,3	-	2,3

GA parameters are identical to corresponding GA runs NY11-NY19, with no creep

‡ unless otherwise specified $N=200$, $p_c=1.0$, $p_m=0.005$

The results of GA runs NY51-NY59 without creeping mutations are given in Table 8.11. The solution GA(2) is identified in all but one of the GA runs. The GA run NY58 with $p_m=0.001$ fails to identify GA(1) or GA(2). Figures 8.12 and 8.13 show the comparison of best generation costs and average generation costs respectively for GA run NY13 with creeping mutations and GA run NY53 without creeping mutations. In general, the best generation costs and average generation costs are lower when creep is not used.

The average number of infeasible solutions per population is reduced when the creeping mutation is removed (GA runs NY51-NY59 and traditional GA runs NY61-NY69) from the GA formulation as less strings are disrupted by this operator. Further experimentation would be necessary to determine the value of creeping mutations for the New York tunnels problem. The experimentation may consider the use of creeping mutations with less frequency.

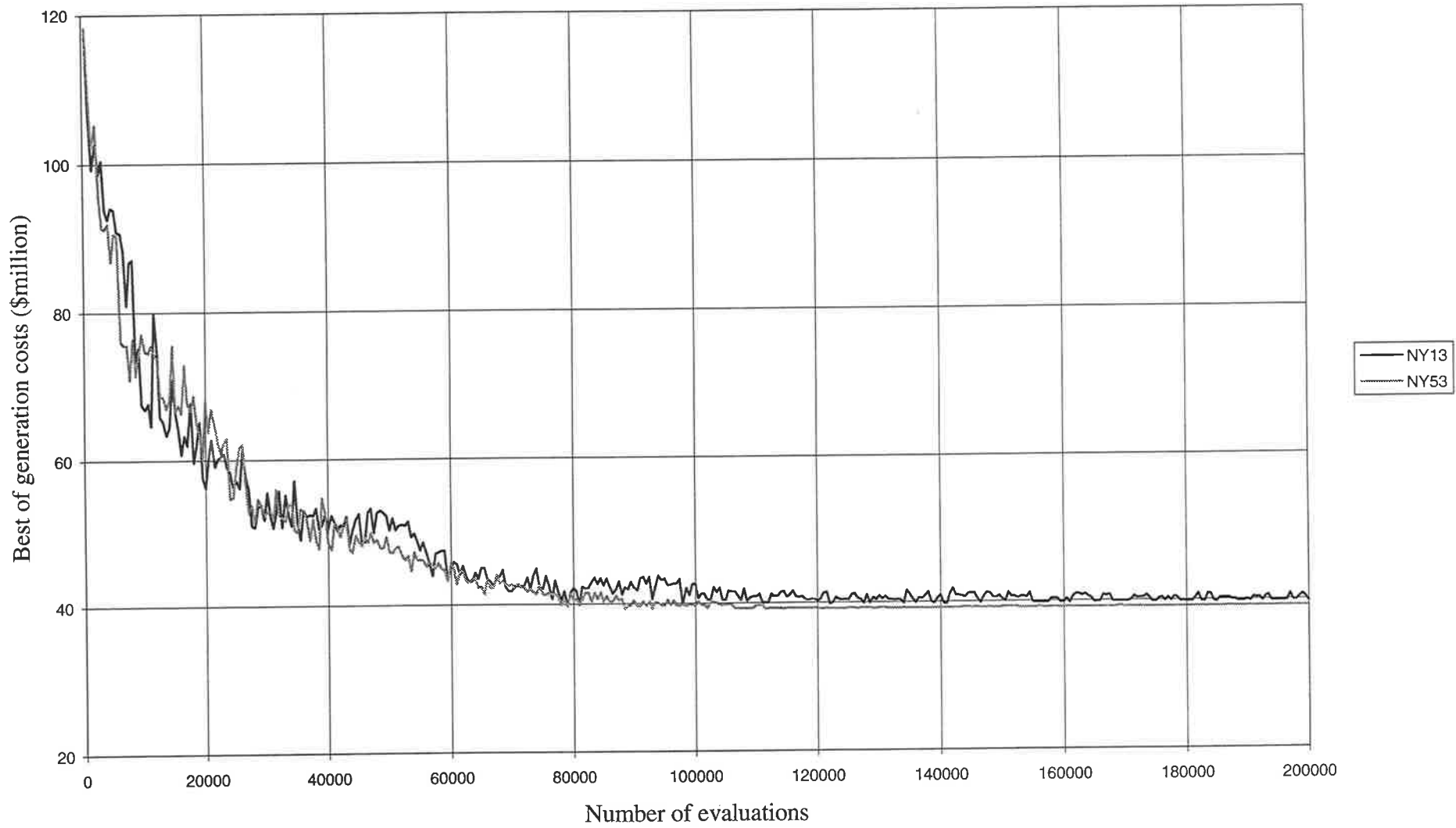


Figure 8.12 Best of generation costs for GA runs NY13 (Improved GA) and NY53 (No creep)

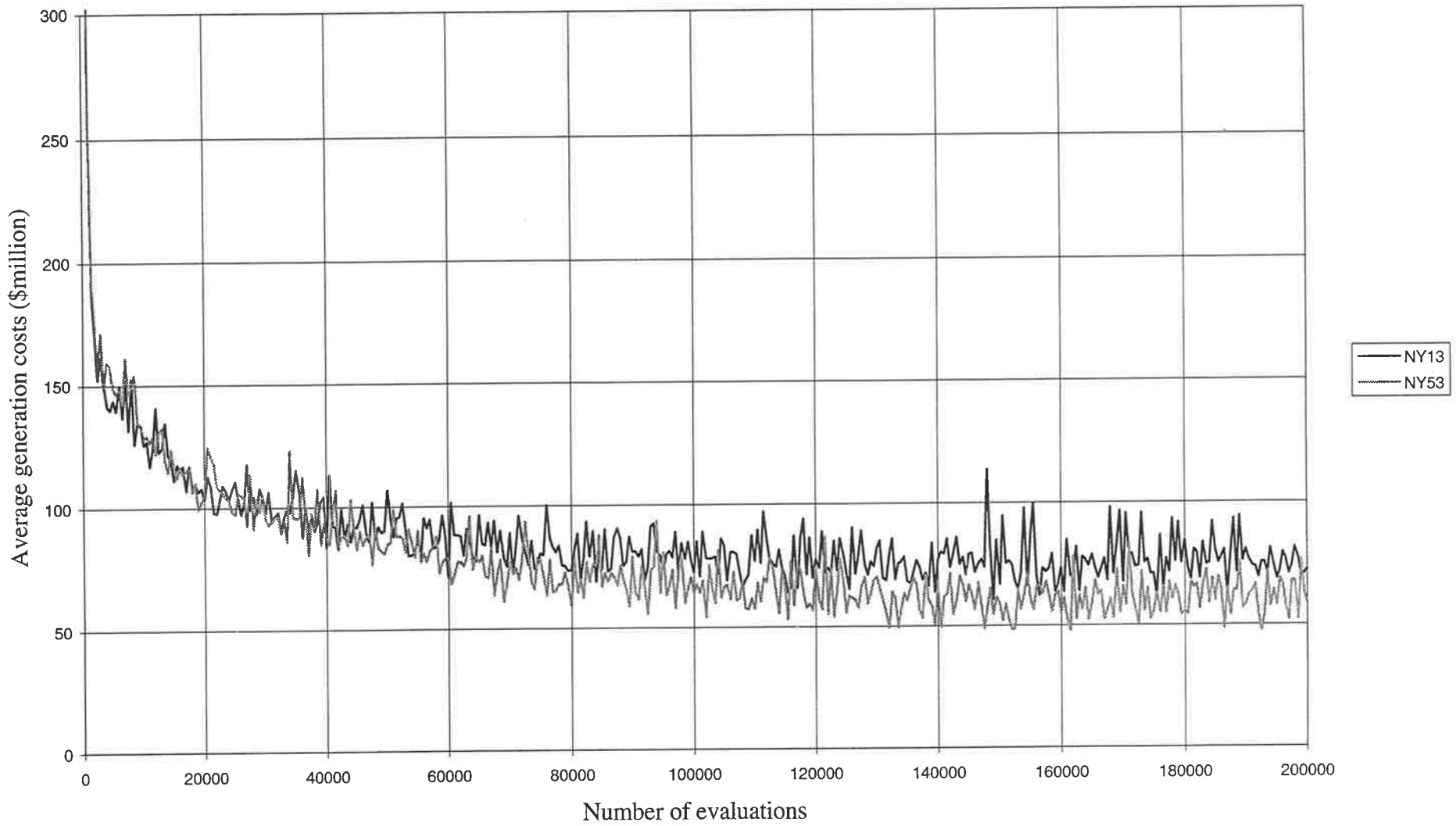


Figure 8.13 Average generation costs for GA runs NY13 (Improved GA) and NY53 (No creep)

Table 8.12 Traditional GA runs NY61-NY69

GA SEARCH FORMULATION AND PARAMETERS# FOR GA RUNS NY61-NY69									
GA RUNS‡	NY61 <i>N</i> =100	NY62 <i>N</i> =200	NY63 <i>N</i> =500	NY64 <i>N</i> =1,000	NY65 <i>p_c</i> =0.25	NY66 <i>p_c</i> =0.5	NY67 <i>p_c</i> =0.75	NY68 <i>p_m</i> =0.001	NY69 <i>p_m</i> =0.01
Max. generations required	2,000	1,000	400	200	3,998	1,999	1,336	1,000	1,000
Prob. creep mutation, <i>p_a</i>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Coding scheme	Binary	Binary	Binary	Binary	Binary	Binary	Binary	Binary	Binary
Fitness	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
Prob. elite mate, <i>p_e</i>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GA SEARCH RESULTS FOR GA RUNS NY61-NY69									
Lowest cost (\$million)	42.631	43.377	41.171	43.209†	43.850	43.713	41.690	41.123	48.466
- after - generation	375	947	316	194	2,910	1,271	1,158	334	641
- after - evaluation	37,500	189,400	158,000	194,000	145,500	127,300	173,400	66,800	128,200
Lowest avg gen cost	59.3	61.5	71.0	72.7	48.8	53.1	56.1	46.0	85.0
- after - generation	267	924	328	146	2,845	800	961	622	698
- after - evaluation	26,700	184,800	164,000	146,000	142,200	80,400	143,800	124,400	139,600
Offline performance (\$m)	52.2	53.8	52.0	55.3	49.0	48.7	50.6	46.1	59.7
Online performance (\$m)	86.0	91.8	98.4	105.2	60.5	71.2	81.4	61.2	116.2

GA parameter sets are identical to the corresponding GA runs NY11-NY19, except that the elements of the improved GA are not included

‡ unless otherwise specified *N*=200, *p_c*=1.0, *p_m*=0.005

† infeasible design (penalty cost included)

The improved GA runs NY11-NY19 consistently determine superior network designs compared to the traditional three-operator GA runs NY61-NY69 summarised in Table 8.12. The traditional GA runs do not achieve solutions GA(1) or GA(2) and in some cases perform very poorly. Figure 8.14 shows the best of generation costs against number of evaluations for the improved GA run NY13 and the equivalent traditional GA run NY63. Figure 8.15 shows the average generation costs against number of evaluations for corresponding GA runs NY13 and NY63. The improved GA formulation exhibits superior performance and generates solutions with significantly lower cost throughout the GA run.

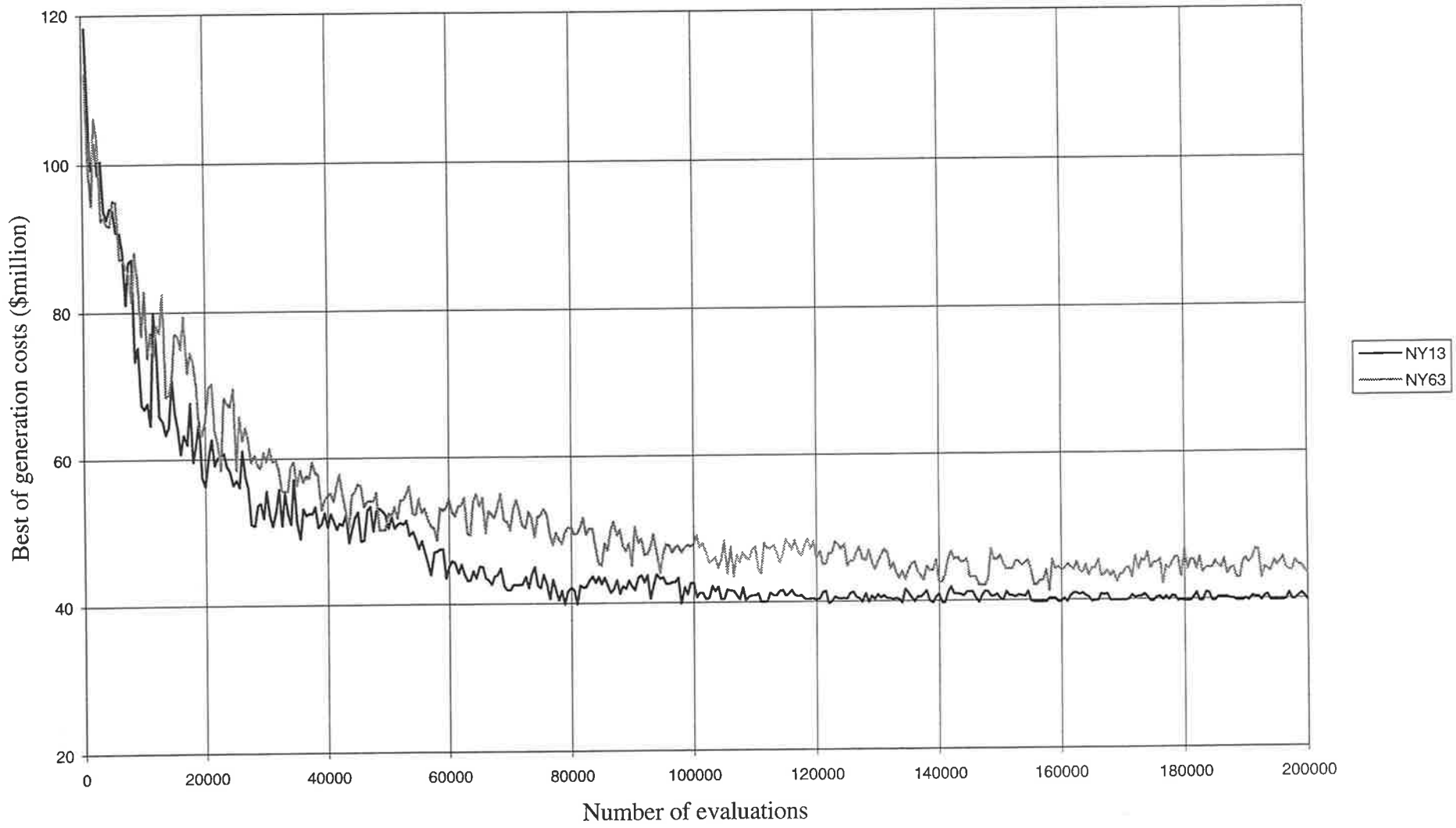


Figure 8.14 Best of generation costs for GA runs NY13 (Improved GA) and NY63 (Traditional GA)

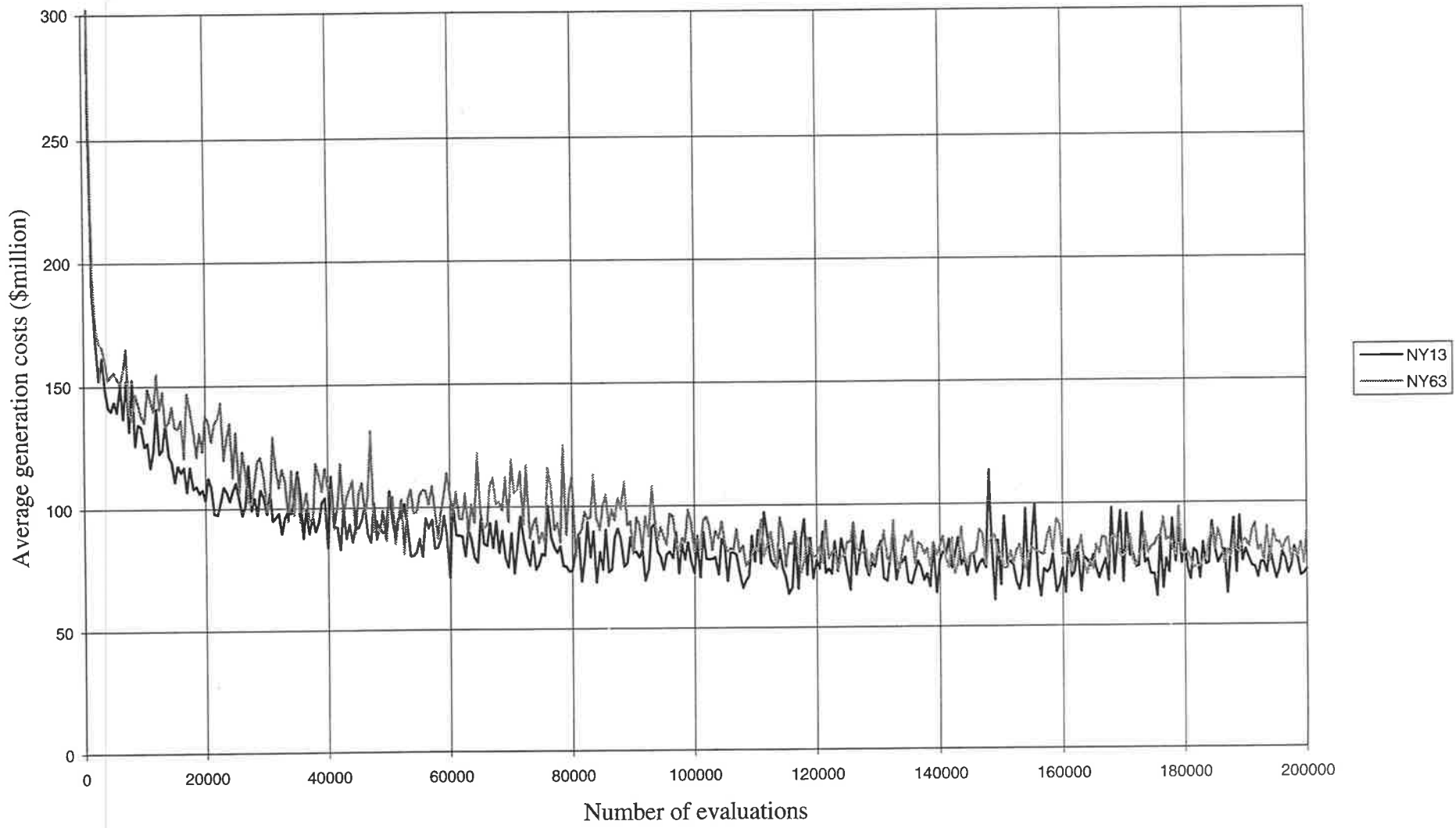


Figure 8.15 Average generation costs for GA runs NY13 (Improved GA) and NY63 (Traditional GA)

8.6 The GA Solutions to the New York Tunnels Problem

The five lowest cost feasible designs GA(1)-GA(5) and three low cost infeasible designs GA(6)-GA(8) identified by the GA model runs in this chapter are summarised in Table 8.13. The coded strings were shown in Figure 8.2. The designs GA(1) for \$38.796 million and GA(2) for \$39.062 million are competing local optima in the solution space. Although there is a small cost difference (0.69%), the designs are quite different. The coded strings representing GA(1) and GA(2) differ by 9 bits using Gray codes and 7 bits using binary codes (Figure 8.2). This was not a favourable situation for the Gray codes representation. Not one of the total 64 GA runs performed in this study succeeded in determining both solutions, although the GA runs usually determined one or the other. The design GA(2) was identified significantly more frequently than design GA(1). The reason for this is not apparent as the complete characteristics of the solution space in this region are not known.

Table 8.13 The five lowest cost feasible GA designs and three low cost infeasible GA designs

Tunnel	Diameters of duplicate tunnels (inches)							
	Feasible designs					Infeasible designs		
	GA(1)	GA(2)	GA(3)	GA(4)	GA(5)	GA(6)	GA(7)	GA(8)
[1]	0	0	0	0	0	0	0	0
[2]	0	0	0	0	0	0	0	0
[3]	0	0	0	0	0	0	0	0
[4]	0	0	0	0	0	0	0	0
[5]	0	0	0	0	0	0	0	0
[6]	0	0	0	0	0	0	0	0
[7]	0	144	156	0	0	0	84	0
[8]	0	0	0	0	0	0	0	0
[9]	0	0	0	0	0	0	0	0
[10]	0	0	0	0	0	0	0	0
[11]	0	0	0	0	0	0	0	0
[12]	0	0	0	0	0	0	0	0
[13]	0	0	0	0	0	0	0	0
[14]	0	0	0	0	0	0	0	0
[15]	120	0	0	120	108	96	0	0
[16]	84	96	96	84	96	96	96	96
[17]	96	108	96	108	96	96	96	96
[18]	84	72	84	72	84	84	84	84
[19]	72	72	72	72	72	72	72	72
[20]	0	0	0	0	0	0	0	0
[21]	72	72	72	72	72	72	72	72
Cost (\$mill.)	38.796	39.062	39.166	39.221	39.284	38.524 38.793 [†]	36.190 38.811 [†]	33.626 38.694 [†]

[†] including penalty costs based on k =\$5million/ft

The GA solutions in Table 8.13 would seem to belong to one of two families of designs. The designs GA(2) and GA(3) duplicate tunnel [7] in City Tunnel No.1. The feasible members of the competing family of designs, GA(1), GA(4) and GA(5), duplicate tunnel [15] at the upstream end of City Tunnel No.2. It would seem certain that tunnels [16], [17], [18], [19] and [21] require duplication. The feasible GA designs duplicate only 6 of the 21 existing tunnels. The coded strings in Figure 8.2 emphasise the coding similarities within the family groups and coding similarities relating members from opposing family groups. The infeasible GA(8) solution which duplicates only 5 tunnels is a coded string which is closely related to both local optima GA(1) and GA(2).

The hydraulic heads at the critical nodes for the GA designs are given in Table 8.14.

Table 8.14 Hydraulic heads for GA designs

Minimum allowable head, feet	Hydraulic heads at the three most critical nodes, feet							
	Feasible designs					Infeasible designs		
	GA(1)	GA(2)	GA(3)	GA(4)	GA(5)	GA(6)	GA(7)	GA(8)
Node 16 260.0 (Surplus)	260.52 +0.52	260.01 +0.01	260.08 +0.08	260.52 +0.52	260.23 +0.23	259.95 -0.05*	259.48 -0.52*	258.99 -1.01*
Node 17 272.8 (Surplus)	272.86 +0.06	272.82 +0.02	272.88 +0.08	272.86 +0.06	273.02 +0.22	272.75 -0.05*	272.28 -0.52*	271.79 -1.01*
Node 19 255.0 (Surplus)	255.71 +0.71	255.71 +0.71	255.04 +0.04	256.43 +1.43	255.39 +0.39	255.10 +0.10	254.50 -0.50*	254.07 -0.93*
Cost (\$million)	38.796	39.062	39.166	39.221	39.284	38.524 38.793†	36.190 38.811†	33.626 38.694†

* a negative value indicates the minimum HGL constraint is violated

† including penalty costs based on $k=\$5\text{million/ft}$

The infeasible GA solutions demonstrate significant cost savings for some small violations of the hydraulic head constraints. The infeasible solutions may be acceptable in some circumstances, particularly if a small hydraulic head deficiency is accompanied by large tunnel cost savings. The design GA(8) for \$33.626 million represents a cost saving of about \$5.17 million (13.3%) compared with the lowest cost feasible design for a hydraulic head deficiency of only about 1 foot at nodes 16, 17 and 19.

The total solution costs including penalty costs for the infeasible GA solutions GA(6)-GA(8) assuming a penalty multiplier $k=\$5\text{million/ft}$ are given in Table 8.13, based on the violations of the minimum allowable HGL in Table 8.14. The total solution costs of the infeasible designs are less than the tunnel construction cost of the best feasible design for \$38.796 million using $k=\$5\text{million/ft}$, but greater than \$38.796 million for $k=\$30\text{million/ft}$.

The equation used to relate head loss to flow in this study is the Hazen-Williams head loss formula which was given by Eqs. 2.3 and 2.4 in Chapter 2. Small variations of the constants in Eq. 2.4 (for US customary units) can be the difference between a solution being classified as feasible or infeasible for the New York tunnels problem. The balanced tunnel flows and nodal heads for the design GA(1) are given in Tables 8.15 and 8.16 respectively. A greater proportion of the flow (57.9%) supplying the city is conveyed by City Tunnel No.2. The design GA(2) which duplicates City Tunnel No.1 (tunnel [7]) conveys a flow of approximately 884 cfs by this tunnel compared to 848 cfs for GA(1). The very long existing tunnel [20] is almost redundant as a consequence of optimisation (flow in tunnel [20] is just 8 cfs). The nodes 16, 17 and 19 at the downstream extremities of the system are the critical nodes.

Table 8.15 Balanced tunnel flows for design GA(1)

Tunnel	Existing diameter (in)	Duplicate diameter (in)	Equivalent diameter (in)	Tunnel cost (\$million)	Head loss (ft)	Head loss/ 1000 ft	Tunnel flow (cfs)
[1]	180	0	180.0	0.0	5.380	0.464	848.285
[2]	180	0	180.0	0.0	7.417	0.375	755.885
[3]	180	0	180.0	0.0	2.148	0.294	663.485
[4]	180	0	180.0	0.0	1.875	0.226	575.285
[5]	180	0	180.0	0.0	1.428	0.166	487.085
[6]	180	0	180.0	0.0	2.190	0.115	398.885
[7]	132	0	132.0	0.0	3.138	0.327	310.685
[8]	132	0	132.0	0.0	2.202	0.176	222.485
[9]	180	0	180.0	0.0	0.031	0.003	58.500
[10]	204	0	204.0	0.0	0.141	0.013	167.974
[11]	204	0	204.0	0.0	1.456	0.100	516.015
[12]	204	0	204.0	0.0	3.204	0.263	867.315
[13]	204	0	204.0	0.0	8.003	0.332	984.415
[14]	204	0	204.0	0.0	8.274	0.392	1076.815
[15]	204	120	221.9	6.463	4.699	0.303	1169.215
[16]	72	84	102.0	7.049	1.332	0.050	57.500
[17]	72	96	111.1	9.859	13.978	0.448	234.200
[18]	60	84	95.8	6.408	6.137	0.256	117.100
[19]	60	72	86.5	3.182	13.168	0.914	178.040
[20]	60	0	60.0	0.0	0.672	0.017	8.040
[21]	72	72	93.7	5.834	13.699	0.519	161.960

Table 8.16 Balanced node hydraulic heads for design GA(1)

Node	Measured HGL (ft)	Minimum allowable HGL (ft)	Residual hydraulic head (ft)
1	300.0	Reservoir	-
2	294.62	255.0	+39.62
3	287.20	255.0	+32.20
4	285.06	255.0	+30.06
5	283.18	255.0	+28.18
6	281.75	255.0	+26.75
7	279.56	255.0	+24.56
8	276.43	255.0	+21.43
9	274.22	255.0	+19.22
10	274.19	255.0	+19.19
11	274.36	255.0	+19.36
12	275.82	255.0	+20.82
13	279.02	255.0	+24.02
14	287.03	255.0	+32.03
15	295.30	255.0	+40.30
16	260.52	260.0	+0.52
17	272.86	272.8	+0.06
18	261.84	255.0	+6.84
19	255.71	255.0	+0.71
20	261.20	255.0	+6.20

Results have been presented for the improved GA model applied to the New York tunnels problem. The lowest cost known feasible discrete tunnel solution has been found by the improved GA for \$38.796 million. In addition, the GA runs have generated a range of solutions which gives the designer a choice of potential designs.

8.7 Comparison of GA Results with Previous Studies

Since the original work by Schaake and Lai (1969), a number of researchers have considered the New York City water supply tunnels as a case study to demonstrate the effectiveness of their respective pipe network optimisation techniques. These previous studies are summarised in Table 8.17.

In the following comparison, a *continuous* diameter design is an optimised set of tunnel diameters that may take on any continuous real value. A *discrete* diameter design is a set of tunnel diameters that are selected from a specified set of available tunnel sizes. A *split pipe* design may be derived from a continuous diameter design by decomposing a length of continuous diameter into partial lengths of the two adjacent discrete diameters (one smaller and one larger) to create a pipe with equivalent hydraulic properties. The designs for the New York water supply tunnels network expansions achieved in previous studies are given in Table 8.18.

Table 8.17 Previous studies of the New York City tunnels problem

Author (year)	Optimisation technique	Diameter type	Lowest cost design (\$million)	Feasible or Infeasible*
Schaake and Lai (1969)	Single step linear programming	Continuous	78.09	Feasible
Quindry et al. (1981)	Linear programming gradient search	Continuous	63.58	Feasible
Gessler (1982)	Partial enumeration (City Tunnel No.1 reinforcement)	Continuous	41.2	Feasible
		Discrete	41.8	Feasible
	Partial enumeration (City Tunnel No.2 reinforcement)	Continuous	46.9	Feasible
		Discrete	49.6	Feasible
Bhave (1985)	Linear programming with heuristics	Continuous	40.18	Feasible
Morgan and Goulter (1985)	Linear programming with heuristics	Split Pipe	38.9	Infeasible
		Discrete	39.20	Infeasible
Kessler (1988)	Decomposition method of two submodels	Split pipe	39.0	Infeasible
Fujiwara and Khang (1990)	Modified nonlinear programming / gradient search	Continuous	36.1	Infeasible
		Split Pipe	36.6	Infeasible
Loganathan et al. (1995)	Outer global search - inner LP optimisation	Split Pipe	38.04	Infeasible

* infeasible by KYPIPE analysis (see Tables 8.19 and 8.20)

In the original work on the problem, Schaake and Lai (1969) used a linear programming approach to find the optimum tunnel diameters for assumed values of the total head at each node. The decision variable for each tunnel was its diameter raised to the power 2.63, thus leading to a set of linear constraints. The nonlinear terms in the objective function were approximated using piece-wise linearisation. No check was made to determine whether the assumed nodal heads led to an optimum solution overall. As shown in Table 8.18, the final solution obtained involves duplicating almost all tunnels in the system at a cost of \$78.09 million (all costs in this chapter are given in 1969 US dollars). As the minimum cost solution to a pipe network problem tends towards a branched system, it is expected that better solutions to the problem can be obtained by duplicating fewer tunnels.

The model of Quindry et al. (1981) is an extension of the linear programming approach used by Schaake and Lai. First, an optimal solution for an assumed set of nodal heads was obtained. The dual variables were then used to identify the relative changes required in the nodal heads so as to get the maximum rate of improvement in the objective function. The heads were adjusted and the linear program was rerun. This procedure was repeated until no further improvement

was obtained. As shown in Table 8.18, the solution obtained involves no duplication of City Tunnel No.1. The total cost of the continuous diameter design was \$63.58 million.

Table 8.18 Designs achieved by previous studies

Tunnel	Diameters of duplicate tunnels, inches							
	Feasible designs				Infeasible designs*			
	Schaake and Lai (1969)	Quindry et al. (1981)	Gessler (1982)	Bhave (1985)	Morgan and Goulter** (1985)	Kessler (1988)	Fujiwara and Khang (1990)	Loganathan et al. (1995)
[1]	52.02	0.0	0	0.0	0	0.0	0.0	0.0
[2]	49.90	0.0	0	0.0	0	0.0	0.0	0.0
[3]	63.41	0.0	0	0.0	0	0.0	0.0	0.0
[4]	55.59	0.0	0	0.0	0	0.0	0.0	0.0
[5]	57.25	0.0	0	0.0	0	0.0	0.0	0.0
[6]	59.19	0.0	0	0.0	0	0.0	0.0	0.0
[7]	59.06	0.0	100	0.0	144	0.0	73.62	120.68
[8]	54.95	0.0	100	0.0	0	0.0	0.0	0.0
[9]	0.0	0.0	0	0.0	0	0.0	0.0	0.0
[10]	0.0	0.0	0	0.0	0	0.0	0.0	0.0
[11]	116.21	119.02	0	0.0	0	0.0	0.0	0.0
[12]	125.25	134.39	0	0.0	0	0.0	0.0	0.0
[13]	126.87	132.49	0	0.0	0	0.0	0.0	0.0
[14]	133.07	132.87	0	0.0	0	0.0	0.0	0.0
[15]	126.52	131.37	0	136.43	0	156.11	0.0	0.0
[16]	19.52	19.26	100	87.37	96	72.0	99.01	98.05
[17]	91.83	91.71	100	99.23	96	96.6	98.75	96.06
[18]	72.76	72.76	80	78.17	84	78.0	78.97	84.26
[19]	72.61	72.64	60	54.40	60	59.78	83.82	72.39
[20]	0.0	0.0	0	0.0	0	0.0	0.0	0.0
[21]	54.82	54.97	80	81.50	84	72.27	66.59	72.03
Cost (\$mill.)	78.09	63.58	41.8	40.18	39.20	39.0	36.1	38.04
	Continuous	Continuous	Discrete	Continuous	Discrete	Split pipe	Continuous	Split pipe

* infeasible by KYPIPE analysis (see Table 8.19)

** only very slightly infeasible

Gessler (1982) used a partial enumeration technique and discrete tunnel sizes to search a subset of the total solution space. He searched two separate regions of the solution space with consideration of the reinforcement of either City Tunnel No.1 or City Tunnel No.2. The lowest cost discrete diameter solution obtained in each case was used as a starting solution for a gradient search technique that used continuous tunnel sizes. The lowest cost discrete diameter design for the reinforcement of City Tunnel No.1 involved the duplication of only seven tunnels (Table 8.18) at a cost of \$41.8 million.

Bhave (1985) used a heuristic procedure based on the identification of an efficient branched configuration. In the method, nodal heads for the branched configuration were progressively adjusted so as to give the maximum reduction in system cost. The method identified City

Tunnel No.2 (without tunnel [20]) as the branched configuration to be optimised. The optimal configuration (Table 8.18) involved the duplication of only six tunnels at a total cost of \$40.18 million.

Morgan and Goulter (1985) applied a linear programming approach coupled with a hydraulic network solver to the New York water supply tunnels problem. They used a split pipe approach in which the decision variables were the lengths of tunnel of a specified diameter that replace the current size. Tunnels may be increased or reduced in size, or eliminated entirely (the last two alternatives do not apply to the New York problem.) After each iteration, hydraulic consistency was checked using the hydraulic network solver. The discrete pipe solution obtained by Morgan and Goulter is given in Table 8.18 and involves duplicating six tunnels at a cost of \$39.20 million. The discrete pipe solution was found to be slightly infeasible but acceptable in terms of normal expected accuracies of simulation modelling. A split pipe solution with a cost of \$38.9 million was also obtained.

Kessler (1988) applied a decomposition technique consisting of two submodels (Kessler and Shamir, 1991) to the New York tunnels problem. In the first submodel the heads at the nodes are fixed, and a minimum concave cost of flow algorithm is used to find the pipe flows. These are then fixed and the head variables are found in the second submodel using linear programming. The two submodels are solved interactively until convergence is achieved (which usually occurs after two iterations). It can be shown that a local optimum is obtained. A split pipe solution with a cost of \$39.0 million was obtained. This solution is shown to be infeasible in Table 8.19.

Fujiwara and Khang (1990) used a two-phase decomposition method that combined the methods of Alperovits and Shamir (1977), Quindry et al. (1981), and Mahjoub (1983) similar to the model described by Kessler and Shamir (1991). In the first phase, a nonlinear programming gradient method was used to find the optimum head loss in each tunnel (and hence the tunnel diameters) for an assumed set of flows. A correction was then applied to the assumed flow in each loop using the Lagrange multipliers associated with the previous solution. This process was continued until it converged on a local optimum. In the second phase, the nodal heads obtained at the end of the first phase were fixed. A nonlinear optimisation model was run that found the optimum flow in each pipe for these nodal heads. This gave a new local optimum that could be used to restart the first phase. Iterations occurred between the two phases in such a way as to obtain a better local optimum solution. Fujiwara and Khang (1990) proposed a continuous diameter pipe solution with a cost of \$36.1 million, but this solution is shown to be clearly infeasible in Table 8.19.

Loganathan et al. (1995) presented a pipe network optimisation procedure with an outer search scheme to choose alternative, feasible flow distributions to initialise an inner linear programming formulation which determines pipe diameters (and nodal heads) of a local optimal solution. The outer global search strategies of multi-start local search and simulated annealing were adopted to move between local optimal solutions and help locate isolated areas of the feasible region. Loganathan et al. (1995) applied their outer search - inner optimisation method to the optimisation of the New York tunnels problem. They identified a split pipe diameter design for \$38.04 million, which was found to be just infeasible in Table 8.20 by the hydraulic analysis model developed in this research and the form of the Hazen-Williams used.

The hydraulic heads at the critical nodes for the previous optimised New York tunnel network expansion designs are presented in Tables 8.19 and 8.20. Table 8.19 gives the hydraulic grade line profile (hydraulic heads) calculated using the conventional pipe network simulation model KYPIPE (Wood, 1974), while Table 8.20 shows the hydraulic heads calculated using the hydraulic simulation model developed in Chapter 2 of this thesis. The HGL values correspond closely for both models. The simulation model developed in this research consistently produced hydraulic heads that were higher by about 0.05 ft (a difference of 0.02%).

Table 8.19 Hydraulic heads for previous designs using KYPIPE

Minimum allowable head, feet	Hydraulic heads at three most critical nodes, feet							
	Feasible designs				Infeasible designs			
	Schaake and Lai (1969)	Quindry et al. (1981)	Gessler (1982)	Bhave (1985)	Morgan and Goulter (1985)	Kessler (1988)	Fujiwara and Khang (1990)	Loganathan et al. (1995)
Node 16 260.0 (Surplus)	260.97 +0.97	260.92 +0.92	260.27 +0.27	260.80 +0.80	261.50 +1.50	258.46 -1.54*	259.25 -0.75*	259.87 -0.13*
Node 17 272.8 (Surplus)	273.77 +0.97	273.62 +0.82	273.06 +0.26	273.34 +0.54	272.75 -0.05*	273.00 +0.20	272.22 -0.58*	272.67 -0.13*
Node 19 255.0 (Surplus)	256.08 +1.08	255.98 +0.98	255.80 +0.80	255.90 +0.90	254.93 -0.07*	255.16 +0.16	254.18 -0.82*	254.86 -0.14*
Cost (\$m)	78.09	63.58	41.8	40.18	39.20	39.0	36.1	38.04

* a negative value indicates the minimum HGL constraint is violated

Fujiwara and Khang (1990) claimed that their design was the lowest cost published design solution to the New York City tunnels problem but it is actually infeasible with the heads at nodes 16, 17 and 19 falling below the minimum allowable values. GA design (1) with a cost of \$38.796 million is the lowest cost feasible design for the New York City water tunnels problem to date. The GA design (1) is a discrete diameter design and the cost is less than continuous diameter designs achieved by other optimisation techniques. A gradient search

could use the GA designs as a starting points to determine cost-effective continuous diameter designs if desired.

Table 8.20 Hydraulic heads for previous designs using the simulation model developed in this research

Minimum allowable head, feet	Hydraulic heads at three most critical nodes, feet							
	Feasible Designs				Infeasible Designs			
	Schaake and Lai (1969)	Quindry et al. (1981)	Gessler (1982)	Bhave (1985)	Morgan and Goulter (1985)	Kessler (1988)	Fujiwara and Khang (1990)	Loganathan et al. (1995)
Node 16 260.0 (Surplus)	261.02 +1.02	260.97 +0.97	260.32 +0.32	260.84 +0.84	261.56 +1.56	258.51 -1.49*	259.30 -0.70*	259.93 -0.07*
Node 17 272.8 (Surplus)	273.81 +1.01	273.66 +0.86	273.10 +0.30	273.38 +0.58	272.79 -0.01*	273.04 +0.24	272.26 -0.54*	272.71 -0.09*
Node 19 255.0 (Surplus)	256.14 +1.14	256.04 +1.04	255.86 +0.86	255.96 +0.96	254.99 -0.01*	255.22 +0.22	254.24 -0.76*	254.92 -0.08*
Cost (\$m)	78.09	63.58	41.8	40.18	39.20	39.0	36.1	38.04

* a negative value indicates the minimum HGL constraint is violated

Three feasible solutions have been generated by the improved GA that are lower in cost (\$38.796m, \$39.062m and \$39.166m) than the Morgan and Goulter (1985) discrete pipe solution in Table 8.18 for \$39.20m. The Morgan and Goulter solution has been found to be only slightly infeasible. As shown in Tables 8.19 and 8.20, the hydraulic head violations are very small and the Morgan and Goulter solution could be considered to be a valid feasible solution to the problem (allowing for small differences in the actual values of the coefficient and exponent used in the Hazen-Williams head loss equation. The Morgan and Goulter solution with a total cost of \$39.44 million (for a tunnel construction cost of \$39.20 million and a penalty cost of only \$0.24 million based on $k=\$30\text{million/ft}$) was frequently generated by the GA model and was regularly a member of the elite population at the termination of the improved GA runs.

8.8 Summary and Conclusions

This chapter has presented the results of the application of the improved genetic algorithm formulation (developed in Chapters 6 and 7) to the classic New York City water supply tunnels (Schaake and Lai, 1969) optimisation problem.

The features of the improved GA include:

- variable power law form of fitness scaling
- creeping mutation
- substrings represented by Gray codes
- use of an elitist strategy

The results for a series of GA runs have been presented for the improved GA formulation, the traditional GA formulation (without the listed features), and intermediate GA formulations which attempt to measure (to some degree) the significance of each feature. The combination of the features in the improved GA (GA runs NY11-NY19) are effective and the improved GA performs significantly better than the traditional three-operator GA (GA runs NY61-NY69). The fitness scaling is the most effective feature of the improved GA. Elitism provides a small improvement. Further experimentation could be required to establish the value of using substrings of Gray codes and the creeping mutation for the New York tunnels problem.

The GA results have been compared to results previously reported in the literature using other more traditional optimisation techniques. The design by Fujiwara and Khang (1990) for \$36.1 million was considered to be the lowest cost known feasible solution, however the solution is shown to be infeasible.

The improved GA generated three tunnel networks with a lower cost than the discrete pipe solution of Morgan and Goulter (1985) for \$39.20 million. The hydraulic simulation model developed in Chapter 2 (adopting the Hazen-Williams formula in Eqs. 2.3 and 2.4) and the conventional KYPIPE hydraulic analysis package both indicate that the Morgan and Goulter solution does not strictly satisfy the minimum HGL profile constraints. The lowest cost feasible discrete pipe solution generated by the GA has a tunnel construction cost of \$38.796 million. This solution is 1% cheaper than Morgan and Goulter's solution and it may be that this solution is the global optimum discrete solution or very close to it.

It is recommended that for any new system, a number of GA runs are performed to experiment with the penalty multiplier, the GA formulations and other GA parameters. Several alternative solutions (feasible and infeasible) are produced by a series of GA runs. The decision maker can choose between similarly priced designs and consider beneficial features of other (perhaps infeasible) designs. Some non-quantifiable criteria (such as anticipated future developments) may be used to decide between alternative designs.

9 GA Optimisation of the Water System Expansion Plan for the Fort Collins - Loveland Water District

9.1 Introduction

This chapter describes the application of the genetic algorithm (GA) search to optimise the design of future expansion plans to the Fort Collins - Loveland Water District's water distribution system. A design to meet the projected year 2015 peak hour demands was prepared using the GA pipe network optimisation model. The GA-optimised design is compared to an expansion plan prepared by a consultant using a hydraulic simulation tool and standard design rules.

The Fort Collins - Loveland Water District (FCLWD) supplies water for agricultural and municipal use to an area of about 60 square miles between Fort Collins and Loveland in Colorado, USA. The population is increasing and the water supply system will require expansion to meet the increasing water demands.

9.2 The 1993 Master Plan

The FCLWD has periodically updated its water system expansion plans in the form of a Master Plan document for many years. The latest re-assessment of the District's Master Plan was carried out in 1993 by a Fort Collins engineering consultant. The Master Plan document (*Fort Collins engineering consultant, 1993*) proposed system expansions in three construction phases to supply water to the District through the year 2015.

The 1993 Master Plan made recommendations for: (1) future sources of supply, (2) new network storages, (3) existing pump station upgrades and new booster pump stations, (4) the locations and settings for pressure reducing valves, and (5) improvements to the pipe network including new pipes and the duplication of existing pipes. The future sources and expansions to the transmission system are sized to supply the projected 2015 maximum day demand. The proposed expansions to the distribution system are sized to supply the 2015 peak hour demands.

The Master Plan study used a hydraulic simulation computer model developed by the U.S.A. Environmental Protection Agency (EPA) called EPANET (Rossman, 1994). The input data and output results of simulating the Master Plan design subject to the 2015 peak hour demands using the EPANET model are provided in Appendix A. The configuration of the Fort Collins -

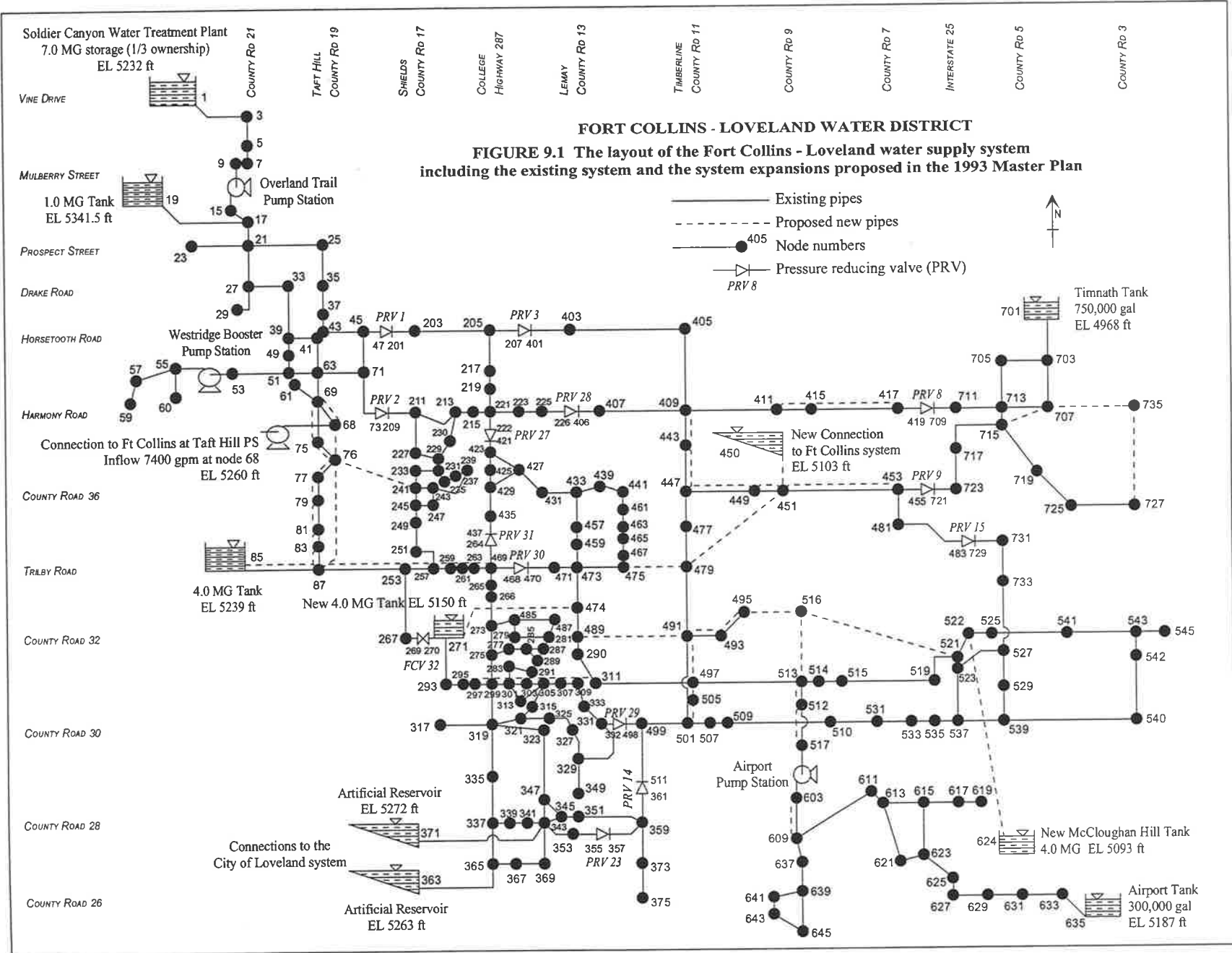
Loveland water supply system (including the existing system and the system expansions proposed in the Master Plan) modelled using EPANET is shown in Figure 9.1.

9.3 The Genetic Algorithm Approach to the FCLWD System

The GA optimisation study is applied to identify a low cost expansion plan for the FCLWD water distribution pipe network for 2015. The design recommended in the 1993 Master Plan and set out in the EPANET hydraulic simulation data in Appendix A is an ideal starting point for the GA design of this large, complex water transfer and distribution system. The 2015 Master Plan system is composed of 323 pipes (of which 277 are existing pipes and 46 are proposed new and duplicate pipes) and 253 junction nodes. The water is supplied from 5 alternative sources of supply including 4 connections to adjacent water systems. There are 2 source pump stations and 5 booster pump stations, although only 2 booster pump stations are modelled. The water is stored to balance peak demand periods in 7 storage tanks distributed throughout the system. There is a flow control valve regulating flow to a storage tank and 13 pressure reducing valves which isolate the system into 5 pressure zones. Some conclusions of the Master Plan with respect to the configuration of system components such as future sources of supply, storage tanks and pump stations are adopted in the GA design strategy.

The GA optimises aspects of the design of the water distribution pipe network including (1) the diameters of the new pipes, (2) the diameters of duplicate pipes (pipes placed parallel to existing pipes) and (3) the pressure settings for pressure reducing valves (PRVs) based on an assumed pipe network configuration for the year 2015 design. The GA is used to select sizes of new pipes and to select sizes of (or omit) duplicate pipes. The GA considers duplicate pipes in the locations recommended by the Master Plan and in other locations where existing pipes are operating with high velocities and/or high head losses. The GA selects appropriate pressure settings for PRVs in the locations recommended by the Master Plan.

The Master Plan predicted the year 2015 peak hour demand pattern (provided in the EPANET hydraulic simulation data in Appendix A) and specified the District's system performance requirements such as minimum and maximum nodal pressures and maximum pipe velocities. The GA design is subject to the same instantaneous 2015 peak hour demand pattern and is expected to satisfy the same system performance constraints to enable a fair comparison between the original Master Plan design and the GA optimised design.



FORT COLLINS - LOVELAND WATER DISTRICT
FIGURE 9.1 The layout of the Fort Collins - Loveland water supply system including the existing system and the system expansions proposed in the 1993 Master Plan

The GA optimisation assumes a baseline system configuration which consists of the existing 1993 system and the following aspects of the 2015 Master Plan design:

- the pipe network layout and pipe sizes (as in 1993)
- the Overland Trail source pump station
- the upgrade of the Taft Hill source pump station proposed by the Master Plan
- the Airport and Westridge booster pump stations
- the connections to the City of Loveland water system
- the proposed new connection to the City of Fort Collins water system
- the storage tanks and their assumed tank water levels
- the new storage tank locations proposed by the Master Plan
- the sites of the PRVs and relocated PRVs proposed by the Master Plan

The GA designs represent a set of improvements to this baseline system configuration. The evaluation of a trial GA design requires the calculation of the pipe costs and a check of the hydraulic feasibility of the proposed design.

The KYPIPE hydraulic simulation model (Wood, 1974) is integrated with the GA optimisation model routines for this study. The KYPIPE simulation model is used to perform accurate hydraulic analyses of the complex Fort Collins - Loveland system designs. The KYPIPE hydraulic simulation model determines the balanced pipe flows and node pressures of the trial GA designs subject to the 2015 peak hour demand pattern. The EPANET (Rossman, 1994) hydraulic simulation model was used to verify the final GA design.

9.4 Sources of Supply in 2015

The Master Plan anticipated the sources of supply to the Fort Collins - Loveland Water District in the year 2015 will include:

- Soldier Canyon (node 1, Figure 9.1) and Overland Trail pump station (nodes 9-15)
- connection to the Fort Collins system at Taft Hill pump station (node 68)
- connections to the Loveland system (nodes 363 and 371)
- a proposed new connection to the Fort Collins system at County Road 9 (node 450)

The District will need additional water supply sources beyond 2008. The Master Plan outlined some of the alternatives such as purchasing additional water from the adjacent systems of Loveland and from Fort Collins at the new County Road 9 connection, or exchanging water with Fort Collins. Another alternative is to increase the supply to the system from Soldier Canyon. For this case, it would be necessary to upgrade the Overland Trail pump station and

provide additional transmission lines. A preliminary investigation was carried out in the development of the Master Plan of the trade-off between purchasing additional water from Fort Collins and Loveland, and producing water at Soldier Canyon. The conclusions favoured making the system improvements and using water from Soldier Canyon rather than purchasing additional water, however, the Soldier Canyon alternative was not implemented in the report. The Master Plan instead chose to expand the Taft Hill pump station and construct a parallel transmission pipeline from the pump station to the existing 4.0 MG Tank (at Trilby Road) to meet the 2015 demands.

The *Taft Hill source pump station* (adjacent to node 68) supplies the District's system from a connection to the City of Fort Collins system. The pump station consists of 3 identical 75 horsepower pumps, each pump with rated discharge, $Q_R=1,750$ gpm, rated head, $H_R=135$ ft and rated speed, $N_R=1,750$ rpm. The Master Plan estimated the capacity of the Taft Hill source pump station by the intersection of the pump curve and the system curve. The maximum capacity of the pump station is about 5,200 gpm, but the reliable capacity (with one standby pump unit) is about 4,200 gpm. The Master Plan found the transmission line between Taft Hill pump station and the 4.0 MG Tank must be duplicated regardless of whether increased supply is obtained from Soldier Canyon or the City system via Taft Hill pump station. The effect on the system curve of the duplication of the pipe was expected to increase the maximum capacity of Taft Hill pump station to about 7,400 gpm with all three existing pumps in operation. The Master Plan found that if a spare pump is purchased by the District as a standby unit, a reliable capacity of 7,400 gpm may be assumed. The cost of system improvements are less for the Taft Hill alternative than for the Soldier Canyon alternative, although more water must be purchased from Fort Collins.

The Master Plan concluded that the increased supply was best achieved by the Taft Hill pump station improvements. The connection to the City of Fort Collins system at Taft Hill pump station is modelled as a demand node (node 68) with a negative demand (an inflow) and known hydraulic grade line for the proposed 2015 design. The Taft Hill pump station operation itself is not modelled. The inflow at the connection of 7,400 gpm for the 2015 design is the expected increased pump station capacity.

The *Overland Trail source pump station* (nodes 9-15) boosts flow from the Soldier Canyon Tanks. The pump station consists of four identical 75 horsepower pumps, each with rated discharge, $Q_R=1,200$ gpm, rated head, $H_R=180$ ft and rated speed, $N_R=1,750$ rpm. The Master Plan estimated the maximum capacity of the pump station to be about 4,900 gpm, however, the reliable capacity (with one standby pump) is about 4,200 gpm. The operation of the Overland Trail pump station was modelled in the Master Plan design for the 2015 peak hour demands with a design head of 165.8 ft and design flow of 4,252 gpm which corresponds to

three parallel pumps in operation and one standby pump as shown in Table 9.1. The pump curve for the Overland Trail source pump station with three pumps on from the Master Plan is shown in Figure 9.2.

Table 9.1 Pump station operation for the 2015 peak hour demands*

Pump Station	Nodes	Operation	Design Head (ft)	Total Design Flow (gpm)
Overland Trail source PS	9-15	3 pumps	165.8	4,252
Airport booster PS	517-603	1 pump	125	620
Westridge booster PS	53-55	1 pump	79	191

* Taft Hill source PS operation is not modelled, however, the pumping station is assumed to be operating 3 pumps with a combined design flow of 7,400 gpm

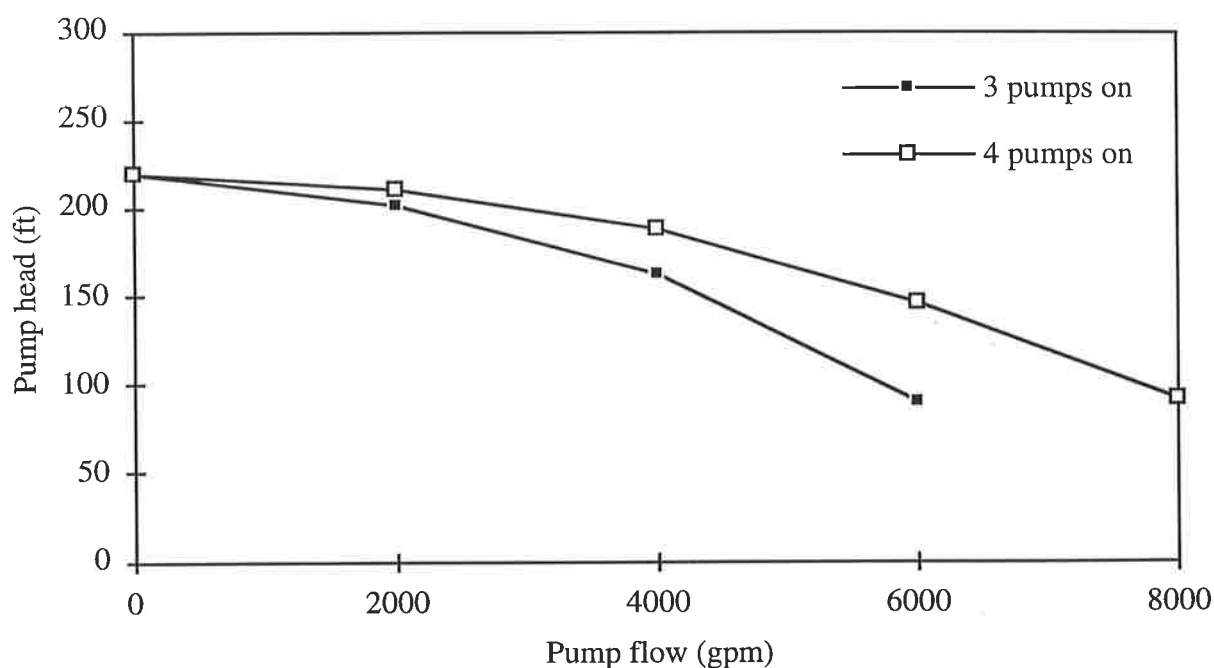


Figure 9.2 Pump curves for the Overland Trail source pump station

The existing connections to the City of Loveland system (nodes 363 and 371) and the proposed new connection to the City of Fort Collins system (node 450) were modelled as artificial reservoirs of known elevations shown in Figure 9.1.

9.5 Booster Pump Stations

The GA models two existing booster pump stations: the Airport booster pump station which pumps to the airport area and supplies the elevated Airport storage tank, and the Westridge booster pump station which supplies the Westridge area (south of Horsetooth along a ridge next

to the foothills) with adequate pressures and fire flows. The booster pump stations are modelled according to Table 9.1 for the 2015 peak hour demands.

The *Airport booster pump station* (nodes 517-603) is equipped with several different sized pumps. The Master Plan concluded the pumping capacity would need to be increased by about the year 2007. The pump station was modelled in the 2015 design for peak hour demands with a design head of 125 ft and design discharge of 620 gpm which corresponds to the ESP-1 operating curve in the Master Plan. The pump curve for the Airport booster pump station with one or three pumps on is shown in Figure 9.3. The improvements to the pump station are not modelled. The hydraulic simulation of the Master Plan design (Appendix A) indicates that there are no pressure problems in the airport area subject to the 2015 peak hour demands.

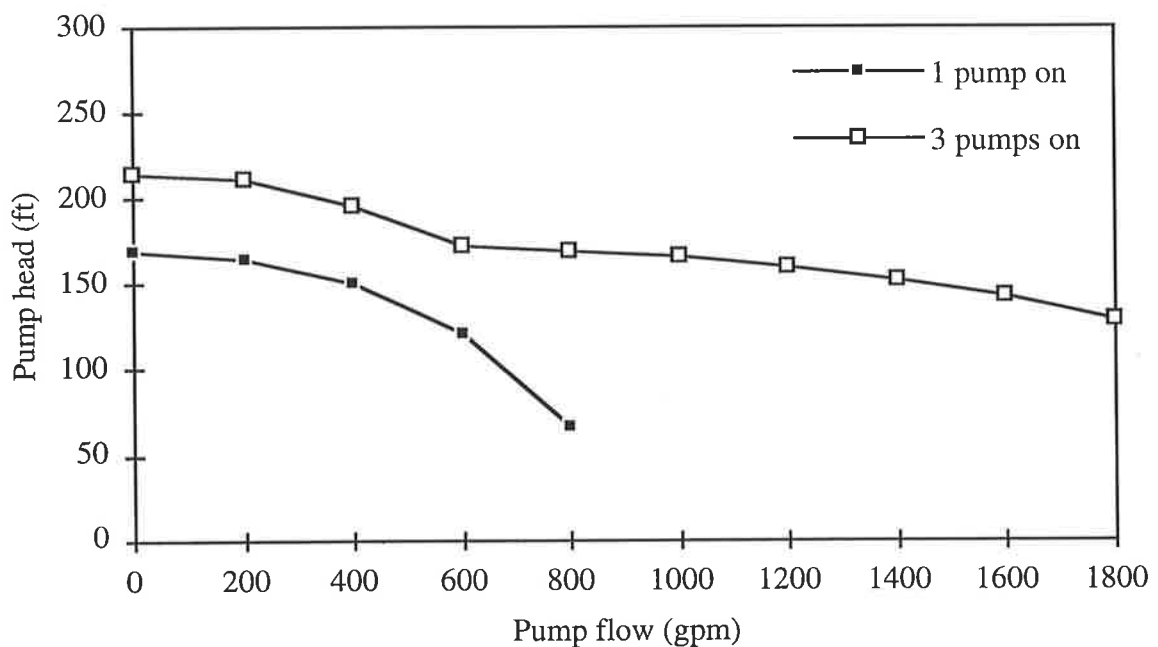


Figure 9.3 Pump curves for the Airport booster pump station

The Master Plan found the capacity of the pumps of the *Westridge booster pump station* (nodes 53-55) will be sufficient beyond 2015, however, a standby pump may have to be purchased. The Westridge booster pump station was modelled in the 2015 design with a design head of 79 ft and design flow of 191 gpm (Table 9.1). This corresponds to only one pump operating which results in low pressures being experienced in the Westridge area (node 60) for the 2015 peak hour demands. The pressures will be improved by operating the two existing pumps.

It was not necessary to model the operation of the other proposed booster pump stations in the water distribution system for the 2015 peak hour demands. The *Burns Ranch booster pump station* supplies a small area west of Overland Trail and south of Drake Road. The capacity of the pump station should be sufficient beyond 2015. The new *County Road 32 booster pump*

station is proposed to serve the ridge west of College Avenue between County Road 28 and County Road 30. The low pressures experienced in the results of the hydraulic simulation for this area (nodes 317 and 319) would be corrected when the County Road 32 booster pump station is modelled. The new *County Road 36 booster pump station* is proposed to serve the area east of Taft Hill Road immediately south of County Road 36. An alternative to serve this area would be a separate pipe from Harmony Road. The effectiveness of this alternative could be evaluated in a future GA optimisation.

9.6 Storage Tanks

The storages (as shown in Figure 9.1) in the Fort Collins - Loveland Water District system as of 1993 consisted of:

- 1/3rd ownership in 7.0 MG storage at Soldier Canyon Filtration Plant (node 1)
- ground level steel 1.0 MG Tank (node 19)
- ground level steel 4.0 MG Tank (node 85)
- elevated 300,000 gallon Airport Tank (node 635)
- ground level steel 750,000 gallon Timnath Tank (node 701)

The total storage capacity is about 8.38 MG. The projected storage requirements of an additional 9.6 MG by 2015 were estimated in the Master Plan based on an equalisation storage equal to 25% of the maximum day demand, a fire fighting storage capable of delivering 2,500 gpm for a fire of duration 3 hours, and an emergency storage equal to one day of average demand.

The Master Plan considered at least three possible sites for the future storage including: the ridge west of College Avenue between County Road 30 and County Road 32, McCloughan Hill, and adjacent to the existing elevated Airport Tank. The Master Plan proposed a new 4.0 MG tank at the site west of College Ave (node 271) and a new 4.0 MG tank on McCloughan Hill (node 624) for the 2015 design.

The existing tanks and the new tanks are modelled in the 2015 design with the water levels as shown in Table 9.2 for the hydraulic analysis of the peak hour demands.

Table 9.2 Storage tank water levels for the 2015 peak hour

Tank	Node	Water Level (ft)
Soldier Canyon Tanks	1	5232
1.0 MG Tank	19	5341.5
4.0 MG Tank	85	5239
New Zone 3 Tank*	271	5150
Airport Tank	635	5187
McCloughan Hill Tank*	624	5093
Timnath Tank	701	4968

* New tanks proposed by the Master Plan

9.7 Existing Pipelines

The 1993 pipe network layout is shown in Figure 9.1. The pipeline data including diameters, lengths and Hazen-Williams roughness values are given in the EPANET hydraulic simulation data in Appendix A. The District tested the Hazen-Williams roughness values of the pipes and a roughness of $C_f=140$ is taken for all new and existing pipes i (with the exception $C_f=110$).

9.8 New Pipes and Duplicate Pipes

The GA optimisation model is used to find the lowest cost combination of pipe sizes for new and duplicate pipes such that the desired system performance requirements are satisfied. The District's system expansions are designed to supply the 2015 peak hour demands and maintain pressures at demand nodes between a minimum of 40 psi and a maximum of 100 psi.

The new pipes were required to have at least a minimum diameter of six inches for consideration by the GA, as the layout of proposed new pipes provides supply to new areas. The new pipes could not be assigned a zero diameter and deleted from the design. The available pipe sizes for new pipes are given in Table 9.3. The required locations of new pipes were determined by the Master Plan study. There were no alternative routes for new pipes considered in this study. The routes of proposed new and duplicate pipes recommended by the Master Plan for 2015 are shown in Figure 9.1.

As part of the Master Plan, EPANET hydraulic analyses were performed of the 1993 FCLWD system subject to the 1993 peak hour demands and the projected 2000 and 2015 peak hour demands. The hydraulic analyses indicated some areas of low pressures and some pipes operating at high velocities.

The Master Plan recommended duplicate pipes to improve the hydraulic performance of the system. The duplicate pipes are new pipes placed parallel to existing pipes (but not necessarily of the same diameter) to improve the system for transmission and distribution. The GA considered the locations of duplicate pipes considered in the Master Plan, as well as some alternative locations. The duplicate pipes are allowed a minimum zero diameter. In this way, the GA helps find the best layout of duplicate pipes by eliminating unnecessary duplications. The GA identifies the parts of the distribution system which require additional capacity. The allowable diameters for duplicate pipes are given in Table 9.3.

The possible new and duplicate pipe sizes include pipes with diameters up to 30 inches. The linear relationship between pipe diameter and cost per unit length of installed pipe in Table 9.3 is based on the Master Plan assumption. The pipe material and construction costs are considered by the GA optimisation. The cost of minor items such as valves, connections and ditch crossings are neglected for the purpose of this study.

9.8.1 The Master Plan pipe network design

The Master Plan observed certain guidelines in determining the layout of water distribution lines: for urban areas, a 12 inch diameter pipe grid system every half mile with alternating 8 inch and 6 inch pipes within quarter sections; for rural areas, a 12 inch diameter pipe grid system every mile with 6 inch interior pipes. The recommended pipe diameters for the Master Plan network expansions are shown in Figure 9.4 and the associated installed pipe costs for the Master Plan design are summarised in Table 9.4. The estimated total pipe cost of the expanded pipe network for the Master Plan design is \$ 5,851,000.

Table 9.3 Mapping of values of decision variables to corresponding integer code

Possible pipe sizes for new pipes		
Integer code	Diameter (inches)	Installed cost (\$/ft)
1	6	15
2	8	20
3	10	25
4	12	30
5	14	35
6	16	40
7	18	45
8	20	50
9	24	60
10	30	75

Possible pipe sizes for duplicate pipes		
Integer code	Diameter (inches)	Installed cost (\$/ft)
1	Do nothing	0
2	3	7.5
3	4	10
4	6	15
5	8	20
6	10	25
7	12	30
8	14	35
9	16	40
10	18	45
11	20	50
12	24	60
13	30	75

Possible settings for PRVs 1 and 3	
Integer code	Pressure setting (psi)
1	40
2	41
3	42
4	43
5	44
6	45
7	46
8	47
9	48
10	49
11	50
12	51
13	52
14	53
15	54
16	55

Possible settings for PRV 2	
Integer code	Pressure setting (psi)
1	Remove
2	45
3	46
4	47
5	48
6	49
7	50
8	51
9	52
10	53
11	54
12	55
13	56
14	57
15	58
16	59
17	60

Possible settings for PRV 8	
Integer code	Pressure setting (psi)
1	Remove
2	55
3	56
4	57
5	58
6	59
7	60
8	61
9	62
10	63
11	64
12	65
13	66
14	67
15	68
16	69
17	70
18	71
19	72
20	73
21	74
22	75

Possible settings for PRV 9	
Integer code	Pressure setting (psi)
1	50
2	51
3	52
4	53
5	54
6	55
7	56
8	57
9	58
10	59
11	60
12	61
13	62
14	63
15	64
16	65
17	66
18	67
19	68
20	69
21	70

Possible settings PRVs 29 and 23	
Integer code	Pressure setting (psi)
1	40
2	41
3	42
4	43
5	44
6	45

Possible settings for PRVs 28, 27, 31 and 30	
Integer code	Pressure setting (psi)
1	40
2	41
3	42
4	43
5	44
6	45
7	46
8	47
9	48
10	49
11	50

Possible settings for PRV 15	
Integer code	Pressure setting (psi)
1	Remove
2	90
3	91
4	92
5	93
6	94
7	95
8	96
9	97
10	98
11	99
12	100

Possible settings for PRV 14	
Integer code	Pressure setting (psi)
1	45
2	46
3	47
4	48
5	49
6	50
7	51
8	52
9	53
10	54
11	55

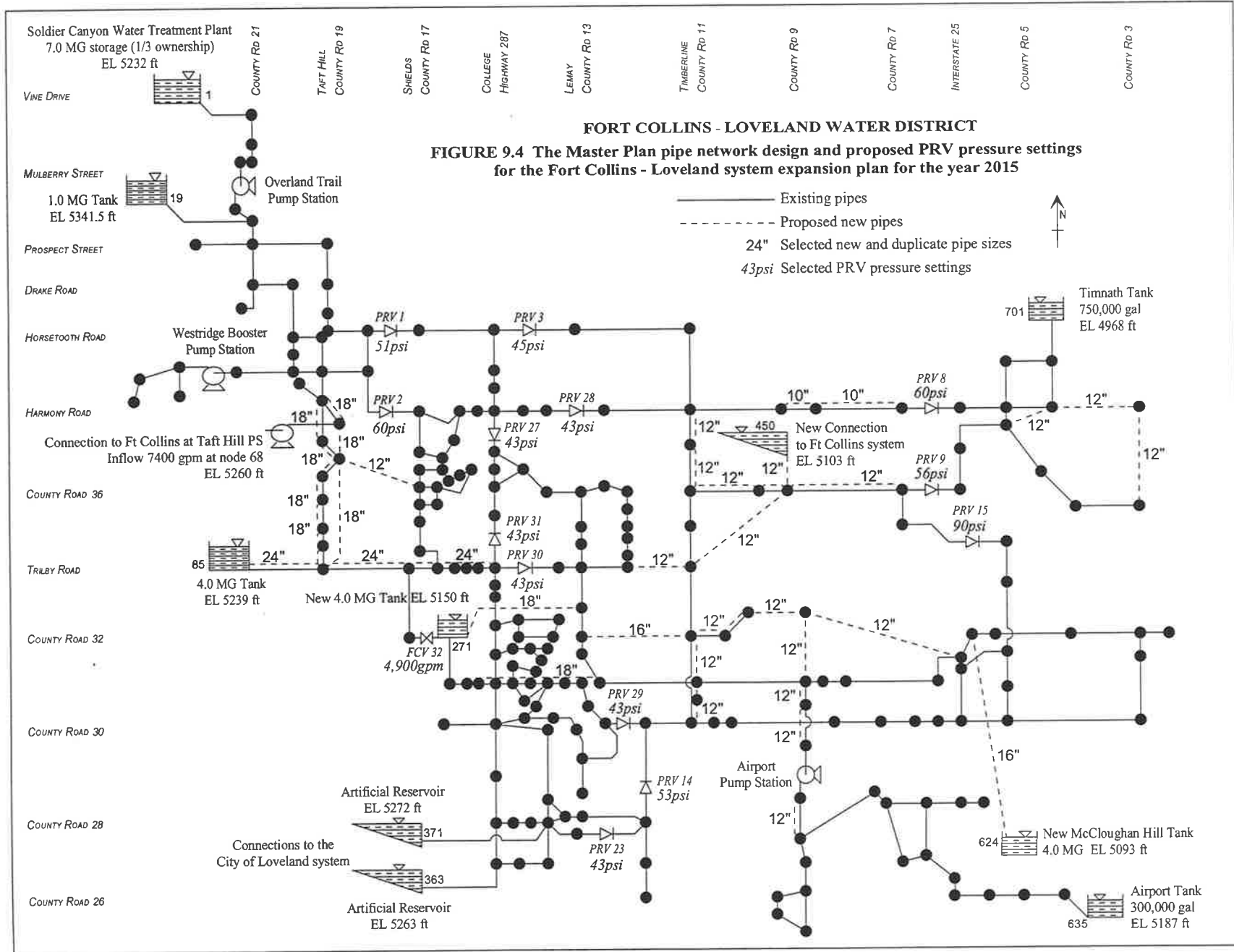


Table 9.4 The Master Plan pipe network design for 2015

Pipe	Pipe Description	Start node	End node	Length (ft)	Diameter (in)	Pipe Cost (\$)
NEW PIPES						
92	Clarendon Hills new pipe	76	241	5,280	12	158,400
448	New connection to Fort Collins system at County Rd 9, continuing southwest to Trilby/Timberline, extending to Paragon Pt	450	451	1,640	12	49,200
478		451	479	8,000	12	240,000
480		475	479	2,640	12	79,200
482	New pipe from new Zone 3 Tank to Lemay	271	474	10,000	18	450,000
484	New pipe along County Rd 32 between Lemay and Timberline	489	491	5,280	16	211,200
522	Duck Lake new pipes east of Timberline to Interstate 25 and south along County Rd 9	495	516	3,000	12	90,000
524		516	513	2,640	12	79,200
528		516	521	8,000	12	240,000
620	New pipe up to new McCloughan Hill Tank	624	522	12,000	16	480,000
733	County Rd 3 loop of new pipes south of Timnath (including County Rd 3 pipe between Walker Mfg and County Rd 36)	715	707	2,640	12	79,200
735		707	735	2,640	12	79,200
737		735	727	5,280	12	158,400
DUPLICATE PIPES						
83	Taft Hill duplicate pipes from Taft Hill Pump Station to Trilby Road	68	69	2,000	18	90,000
78		68	76	7,920	18	356,400
79		76	87	5,280	18	237,600
87		69	75	3,600	18	162,000
89		75	76	2,400	18	108,000
94		76	77	1,440	18	64,800
96		77	79	1,060	18	47,700
98		79	81	3,840	18	172,800
102		81	83	215	18	9,675
104		83	87	200	18	9,000
100	Trilby Road duplicate pipes between the existing 4.0 MG Tank and College Ave	85	87	2,640	24	158,400
272		87	253	5,280	24	316,800
274		253	257	210	24	12,600
264		257	259	2,800	24	168,000
266		259	261	500	24	30,000
270		261	263	2,000	24	120,000
276		263	469	10	24	600
310	Duplicate pipe south of new Zone 3 Tank between College and Lemay	295	311	5,480	18	246,600
414	Harmony Road duplicate pipes between County Rd 7 and County Rd 9	411	415	210	10	5,250
418		415	417	5,280	10	132,000
442	Timberline duplicate pipes between Harmony and County Rd 36	409	443	5,280	12	158,400
452		443	447	210	12	6,300
454	County Rd 36 duplicate pipes between Timberline and County Rd 7	447	449	2,640	12	79,200
456		451	449	2,640	12	79,200
458		451	453	5,280	12	164,400
504	Duplicate pipes along Timberline between County Rd 32 and County Rd 30	491	497	2,640	12	79,200
506		505	497	2,430	12	72,900
510		501	505	210	12	6,300
516	Duck Lake duplicate pipes east of Timberline	491	493	2,640	12	79,200
518		493	495	1,320	12	39,600
526	County Rd 9 duplicate pipes between County Rd 30 1/2 and County Rd 28	513	512	2,588	12	77,640
601		512	517	1,545	12	46,350
610		603	609	4,000	12	120,000
TOTAL PIPE COSTS						\$ 5,851,000

9.8.2 The Genetic Algorithm pipe network design

The genetic algorithm search is employed to optimise the new and duplicate pipe diameters and the pressure reducing valve (PRV) settings for the Fort Collins - Loveland system expansion plan. The coded structures representing GA designs consisted of strings of integer numbers as shown in Figure 9.5. The strings were constructed of 66 integer numbers including:

- 13 integer numbers representing the set of new pipe diameters
- 40 integer numbers representing the set of possible duplicate pipe diameters
- 13 integer numbers representing the set of PRV pressure settings

The first 13 integer positions for new pipes could take on integer values between 1 and 10, the integer values mapping to the ten available pipe sizes for new pipes according to Table 9.3 (minimum pipe diameter of 6"). The 40 integer positions for duplicate pipes could take on integer values between 1 and 13 representing the 12 available pipe sizes (including the 3" and 4" pipe diameters) and the 'do nothing' option as shown in Table 9.3. The last 13 integer positions corresponding to PRV pressure settings could take on a number of integer values depending on the range of allowable pressure settings for each individual PRV. For example, there are 6 possible PRV settings for *PRV 23* and there are 22 possible PRV settings for *PRV 8* (Table 9.3). The PRV pressure settings are considered further in Section 9.9. In Figure 9.5, the coded string solutions are presented for the proposed Master Plan pipe network design and the GA optimised pipe network expansions. The coded strings represent two solutions in an immense solution space of approximately 5×10^{71} possible solutions.

The integer codes were used in preference to the binary alphabet (binary codes or Gray codes) for this case study. If the problem is formulated using substrings of binary code to represent the possible choices for decision variables, there would be a large number of redundant binary codes (since the number of available pipe diameters and allowable PRV pressure settings are not a power of 2) and the size of the search space would increase significantly.

The GA design is presented in Figure 9.6. Table 9.5 summarises the GA pipe network design including the alternative routes for duplicate pipes considered by the GA. The alternative routes included: (i) the connections to the Loveland system, (ii) the transmission line south of Trilby Road to the flow control valve in the fill line to the new Zone 3 Tank, and (iii) the transmission line below the new Zone 3 Tank to the south and then east to College Avenue. These existing pipes were observed to be operating with high velocities and high head losses in the Master Plan design for the 2015 peak hour demands. Of the additional possible duplicate pipe locations investigated, the GA duplicated only the two pipes which connect to the City of Loveland system.

In the GA pipe network design in Figure 9.6, several of the new 12 inch diameter pipes proposed in the Master Plan have been re-sized to 6 inch diameter (i.e., the minimum pipe size considered by the GA for new pipes). The results of simulating the proposed GA system expansions subject to the 2015 peak hour demand pattern using the EPANET (Rossman, 1994) hydraulic simulation model is provided in Appendix B. The GA design produces a flow pattern similar to the Master Plan design, exhibiting adequate pressures throughout the system when subjected to the same 2015 peak hour demands.

Table 9.5 shows that the GA sizes all but 4 of the 13 proposed new pipes at the minimum diameter of 6 inches. The significant new pipes include: (1) the new pipe to Clarendon Hills between Taft Hill and Shields (between nodes 76 and 241) with diameter 12 inches; (2) the new connection to the Fort Collins system at County Road 9 (node 450) with diameter 10 inches; (3) the new pipe from the new Zone 3 Tank (node 271) east to Lemay (node 474) with diameter 24 inches; (4) and the new pipe up to the new McCloughan Hill Tank (node 624) with diameter 16 inches. If new pipes could be eliminated by meeting demands in the area via existing or duplicate pipes, then the GA might identify some of the 6 inch pipes as being unnecessary, and the total cost could be further reduced.

The GA design suggests duplication of only 9 pipes from the possible 40 considered. The significant duplicate pipes are located: (1) between the Taft Hill pump station and Trilby Road; (2) between the existing Zone 2 4.0 MG Tank (node 85) and College; (3) the connections to the City of Loveland system; (4) and along County Road 36 between the new connection to the City of Fort Collins system and the Timnath pressure zone.

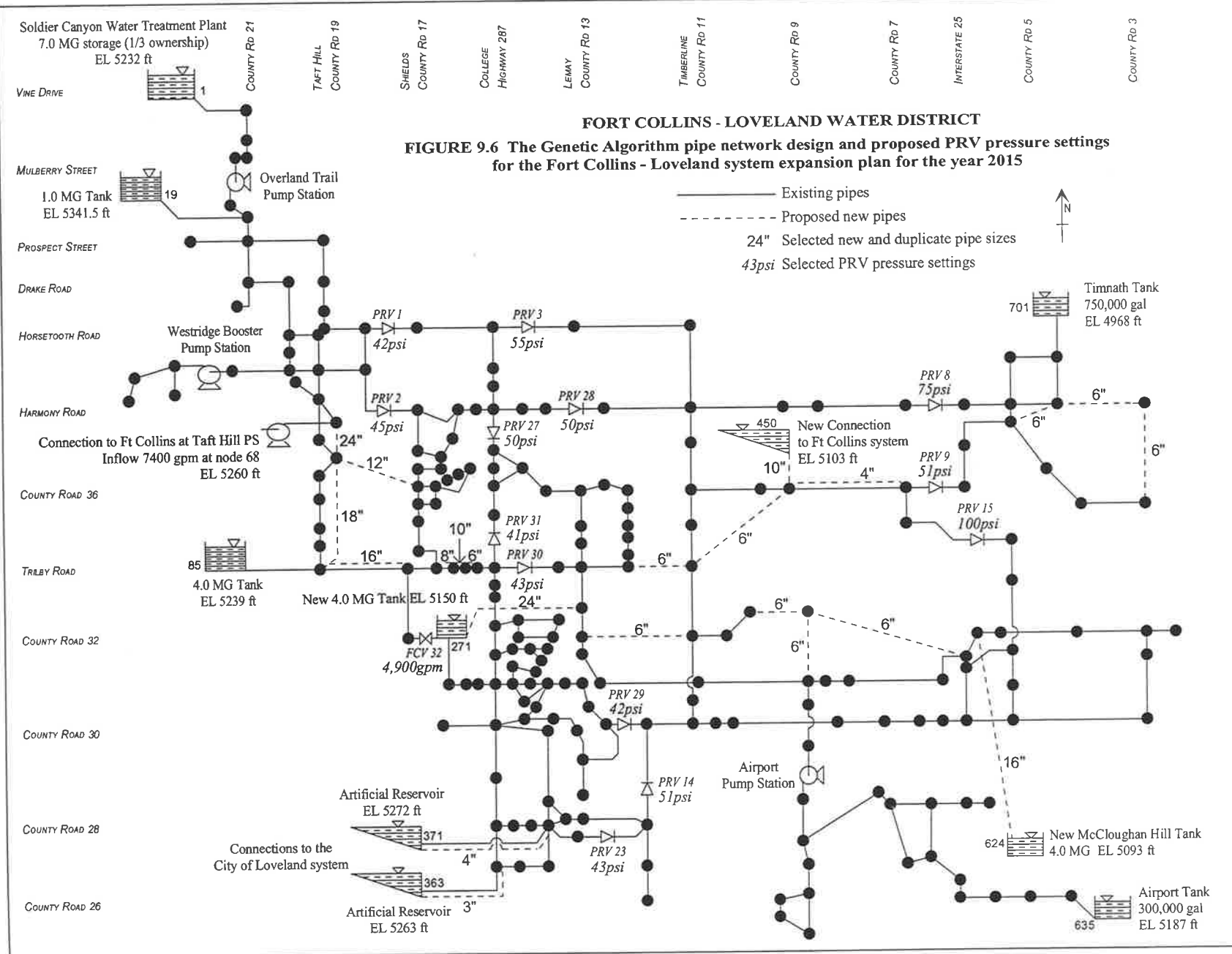


Table 9.5 The GA pipe network design for 2015

Pipe	Pipe Description	Start node	End node	Length (ft)	Diameter (in)	Pipe Cost (\$)
NEW PIPES						
92	Clarendon Hills new pipe	76	241	5,280	12	158,400
448	New connection to Fort Collins system at	450	451	1,640	10	41,000
478	County Rd 9, continuing southwest to	451	479	8,000	6	120,000
480	Trilby/Timberline, extending to Paragon Pt	475	479	2,640	6	39,600
482	New pipe from new Zone 3 Tank to Lemay	271	474	10,000	24	600,000
484	County Rd 32 between Lemay and Timberline	489	491	5,280	6	79,200
522	Duck Lake new pipes east of Timberline to	495	516	3,000	6	45,000
524	Interstate 25 and south along County Rd 9	516	513	2,640	6	39,600
528		516	521	8,000	6	120,000
620	New pipe up to new McCloughan Hill Tank	624	522	12,000	16	480,000
733	County Rd 3 loop of new pipes south	715	707	2,640	6	39,600
735	of Timnath (including County Rd 3 pipe	707	735	2,640	6	39,600
737	between Walker Mfg and County Rd 36)	735	727	5,280	6	79,200
DUPLICATE PIPES						
83	Taft Hill duplicate pipes from	68	69	2,000	0	-
78	Taft Hill Pump Station to Trilby Road	68	76	7,920	24	475,200
79		76	87	5,280	18	237,600
87		69	75	3,600	0	-
89		75	76	2,400	0	-
94		76	77	1,440	0	-
96		77	79	1,060	0	-
98		79	81	3,840	0	-
102		81	83	215	0	-
104		83	87	200	0	-
100	Trilby Road duplicate pipes between the	85	87	2,640	0	-
272	existing 4.0 MG Tank and College Ave	87	253	5,280	16	211,200
274		253	257	210	0	-
264		257	259	2,800	8	56,000
266		259	261	500	10	12,500
270		261	263	2,000	6	30,000
276		263	469	10	0	-
310	South of new Zone 3 Tank (College to Lemay)	295	311	5,480	0	-
414	Harmony Road duplicate pipes between	411	415	210	0	-
418	County Rd 7 and County Rd 9	415	417	5,280	0	-
442	Timberline duplicate pipes between	409	443	5,280	0	-
452	Harmony and County Rd 36	443	447	210	0	-
454	County Rd 36 duplicate pipes between	447	449	2,640	0	-
456	Timberline and County Rd 7	451	449	2,640	0	-
458		451	453	5,280	4	54,800
504	Duplicate pipes along Timberline between	491	497	2,640	0	-
506	County Rd 32 and County Rd 30	505	497	2,430	0	-
510		501	505	210	0	-
516	Duck Lake duplicate pipes east of	491	493	2,640	0	-
518	Timberline	493	495	1,320	0	-
526	County Rd 9 duplicate pipes between	513	512	2,588	0	-
601	County Rd 30 1/2 and County Rd 28	512	517	1,545	0	-
610		603	609	4,000	0	-
ADDITIONAL DUPLICATE PIPES CONSIDERED IN THE GA STUDY						
800	Duplicate pipes which connect to the	371	343	385	4	3,850
801	City of Loveland system	363	365	450	3	3,375
278	Duplicate pipes between Trilby Road and	253	267	5,280	0	-
282	the new Zone 3 Tank at County Rd 32	267	269	2,000	0	-
283		270	271	100	0	-
286	Duplicate pipes between new Zone 3 Tank	271	293	2,640	0	-
288	at County Rd 32 south and east to College	293	295	2,440	0	-
TOTAL PIPE COSTS						\$ 2,966,000

Table 9.6 presents a comparison of pipe costs between the Master Plan design and GA design. The total pipe cost for the GA design is \$ 2,966,000. This represents a cost saving of \$ 2,885,000, or 49.3% on the installed cost of new and duplicate pipes over the Master Plan design. Most of the savings (\$2.37 million) are achieved by reducing the number of pipes being duplicated and optimising their diameters.

Table 9.6 Summary of pipe costs

Design	Cost of new pipes	Cost of duplicate pipes	Total pipe cost
Master Plan design	\$2.394m	\$3.457m	\$5.851m
Genetic algorithm design	\$1.881m	\$1.085m	\$2.966m
SAVINGS	\$0.513m (21.4%)	\$2.372m (68.6%)	\$2.885m (49.3%)

9.9 Pressure Reducing Valve Settings

The Fort Collins - Loveland Water District system contains a number of pressure reducing valves (PRVs) and a flow control valve (FCV). The operation and analysis of PRVs was discussed in Section 2.10.9. There are three possible modes of operation of a PRV. The PRV is designed to maintain a constant pressure immediately downstream of the valve equal to the pressure setting (the operative mode). If the pressure upstream of the valve is less than the valve pressure setting, the flow through the valve is unrestricted (the inoperative mode). If the pressure downstream of the valve is greater than the pressure upstream of the valve, the valve shuts to prevent reverse flow (the shut check valve mode). FCVs limit the flow through the valve to the specified flow setting. The flow setting (4,900 gpm) of the FCV in the Fort Collins - Loveland system was not optimised by the GA.

The pressure reducing valves separate the system into pressure zones and maintain pressures within acceptable limits. The District is separated into five pressure zones; the elevation of the service area ranges from 5,280 feet to 4,800 feet (above mean sea level) across the system. The pressure zones are as follows:

- *1 MG Tank Zone 1* The highest pressure zone served from the 1.0 MG Tank (node 19)
- *4 MG Tank Zone 2* This pressure zone is served from the existing 4.0 MG Tank (node 85); PRVs 1 and 2 separate this zone from the adjacent highest Zone 1
- *PRV Zone 3* The intermediate PRV pressure zone is served from the new Zone 3 Tank (node 271) and the new McCloughan Hill Tank (node 624); it is created using PRVs 3, 14, 27, 28, 29, 30 and 31

9 GA optimisation of the water system expansion plan for the FCLWD

- *Timnath Zone 4* The lowest pressure zone is served from the Timnath Tank (node 701); PRVs 8 and 9 separate the PRV Zone 3 from the Timnath Zone 4
- *Airport Zone 5* The Airport pressure zone is served from the elevated Airport Tank (node 635)

The Master Plan identified redundant valves, relocated some existing valves, and adjusted the settings for some valves for the 2015 system layout. The recommendations of the Master Plan for 2015 are summarised in Table 9.7.

Table 9.7 Master Plan recommendations for PRVs

Valve	Nodes	Pressure setting (psi)	Discussion (from the Master Plan)
<i>PRV 1</i>	47-201	51	
<i>PRV 2</i>	73-209	60	The system cannot achieve 60 psi upstream of <i>PRV 2</i> until the new Clarendon Hills pipe is constructed.
<i>PRV 3</i>	207-401	45	The Master Plan suggested relocation of the valve (1/8th mile east) to the zone boundary would be ideal but is not necessary.
<i>PRV 4</i>	217-219	-	The valve should be relocated to <i>PRV 27</i> to the south of Harmony. It is hoped the valve relocation will improve pressures in Clarendon Hills and Fairway Estates.
<i>PRV 7</i>	413-415	-	The Master Plan suggested the valve be replaced with an altitude control valve located 1/8th mile west of County Rd 7 to control the operation of the Timnath tank.
<i>PRV 8</i>	419-709	60	<i>PRV 8</i> is modelled although the Master Plan suggested this valve be replaced with an altitude control valve as <i>PRV 7</i> is replaced.
<i>PRV 9</i>	455-721	56	The 1993 pressure setting (50 psi) is necessary at least until the duplication of the small pipes east of I25.
<i>PRV 10</i>	513-505	-	This valve is replaced by <i>PRV 29</i> west of County Rd 11.
<i>PRV 11</i>	507-509	-	This valve is replaced by <i>PRV 29</i> .
<i>PRV 12</i>	533-535	-	This valve to be abandoned when <i>PRV 10</i> and <i>PRV 11</i> are abandoned.
<i>PRV 13</i>	443-445	-	This valve should be abandoned when <i>PRV 4</i> is relocated and new <i>PRV 28</i> is introduced.
<i>PRV 14</i>	361-511	53	
<i>PRV 15</i>	483-729	90	The Master Plan acknowledged the pressure setting of 90 psi may be too high.
<i>FCV 16</i>		-	The flow control valve is no longer needed and is not modelled.
<i>PRV 17</i>	255-257	-	The Master Plan suggested <i>PRV 17</i> be eliminated.
<i>PSV 18</i>	295-297	-	The pressure sustaining valve is not modelled because the line is converted to a low head transmission line from the new Zone 3 Tank.
<i>PRV 19</i>	514-515	-	This valve should be abandoned when <i>PRV 10</i> and <i>PRV 11</i> are replaced by <i>PRV 29</i> .
<i>PRV 20</i>	266-265	-	This valve should be relocated to the Trilby intersection. <i>PRV 30</i> and <i>PRV 31</i> are introduced to replace <i>PRV 20</i> .
<i>PRV 23</i>	355-357	43	
<i>PRV 27</i>	222-421	43	This valve is <i>PRV 4</i> relocated.
<i>PRV 28</i>	226-406	43	New valve required as a consequence of the relocation of <i>PRV 4</i> .
<i>PRV 29</i>	332-498	43	New valve replaces <i>PRV 10</i> and <i>PRV 11</i> .
<i>PRV 30</i>	468-470	43	This valve is <i>PRV 20</i> relocated.
<i>PRV 31</i>	264-437	43	Introduced to replace <i>PRV 20</i> .
<i>FCV 32</i>	269-270	4,900*	A new flow control valve controls the flow from Zone 2 to the new Zone 3 tank west of College Avenue. The current flow setting is 4,900 gpm.

* flow setting (gpm) for the flow control valve

The GA is used in this study to determine optimal settings of the PRVs operating in the system. The procedure used to accomplish this is described below. The GA optimisation has assumed the same locations for the PRVs as recommended in the Master Plan; no optimisation of PRV locations was attempted.

To maintain the structure of the pressure zones outlined in the Master Plan, the PRV pressure settings should be chosen such that the static pressure in the zone below is between 50 psi and 100 psi. The range of pressure settings considered for the PRVs is determined by the range of elevations of nodes downstream of each PRV (see Table 9.8). Generally, the furthest downstream node with the highest ground elevation will determine the minimum PRV pressure setting to provide a static pressure of at least 50 psi. Similarly, the node downstream of lowest ground elevation will determine the maximum PRV pressure setting so that the static pressure does not exceed 100 psi. The range of PRV pressure settings considered by the GA in Table 9.8 approximates the allowable range of PRV pressure settings. The GA considered a step in pressure setting of 1 psi in the range of allowable pressure settings. The maximum and minimum downstream elevations in Table 9.8 reasonably match the high and low service area elevations for the pressure zones given in the Master Plan.

Table 9.8 Range of pressure settings considered in GA optimisation for PRVs

Valve	PRV elevation (ft)	Maximum downstream elevation (ft)	Minimum allowable pressure setting (psi)	Minimum downstream elevation (ft)	Maximum allowable pressure setting (psi)	Pressure setting range considered by GA (psi)
PRV 1	5110	5082	37.9	5000	52.4	40 → 55
PRV 2	5090	5085	47.9	4960	43.7	45 → 60
PRV 3	5025	4985	32.7	4920	54.6	40 → 55
PRV 8	4845	4850	52.2	4842	98.7	55 → 75
PRV 9	4850	4850	50.0	4843	97.0	50 → 70
PRV 14	4980	4970	45.7	4870	52.4	45 → 55
PRV 15	4850	4970	101.9	4835	93.5	90 → 100
PRV 23	5005	4980	39.2	4870	41.6	40 → 45
PRV 27	5000	5000	50.0	4870	43.7	40 → 50
PRV 28	5000	4990	45.7	4850	35.1	40 → 50
PRV 29	5000	4970	37.0	4870	43.7	40 → 45
PRV 30	5000	4990	45.7	4870	43.7	40 → 50
PRV 31	5000	5000	50.0	4870	43.7	40 → 50

The PRV pressure settings which result in static pressures of just less than 50 psi or just greater than 100 psi should be acceptable since in general, the high elevations in each pressure zone are not far downstream of the PRVs (so there are insignificant head losses), while the low

elevations of the pressure zone occur some distance downstream of the PRVs (with significant head losses).

A comparison of the pressure settings for the PRVs chosen by the Master Plan and by the GA are summarised in Table 9.9. *PRV 15* is not operating for the Master Plan design subject to the 2015 peak hour demands, since the system cannot maintain upstream pressures higher than the PRV pressure settings. By comparison, *PRV 8* and *PRV 15* are not operational for the GA design subject to the 2015 peak hour demands. *PRV 23* and *PRV 29* are closed and operating as check valves for the Master Plan design subject to the 2015 peak hour demands, to avoid flow reversal. *PRV 23*, *PRV 30* and *PRV 31* are closed for the GA design subject to the 2015 peak hour demands. It is likely *PRV 30* and *PRV 31* are closed due to the increased capacity to this area from the new Zone 3 Tank via node 474 and node 473.

Table 9.9 Pressure reducing valves and selected pressure settings

Valves	Nodes	Master Plan PRVs		Genetic algorithm PRVs	
		Pressure setting (psi)	Status for 2015 peak hour	Pressure setting (psi)	Status for 2015 peak hour
<i>PRV 1</i>	47-201	51	Operating	42	Operating
<i>PRV 2</i>	73-209	60	Operating	45	Operating
<i>PRV 3</i>	207-401	45	Operating	55	Operating
<i>PRV 8</i>	419-709	60	Operating	75	Not operating
<i>PRV 9</i>	455-721	56	Operating	51	Operating
<i>PRV 14</i>	361-511	53	Operating	51	Operating
<i>PRV 15</i>	483-729	90	Not operating	100	Not operating
<i>PRV 23</i>	355-357	43	Closed	43	Closed
<i>PRV 27</i>	222-421	43	Operating	50	Operating
<i>PRV 28</i>	226-406	43	Operating	50	Operating
<i>PRV 29</i>	332-498	43	Closed	42	Operating
<i>PRV 30</i>	468-470	43	Operating	43	Closed
<i>PRV 31</i>	264-437	43	Operating	41	Closed

9.10 System Performance of Master Plan and GA Design

In the Master Plan, the existing and predicted land use plans and population projections for the District were used to estimate water demands through to year 2015. The maximum daily demands for 2015 were predicted and peak hour demands for 2015 were derived from these. The 2015 designs are required to supply the 2015 peak hour demands while satisfying the District’s system performance requirements. The net 2015 peak hour demand on the system analysed in the EPANET hydraulic simulation is about 27,240 gpm (39.2 MGD).

The GA evaluates the hydraulic feasibility of trial GA designs by checking that the various system performance requirements are not violated. The designs which are infeasible are not discarded by the GA search. The infeasible designs are penalised by adding a penalty cost to the computed pipe cost to arrive at an equivalent total cost for that trial design. The penalty cost applied is a function of the magnitude of the violations of the hydraulic constraints. The GA does not discard infeasible designs as they may represent very good designs if the violations are small. Subsequent generated trial designs may incorporate the best features of these nearly feasible designs as the GA search procedure progresses.

The KYPIPE hydraulic simulation model was integrated into the GA model to evaluate the hydraulic feasibility of the proposed GA designs. The EPANET hydraulic simulation model was used as a check in this study to predict the pipe flows and node pressure heads for both the Master Plan design and the proposed GA design subject to 2015 peak hour demands (see Appendix A and Appendix B respectively).

9.10.1 Node pressures and pipe velocities

For the peak hour demands, the District's water system is designed to maintain pressures between 40 psi and 100 psi. The GA evaluation procedure penalises designs with nodes having pressures less than 40 psi. A small number of nodes are allowed pressures less than 40 psi (following some discussion with FCLWD regarding the performance of the Master Plan design) as detailed in Table 9.10.

The minimum pressure constraint of 40 psi is not applied to transmission lines which are low head, high volume systems that do not provide domestic service, however, the pressures in the transmission system should not be negative. The pressures at some of the critical nodes in Table 9.10 would be improved if the pump stations which serve them are modelled. The pressure at node 337 is 39.9 psi for the GA design which represents a very small violation of the minimum pressure constraint. The GA design shows an overall reduction of pressures in the system compared to the Master Plan design. The average pressure for the GA design subject to the 2015 peak hour demands is 60.60 psi, while the average pressure for the Master Plan design is 67.72 psi.

Table 9.10 Summary of low pressures for the proposed designs

Node	Critical pressures (psi)		Discussion
	Master Plan Design	GA Design	
3	35.66	35.04	Nodes 3, 5, 7 and 9 between Soldier Canyon and Overland Trail experience low pressures. These nodes are considered to be part of the transmission system such that pressures less than 40 psi are considered acceptable.
5*	36.53	35.77	
7*	18.11	17.21	
9	28.98	28.05	
29	33.13	29.71	Node 29 is served by the Burns Ranch booster PS which has not been modelled.
60*	21.26	18.41	Node 60 is served by the Westridge booster PS which is modelled with only one pump on.
269	7.04	2.17	Nodes 269 and 270 represent the fill line to the proposed new Zone 3 tank.
270	6.75	6.75	
293*	-0.08	4.33	Nodes 293 and 295 are considered to be part of the transmission system.
295	30.51	39.00	
317*	29.40	12.13	Nodes 317 and 319 will be part of a tank system served by the County Rd 32 booster PS which has not been modelled.
319*	50.00	32.74	

* nodes with demands

A small number of nodes are allowed pressures greater than 100 psi. The high pressures are not so high as to warrant changes to the system design. In such circumstances, the District would typically install individual domestic service line PRVs. For the Master Plan design subject to the 2015 peak hour demands, there are five nodes with pressures greater than 100 psi. The highest pressure is 112.4 psi at node 233. For the GA design, there are three nodes with pressures greater than 100 psi and the highest pressure is 106.4 psi at node 545.

The District would prefer to keep pipe velocities below 5 fps. The District acknowledges some pipes (mostly transmission lines) have velocities which exceed 5 fps. Pipes with velocities greater than 5 fps may require special design consideration for events such as those caused by water hammer. There are 14 pipes for the Master Plan system and 31 pipes for the GA system which exceed a velocity of 5 fps for the 2015 peak hour. The average pipe velocity is 1.93 fps for the Master Plan design and is 2.47 fps for the GA design.

9.10.2 Supply flows, transmission flows, tank outflows and inflows

The flows to and from pressure zones including supply flows, tank outflows and inflows, and significant transmission flows are summarised in Table 9.11 for both the Master Plan design and the GA design for the 2015 peak hour demands.

Table 9.11 Pressure zone inflows for the 2015 peak hour demands

Pressure Zone	Nodes	Pipe description	Pipe flows (gpm)	
			Master Plan	GA
Zone 1	1-3	Supply from Soldier Canyon	4,557.3	4,647.5
	19-17	Tank flow from 1.0 MG Tank	1,932.2	2,480.9
	68	Supply from Taft Hill PS	7,400.0	7,400.0
Zone 2	68-87	Transmission flow from Taft Hill PS	6,283.1	5,648.2
	85-87	Tank flow from 4.0 MG Tank	3,556.3	1,566.3
	47-201	Flow via PRV 1	2,097.9	3,112.9
	73-209	Flow via PRV 2	352.7	199.6
	270-271	FCV 32 to new Zone 3 Tank	4,900.0	4,900.0
Zone 3	271-474	Tank flow east from new Zone 3 Tank	4,179.8	7,538.1
	271-293	Tank flow south from new Zone 3 Tank	3,917.9	6.0
	371-343	Supply from existing Loveland connection	1,572.4	2,433.8
	363-365	Supply from existing Loveland connection	510.2	963.0
	450-451	Supply from new Fort Collins connection	1,932.0	1,813.9
	624-522	Tank flow from McCloughan Hill Tank	930.5	1,390.7
	207-401	Flow via PRV 3	125.3	137.9
	361-511	Flow via PRV 14	78.5	82.0
	483-729	Flow via PRV 15	44.2	28.8
	355-357	Flow via PRV 23	0.0	0.0
	222-421	Flow via PRV 27	1,086.9	2,254.8
	226-406	Flow via PRV 28	491.3	571.2
	332-498	Flow via PRV 29	0.0	135.0
	468-470	Flow via PRV 30	423.1	0.0
	264-437	Flow via PRV 31	1,019.6	0.0
Zone 4	701-703	Tank flow from Timnath Tank	-21.0	153.8
	419-709	Flow via PRV 8	571.7	407.8
	455-721	Flow via PRV 9	87.3	76.4
Zone 5	517-603	Flow to Airport zone via Airport PS	890.8	810.1
	635-633	Tank flow from elevated Airport Tank	1,673.2	1,753.9

The Master Plan design supplies about 2,080 gpm from the City of Loveland system connections and about 1,930 gpm from the new City of Fort Collins system connection at County Road 9. The District indicated a desire to limit the inflows from the Loveland systems to 5.0 MGD (3,470 gpm) and from the Fort Collins system to 5.0 MGD (3,470 gpm). The GA considered the option to duplicate the pipes connected to the Loveland system. Duplication of these pipes results in the supply from Loveland being increased to 3,400 gpm. The relatively short sections of existing pipe connected to the Loveland system operate with high velocities and experience high head losses. The diameter of the new pipe connection to the Fort Collins system has also been determined by the GA.

The GA design constrains the inflows to the City systems to the specified maximum rate of 5.0 MGD (3,472 gpm) for the 2015 peak hour demand pattern by way of a penalty function. The GA design draws about 3,400 gpm from the Loveland system and about 1,800 gpm from the Fort Collins system. The GA design chooses to duplicate the connections to Loveland and increases the flow from Loveland up to close to the maximum allowable flow rate. The cost of purchasing the additional water from Loveland may be significant or may be offset by cost

savings elsewhere. The GA could be set up to consider this trade-off in an analysis to determine optimal water sources.

The GA design provides virtually no water to the south from the new 4.0 MG Zone 3 Tank (node 271). The demands in this area are met with increased flows from Loveland. The new pipe to the east of the new Zone 3 Tank is assigned a diameter of 24 inches by the GA (compared with 18 inches in the Master Plan) and consequently, flows from the new Zone 3 Tank are high in this direction.

The GA design supplies an amount of water from the new connection to the Fort Collins system (node 450) similar to the amount supplied in the Master Plan design. The GA design reduces the sizes of the new pipe connected to the Fort Collins system and the new and possible duplicate pipes distributing the flow.

The total flow from external sources (Soldier Canyon, Taft Hill, City system connections) is 17,260 gpm for the GA design compared with 15,970 gpm for the Master Plan design. The increased supply to the system reduces the total demand met by the system's tanks. There is a significant reduction in the flow requirements of the existing 4.0 MG Tank (node 85). The Timnath Tank is emptying for the GA design subject to the 2015 peak hour demands while the tank is slowly filling for the Master Plan design.

9.10.3 Taft Hill source pump station

The Master Plan recommends the District purchase a standby pump unit for the Taft Hill source pump station and duplicate the transmission line between Taft Hill and the existing Zone 2 4.0 MG Tank (node 85). The reliable capacity of the pump station would be increased to about 7,400 gpm with these improvements.

The Taft Hill source pump station can supply 7,400 gpm. The supply is provided as a negative node demand at node 68. The hydraulic grade line (HGL) at node 68 for the Master Plan design subject to the 2015 peak hour demands is approximately 5,260 ft. It has been assumed in the GA design that the HGL at node 68 should be maintained at about 5,260 ft for the GA design to simulate the pump head achieved by the pump station for the design pump discharge of 7,400 gpm.

The GA constrains the HGL at node 68 to less than 5,260 ft by way of a penalty function. The GA tends to provide a HGL at node 68 close to 5,260 ft by the optimisation of the duplicate pipe sizes between Taft Hill and the existing Zone 2 4.0 MG Tank. The GA design duplicates

the pipe from node 68 to node 76 with a 24 inch diameter pipe and from node 76 to node 87 with an 18 inch diameter pipe. The resulting HGL at node 68 is 5,260.6 ft.

In future studies it would be useful to simulate the exact operation of the Taft Hill pump station. This would allow the GA to determine the best operating point on the pump stations characteristic curve, rather than be restricted to a specified operating point.

9.11 Extended Period Simulation (EPS)

A hypothetical extended period simulation (EPS) of the projected 2015 maximum day demands is performed on both the Master Plan design and the GA design. The EPS analyses are only hypothetical because actual data was not available to do an accurate EPS. A series of assumptions were made in the EPS analysis, particularly with respect to the variation of demands and the water system operation for the maximum day in 2015. Despite these assumptions, the EPS analyses provide a fair comparison of the system behaviour of the Master Plan design and the GA design subject to an approximate maximum daily demand pattern.

The design outlined in the Master Plan is designed to supply and transfer the 2015 maximum day demands and the document describes some of the approaches to the design of transmission pipelines, pump station upgrades and storage tanks. The GA design was subjected to the 2015 EPS analysis to find out if it performs as effectively as the Master Plan design for an identical demand pattern and identical system operation.

9.11.1 Maximum day demands for 2015

The Master Plan assumes that the peak hour demand on the maximum day is equal to 1.5 times the average demand on the maximum day, based on experience in the U.S.A. The 2015 peak hour demands are given in Appendix A. Using this assumption, a typical three-hourly variation of demands has been constructed to approximate the maximum day demand pattern as shown in Table 9.12. The approximation of the maximum day demand pattern assumes a peak demand period between about 6am and 3pm a lower demand period through the night between 9pm and 6am.

Table 9.12 Demand variation assumed for EPS of the 2015 maximum day

Time of day	Multiple of average hour demand for maximum day	Multiple of peak hour demands
6am to 9am	1.3	0.866
9am to 12 noon	1.5*	1.0*
12 noon to 3pm	1.3	0.866
3pm to 6pm	1.1	0.733
6pm to 9pm	0.9	0.600
9pm to 12 midnight	0.7	0.466
12 midnight to 3am	0.5	0.333
3am to 6am	0.7	0.466

*peak hour demand

9.11.2 Variable head storage tanks

The EPS simulation of the 2015 maximum day demand pattern commences at 6am just before the high demand morning period. The initial tank water levels at 6am are assumed to be at the maximum tank water levels. In other words, the tanks are assumed to be full before the peak day begins. The new 4.0 MG Zone 3 Tank (node 271) and the new 4.0 MG McCloughan Hill Tank (node 624) are assumed to have identical dimensions to the existing 4.0 MG Tank (node 85) as shown in Table 9.13.

Table 9.13 Tank dimensions and initial water levels assumed for the EPS

Tank	Elevation (ft)	Starting water level (ft)	Minimum water level (ft)	Maximum water level (ft)	Diameter (ft)
1.0 MG	5,327.5	5,355.5	5,336.5	5,355.5	78
4.0 MG	5,221	5,249	5,230	5,249	156
Zone 3 Tank*	5,132	5,160	5,141	5,160	156
McCloughan Hill*	5,075	5,103	5,084	5,103	156
Airport	5,053	5,207	5,175	5,207	60
Timnath	4,958	4,992	4,958	4,992	40

* New tanks proposed by the Master Plan

In Table 9.14, the total volume of the tanks is compared with the available operating storage derived from the tank dimensions in the EPANET model data provided by the District. The EPANET simulation assumes the tanks are cylindrical. There is some disagreement between the total storage and available storage of the elevated Airport Tank. The EPANET data suggests an available operating volume of 680,000 gal while the Master Plan document suggests the total volume is only 300,000 gal. This is a concern due to the poor performance of the Airport Tank for the EPANET EPS (even when using an operating volume of 680,000 gal) of the assumed 2015 maximum day demands. This discrepancy would need to be resolved in future studies.

Table 9.14 Tanks available operating storage

Tank	Total volume (gal)	Available operating volume (gal)
1.0 MG Tank	1,000,000	680,000
4.0 MG Tank	4,000,000	2,720,000
New Zone 3 Tank	4,000,000	2,720,000
McCloughan Hill Tank	4,000,000	2,720,000
Airport Tank	300,000*	680,000
Timnath Tank	750,000	320,000

* an anomaly

Ideally, the balancing storages in the tanks would discharge to help the system supply the peak demands on the maximum day and the system would recharge the tanks during the night. The tank water levels are at the maximum operating levels at the start of the peak demand period. They are expected to fall and then rise again to their starting water levels by the end of the 24 hour extended period simulation. The tank water level variations are shown in Figures 9.7 to 9.12 and are summarised in Table 9.15.

Table 9.15 Tank water level variations for the EPS of the 2015 maximum day

Tank	Initial water level (ft)	Final water level (ft)		Lowest water level (ft)	
		Master Plan	GA	Master Plan	GA
1.0 MG	5,355.5	5,355.5*	5,355.5*	5,343.5	5,338.3
4.0 MG	5,249	5,241	5,249*	5,236.7	5,245
New Zone 3	5,160	5,141.9	5,143.9	5,141**	5,143.9
McCloughan Hill	5,103	5,095.6	5,092.6	5,095.6	5,092.6
Airport	5,207	5,183.6	5,180.3	5,176.7	5,175**
Timnath	4,992	4,979.9	4,992*	4,975.3	4,982.9

* Tank water level returns to initial water level

** Tank empties

The balancing storage in the *existing Zone 1 1.0 MG Tank* (Figure 9.7) is used effectively for both the Master Plan and GA designs. The tank discharges between 6am and 3pm and then recharges and the tank water level returns to the starting water level as the EPS proceeds.

At the end of the 2015 maximum day, the *existing Zone 2 4.0 MG Tank* (Figure 9.8) is full again for the GA design, but reaches only about 60% full for the Master Plan design.

The *new Zone 3 4.0 MG Tank* (Figure 9.9) discharges throughout the 2015 maximum day and is *empty* at midnight for the Master Plan design. The tank may not be empty, but the water level is at the minimum operating water level and is effectively empty for the purpose of the EPS. This tank does not completely empty for the GA design, however, at the end of the EPS, the tank water levels in the new Zone 3 Tank are low for both the GA design and the Master Plan design.

9 GA optimisation of the water system expansion plan for the FCLWD

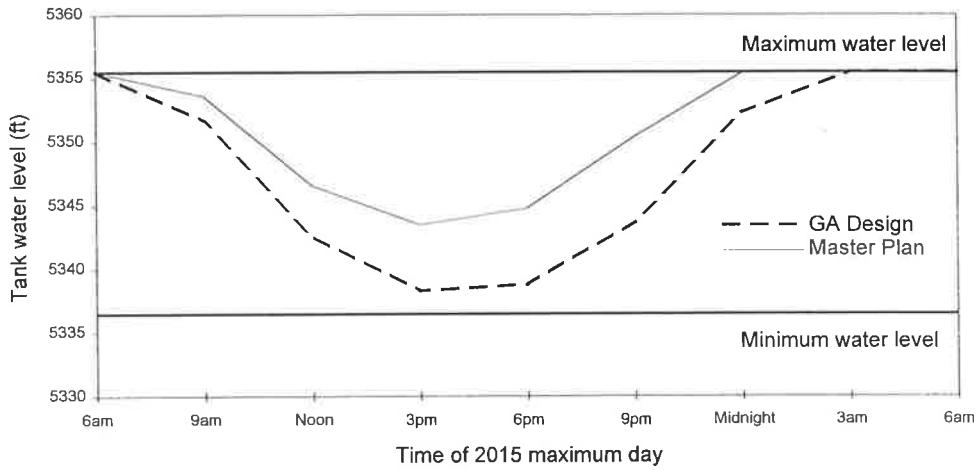


Table 9.7 Zone 1 1.0MG Tank water level variation for EPS

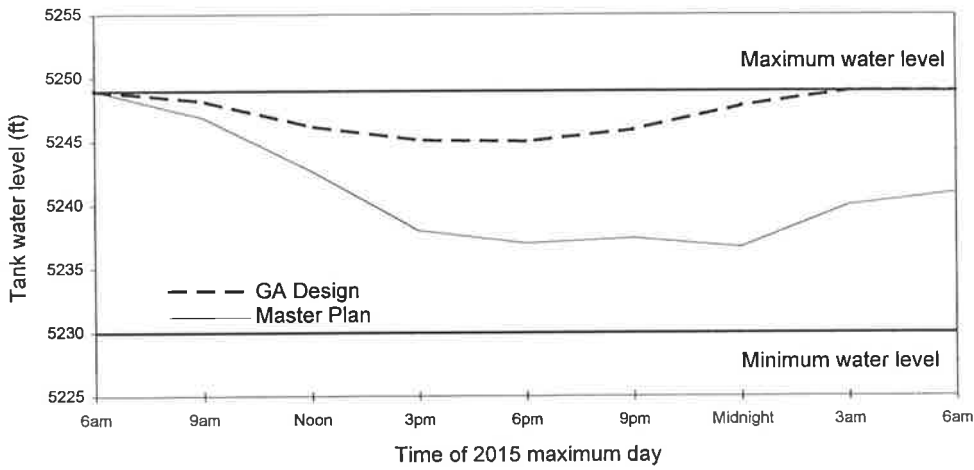


Table 9.8 Zone 2 4.0MG Tank water level variation for EPS

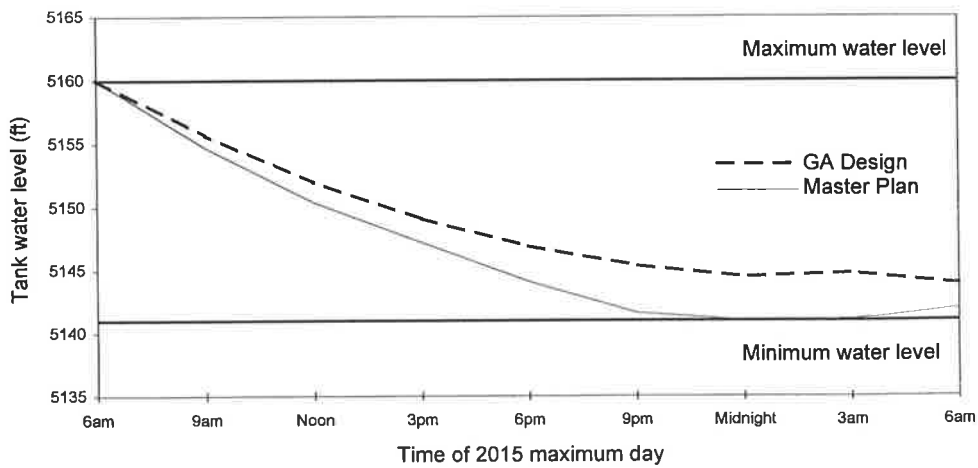


Table 9.9 New Zone 3 4.0MG Tank water level variation for EPS

9 GA optimisation of the water system expansion plan for the FCLWD

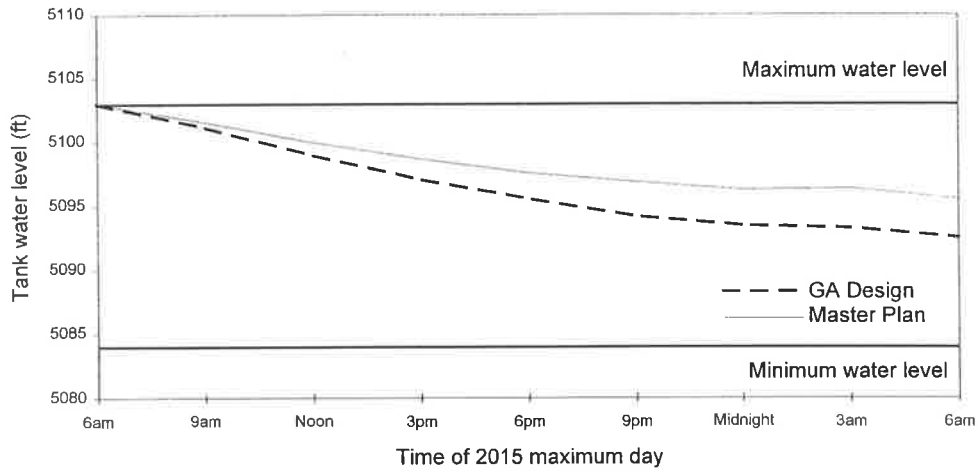


Table 9.10 New McCloughan Hill Tank water level variation for EPS

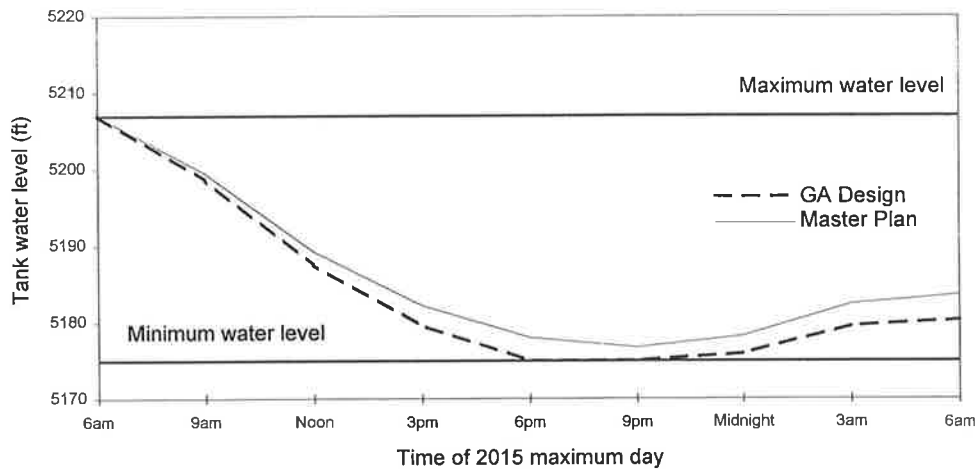


Table 9.11 Elevated Airport Tank water level variation for EPS

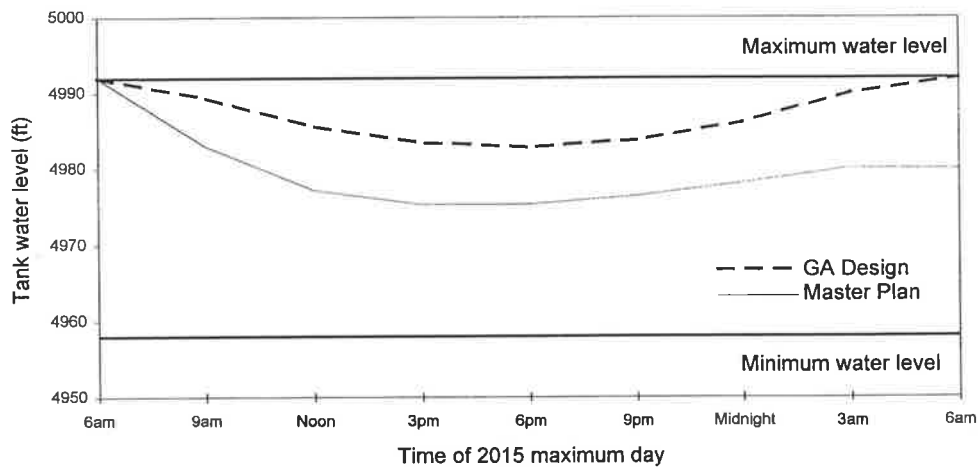


Table 9.12 Timnath Tank water level variation for EPS

The new *McCloughan Hill Tank* (Figure 9.10) steadily discharges throughout the 2015 maximum day. The discharge rate is faster for the GA design. At the end of the EPS, the McCloughan Hill Tank is about half full for both the GA design and the Master Plan design.

The *elevated Airport Tank* (Figure 9.11) drains quickly and is empty at 6pm for the GA design while it almost empties for the Master Plan design. The Airport Tank water level variation for the Master Plan design is similar to that of the GA design, except the tank does not empty. The tank does begin to refill again through the night and at the end of the 2015 maximum day is about 20% full for both designs.

At the end of the EPS, the *existing Timnath Tank* (Figure 9.12) is full again for the GA design but is only about 65% full for the Master Plan design.

The EPS analysis has shown that the performance of both the Master Plan design and the GA design are similar under EPS conditions. However, it is obvious that neither design is satisfactory and both designs would need to be re-evaluated to add additional pipes or tank volumes to satisfy the EPS of the 2015 maximum day. This could be considered in a future GA optimisation.

9.11.3 Pump station characteristics for EPS

The pump station characteristics for the 2015 maximum day EPS are assumed as shown in Table 9.16. In Table 9.16, H_0 is the pump shut-off head, H_1 and Q_1 are the rated head and rated discharge for the summed pump curves, H_2 and Q_2 are intermediate points on the curve, and Q_3 is the maximum pump station discharge.

Table 9.16 Pump station operation* for the EPS analyses

Pump Station	Operation	H_0 (ft)	H_1 (ft)	Q_1 (gpm)	H_2 (ft)	Q_2 (gpm)	Q_3 (gpm)
Overland Trail	4 pumps	220.0	190.0	4,000.0	170.0	5,000.0	8,000.0
Airport	3 pumps	215.0	180.0	600.0	160.0	1,200.0	1,800.0

* Taft Hill source PS operation is not modelled, however, the pumping station is operating 3 pumps with a combined design flow of 7,400 gpm

The Overland Trail source pump station and the Airport booster pump station are operated at maximum capacity all day for the maximum day. The characteristic curves used for the EPANET EPS analysis are derived from the Master Plan document. The pump head characteristic curve for the Overland Trail source pump station with four pumps on is shown in Figure 9.2 and the pump curve for the Airport booster pump station with three pumps on is given in Figure 9.3.

The Westridge booster pump station is modelled for the maximum day EPS with just one of the two existing pumps operating, in the same way that it was modelled for the peak hour analysis. Several other existing and proposed booster pump stations are not modelled for consistency as per the original Master Plan.

The operation of the Taft Hill source pump station is not modelled, but the pump station is assumed to supply the reliable capacity of 7,400 gpm all day during the maximum day. The supply from the connection to the City of Fort Collins system via the Taft Hill pump station is simulated as an inflow (negative demand) at node 68 for the EPS using the EPANET model.

The Master Plan design subject to the 2015 peak hour demands simulates the Taft Hill source pump station as an inflow of 7,400 gpm at node 68 and the corresponding HGL at node 68 is 5,260 ft. The EPS analyses only approximately simulates the operation of the Taft Hill pump station as the HGL at node 68 is not controlled. It is assumed that the HGL at node 68 should not exceed 5,260 ft. The Master Plan design generally maintains the HGL at the Taft Hill connection between 5,260 ft and 5,270 ft during the EPS. For the GA design, the HGL at the connection generally varies between 5,270 ft and 5,290 ft during the EPS. These aspects should be corrected in future studies. One demand period for the Master Plan EPS and two demand periods for the GA EPS exhibited much higher HGLs at the node 68 connection. It would be useful to simulate the exact operation of the Taft Hill pump station, for a more accurate EPS.

The EPS of the 2015 maximum day using the EPANET model determines the power output of the pump stations. The average power output of the pump stations are compared in Table 9.17 for the GA design and the Master Plan design.

Table 9.17 Average pump station power output for the 2015 maximum day

Pump Station	Average power output (horsepower)	
	Master Plan design	Genetic algorithm design
Overland Trail PS	209	199
Taft Hill PS	131	131
Airport PS	39.3	43.1
Westridge PS	3.1	3.1

The power output of the Taft Hill source pump station is based on the simulated pump flow of 7,400 gpm and corresponding pump head of approximately 70 ft. The power consumed by the pump stations will depend on the efficiency characteristic curves of the pumps. The average power consumed on the average demand day in 2015 could be used to approximate annual power costs for pumping.

9.11.4 The sources of supply

The Soldier Canyon Tanks are assumed to be maintained at full level by the water treatment plant for the whole day during the EPS. It is assumed the supply can match the demand throughout the maximum day. Soldier Canyon is modelled as a reservoir of constant elevation. In addition, the connections to the City systems are modelled as reservoirs of constant elevation. Again, it is assumed the supplies from the connections can meet the needs of the District’s maximum day. The water surface elevations of the reservoirs used in the computer models are given in Table 9.18.

Table 9.18 The HGL of sources of supply for EPS

Source modelled as reservoir	Assumed constant water surface elevation (ft)
Soldier Canyon (node 1)	5247
Loveland connection (node 363)	5263
Loveland connection (node 371)	5272
Fort Collins connection (node 450)	5103

The total supplies from the sources of supply for the 2015 maximum day are summarised in Table 9.19. The GA system design demands a total of 21.78 MGD from the combined sources. The Master Plan system demands a comparable 21.28 MGD. The GA system design takes more water from the City systems but less from Soldier Canyon as compared to the Master Plan design.

Table 9.19 Estimated total supplies from the sources of supply for EPS

Source of supply	Master Plan supply (MGD)	Genetic algorithm supply (MGD)
Soldier Canyon	6.99	6.49
Taft Hill	10.66	10.66
Loveland connection (node 363)	0.48	0.79
Loveland connections (node 371)	1.59	2.21
New Fort Collins connection	1.56	1.63
TOTAL	21.28	21.78

9.11.5 System performance for EPS

The District's water system is designed to ideally maintain pressures at nodes throughout the system between 40 psi and 100 psi. A small number of nodes are allowed pressures less than 40 psi under the peak hour loading condition (see Table 9.10). Over the period of the EPS, these are the same nodes which exhibit low pressures. The most critical of these nodes for the GA design subject to the 2015 maximum day demand pattern are perhaps node 269 (lowest pressure experienced is 5.96 psi) and node 270 (lowest pressure is 4.13 psi) on the fill line to the proposed new Zone 3 Tank, and node 293 (lowest pressure is 1.71 psi) which is part of the transmission system downstream of the new Zone 3 Tank. The GA design experiences low pressures in the Airport area at 6pm when the Airport Tank is empty. The lowest pressure of 22.55 psi occurs at node 619 in the Airport area.

The Master Plan design subject to the 2015 maximum day demands shows a similar pattern of pressures. The Airport Tank gets low but does not empty. The pressure at node 293 is critical since it is as low as -2.2 psi at 9pm. This node 293 is part of the transmission system from the new Zone 3 Tank. Negative pressures are not acceptable.

9.12 Summary and Conclusions

In this study, genetic algorithm optimisation has successfully been applied to the design of the large-scale Fort Collins - Loveland Water District's water distribution system for the year 2015 expansion plan. The GA technique was used to size new and duplicate pipes and to choose appropriate pressure settings for the pressure reducing valves in the system. The GA design is shown to satisfy system performance requirements, such as minimum pressures of 40 psi for the system subject to the 2015 peak hour demands. The cost of the new and duplicate pipes for the GA system design is \$2.966 million, which represents a savings of \$2.885 million (or 49.3%) compared to the Master Plan pipe network design with pipe costs of \$5.851 million.

Following the optimisation step, the GA design and the Master Plan design were both subjected to an extended period hydraulic simulation (EPS) using an assumed maximum day demand pattern for 2015 to evaluate the transmission system. The EPS was performed to allow a fair comparison between the two designs. Although the Master Plan system was designed with consideration given to sizing pipelines for transmission, the design failed to perform adequately for the EPS analysis. Not surprisingly, the GA design demonstrated similar unacceptable system performance for the EPS, under equivalent system operation to that of the Master Plan design. The GA design did perform slightly better in that three of the six storage tanks refilled on the maximum day, whereas only one tank refilled for the Master Plan design.

Extended period simulation was not a design criteria for the original Master Plan design and as a result was not used in the GA design process. The GA technique, however, has the capability of directly considering the EPS requirements to ensure all tanks refill at the end of a peak demand day.

9.13 Scope for Further GA Optimisation Expansion Planning

This GA study of the FCLWD year 2015 design was limited in its scope to optimising new and duplicate pipes, and PRV settings. The GA technique could be used for the optimal design of other decision variables, such as transmission pipelines, pumping station facilities, storage tank locations and capacities, or to decide upon the most economic future sources of supply.

The GA technique is well-suited to optimising design decisions such as those which may be encountered by the Fort Collins - Loveland Water District towards 2015. The decisions for the future sources of supply and the designs of the transmission system and distribution system are closely dependent. The GA has the potential to optimise such aspects of the design simultaneously to create a compatible system design and take advantage of all the available cost savings (both capital costs and pump operating costs). The GA model may be integrated with a specific evaluation scheme to consider the District's particular design decisions.

In summary, the following aspects of the FCLWD design could be optimised with a more comprehensive GA analysis:

- The trade-offs between producing water at the water treatment plant and purchasing (or exchanging) water with adjacent systems, including the corresponding transmission system expansions which would be necessary.
- The upgrades of source pump stations and booster pump stations and approximate operating schedules for the existing and upgraded pumping stations.
- The locations and sizes of new storage tanks and desired operating water levels for new and existing tanks.
- The locations and settings of flow regulating devices such as pressure reducing valves and flow control valves.
- The sizing and layout of new and duplicate pipes for transmission or distribution.

The GA optimisation model would incorporate a simulation model to analyse the design for the demands associated with a particular year (e.g., year 2000 and/or 2015) subject to:

- the projected instantaneous peak hour demands
- the projected maximum day demand pattern (EPS to ensure tanks refill)

9 GA optimisation of the water system expansion plan for the FCLWD

- the projected average day demand pattern (to determine average annual pumping costs)
- and perhaps other demand conditions such as fire demands.

The GA optimisation model would evaluate proposed designs in terms of system performance requirements specified by the District such as:

- minimum and maximum pressures at the demand nodes
- maximum velocities in pipes
- maximum flows from adjacent systems
- maximum inflow from the water treatment plant
- acceptable tank water levels
- refilling of tanks following a peak demand day such that final tank water levels are at least equal to start tank water levels
- pump station flows within the operational limits of the pumping facilities
- PRV settings which maintain the pressure zone arrangement

The GA optimisation model would estimate pipe and equipment costs, installation and construction costs and also system operating costs associated with the GA designs including:

- pipe material, supply and laying costs for new and duplicate pipes
- pump equipment costs and construction costs for new or upgraded pump stations
- construction costs (and land acquisition costs) for new storage tanks (a function of volume and tank site)
- the costs of producing and supplying water from the water treatment plant
- the costs of purchasing water from the adjacent City systems
- and the power costs for operating the pumping stations

Finally, the GA optimisation model could be set up to consider progressive system expansions, for example from 2000 to 2015 by optimising the additions at each time step. A comprehensive GA optimisation study could be expected to produce a range of lowest cost expansion plan designs that satisfy all the design criteria and provide a reliable water supply system. The year 2015 GA design presented in this report is one of a number of low cost designs identified in the GA optimisation study. Other low cost designs exhibited different system features which could potentially be of interest to the District to meet other non-quantifiable objectives they may have in preparing their system expansion plans.

10 Conclusions and Recommendations

A methodology for applying genetic algorithms to pipe network optimisation has been developed and tested in this thesis. The genetic algorithm (GA) model has been based on a traditional GA strategy. A traditional three-operator GA applies proportionate selection, one-point crossover and random bit-wise mutations to an evolving population of fixed-length binary strings and is effective for most applications. In this thesis, various elements of the GA structure are tested in a series of experiments and applications, and through progressive development, an efficient improved GA is formulated to search the pipe network optimisation solution space. The modifications to the traditional GA are intended to exploit the power of the GA.

The GA user has considerable freedom to formulate the GA search for a given problem. The user can take maximum advantage of this flexibility by understanding the problem to be solved and the nature of the solution space to be searched. A summary of the steps for implementing the GA procedure that has been developed in this thesis follows:

- (1) a coding scheme is created, based on some form of coded structure with the potential to represent all of the possible solutions.

To construct an efficient coded structure to represent the set of all possible solutions (the solution space) for a pipe network optimisation problem, the user first identifies the decision variables (e.g., pipes to be sized, PRV settings to be selected, etc.) and the allowable choices for each decision variable (e.g., available pipe sizes, allowable range of pressure settings, etc.). Some creativity may be required to decide how to best represent the unknown decision variables by pieces of genetic code and to decide how the various pieces of code may be best arranged to form a coded structure. Decoding routines are assembled that recognise the trial solutions represented by coded structures (consisting of look-up tables which map pieces of code to decision variable choices).

- (2) a method of evaluation is devised to allocate fitness values to any coded structure solution that may be generated.

A cost function computes the cost of trial solutions (e.g., pipe costs, energy costs for pumping, treatment plant expansion costs, water production costs). A hydraulic simulation model is formulated with the known system data (for the existing system components) and the decision variable choices (for the proposed system expansions), and the hydraulic feasibility of the solution is evaluated for the demand condition(s). The results of the hydraulic simulations are

compared with the expected system performance requirements (e.g., minimum pressures) and infeasible solutions (those that do not satisfy the performance constraints) are penalised. A fitness function is assembled to assign fitness values to coded solutions, in terms of estimated system expansion costs and penalty costs.

- (3) a strategy is developed, including GA operators and search parameters, to generate new fitter populations of coded solutions from the current population using the information provided by the fitness values and the code of the parent solutions.

The traditional three-operator GA has been applied in this thesis to a two-reservoir network expansion problem. An exhaustive enumeration of all possible solutions (about 4 million) to the relatively simple two-reservoir *Gessler* problem was conducted in Chapter 5 of this thesis. The exhaustive enumeration identified the global optimal solutions, feasible and infeasible near-optimal solutions, and characteristics of a typical solution space for the pipe network optimisation problem. The performance of various GA formulations was assessed by comparison with the results of the exhaustive enumeration. Initial results using the traditional GA were encouraging, and an extensive experimental analysis followed to determine how the GA formulation may be improved for this application. A series of experiments were carried out (reported in Chapters 6 and 7) to consider:

- the GA search parameters adopted,
- representation mappings (binary codes, Gray codes and integer codes),
- treatment of the constraints of the optimisation by way of the penalty function method,
- alternative selection methods (tournament selection and proportionate selection),
- an appropriate form of the fitness function and fitness scaling mechanisms,
- various crossover mechanisms (one-point, multi-point and uniform crossover),
- random bit-wise and decision-variable-wise (creeping or adjacency) mutations, and
- the introduction of an elitist concept.

The following elements of the formulation were found to enhance the performance (efficient convergence to lowest cost solutions) of the GA search for pipe network optimisation:

- proportionate selection combined with a power law form of fitness scaling (tournament selection was also very effective),
- both random bit-wise mutations and decision-variable-wise creeping mutations (both with low probability), and
- an elitist concept (in which a small population of the best solutions identified by the GA is maintained in parallel with the working population and members of this *elite* population are occasionally mated with members of the working population).

In addition, it was found that:

- one or two-point crossover provided a sufficient amount of exploration of the search space,
- the use of coded substrings of Gray codes to form the coded string solution was effective (however, the most appropriate coding scheme is partly problem-dependent), and
- the penalty applied to infeasible solutions by the penalty function should be a function of the distance from feasibility (the choice of an appropriate penalty function and penalty factors is problem-dependent and some trial-and-error adjustment of the penalty function(s) is usually required).

Some experimentation with the GA strategy (the combination of GA operators and GA search parameters) is usually required, since the most effective strategy will depend on the size of the solution space (the length of the coded string) and the formulation of the GA model including the choice of coding scheme, fitness evaluation scheme and other GA operators and GA search parameters. A number of GA runs are usually carried out using various traditional and modified operators, to find those best suited to the pipe network optimisation problem and the structure of the coded solution adopted. Additional GA runs are usually carried out using varying GA parameters (such as population size, penalty function coefficients which tighten or relax constraints, and other parameters that drive the GA operations such as the probability of crossover and the probability of mutation). Experimentation is required to find the best performing set of GA parameters and penalty functions.

Consequently, a range of low cost pipe network design solutions are identified by the GA method. The alternative solutions may be quite different network configurations, thus giving decision-makers a choice in network features. The alternative configurations may then be compared in terms of other important (but perhaps non-quantifiable) objectives such as environmental issues, community concerns or possible future developments. Additionally, the penalties may be relaxed such that a number of low cost solutions that are marginally infeasible (just fail to meet the performance constraints) are identified. The engineer can weigh up the trade-off between constraint violations and the cost savings.

Following the analysis of the *Gessler* problem, the improved performance of the GA model formulated in this thesis was confirmed by the application to larger and more complex pipe network optimisation problems. In Chapter 7, large pipe network optimisation problems with known optimal solutions were manufactured by considering the simultaneous optimisation of multiple *Gessler* problems. A modified GA formulation, incorporating coded solutions represented by strings of bits in Gray codes, power law fitness scaling, creeping mutations and the elitist strategy proved to be very effective. The improved GA developed in this research consistently identified global optimal solutions after relatively few solution evaluations

(400,000) compared to the vast solution space for the simultaneous solution of five *Gessler* problems (2^{120} or 1.329×10^{36} possible solutions for a string length of 120 bits). The global optimal solution is not usually known for problems of this size.

The GA was also applied to the benchmark New York City tunnels network expansion problem in Chapter 8. The global optimal solution to the New York tunnels problem is not known, however, the best regions of the solution space have been previously identified by the models of several prominent pipe network optimisation researchers. A series of GA runs were performed for the improved GA, the traditional GA and various intermediate GA formulations to quantify the significance of each proposed feature of the improved GA formulation. The performance of the improved GA was far superior to the traditional GA, and fitness scaling and the elitist strategy were the most effective elements of the improved GA model. The improved GA generated three tunnel networks with a lower cost than the previously lowest cost discrete pipe solution determined by Morgan and Goulter (1985).

A primary objective for researchers of pipe network optimisation is to make the progression from theoretical pipe network optimisation to practical water distribution system design. The optimisation of the year 2015 system expansion plan for the large-scale water transmission and distribution system managed by the Fort Collins - Loveland Water District, Colorado, was used to test the capabilities of the GA model in a realistic design situation in Chapter 9. The Fort Collins - Loveland system supplies water for agricultural and municipal use to an area of about 60 square miles between the cities of Fort Collins and Loveland. The system is composed of about 320 pipes (of which 46 are proposed new or duplicate pipes) and 13 major pressure reducing valves (PRVs) isolating the system into 5 pressure zones. The methodology for optimising pipe network designs using GAs developed in this research was applied to this problem. The GA model was used to optimise the diameters of new and duplicate pipes and the pressure settings of PRVs. The GA design achieved significant cost savings (about 49%) by comparison to a Master Plan design prepared by a Fort Collins engineering consultant using a hydraulic simulation model and standard design guidelines. The design determined by the GA applied to a problem involving many of the real concerns of designers, is evidence that the GA model has the potential to become a valuable, practical design tool.

The performance exhibited by certain elements of the GA strategy is summarised below.

Coded string representation

In this thesis, fixed-length coded strings were formed by concatenated substrings corresponding to decision variables of the pipe network optimisation, and the substrings were coded using binary codes, Gray codes or integers. Binary codes have performed effectively in this thesis, however, experiments have indicated that Gray codes may be a more appropriate coding scheme for this problem. The use of Gray codes ensures that similar coded strings represent similar network solutions. Gray codes eliminate discontinuities, such as the Hamming Cliff of binary codes, that may obstruct the GA as it accumulates highly-fit string similarities within a population of coded strings. Fortunately, the GA seems to be robust enough to overcome biases introduced by the coding scheme.

The performance of the integer coding scheme was shown to be inferior for the relatively small *Gessler* problem in Chapter 6, however, the integer coding scheme was effective for the more complicated Fort Collins - Loveland problem in Chapter 9. The use of an integer coding scheme for the *Gessler* problem, with only eight decision variables and hence eight substrings forming the coded string, does not provide enough genetic information for the GA to operate with. The use of integer values may be more effective where the possible choices for these decision variables are not a power of 2, since redundant binary codes can increase the size of the solution space significantly. In addition, strings of integer numbers are more manageable for problems with many decision variables and many choices for decision variables. The coding scheme adopted influences the choice of GA operators used to manipulate parent coded strings to produce child strings.

Penalty function method

The penalty applied to infeasible solutions by the penalty function should be some function of the distance from feasibility (that is, violations of the constraints should be penalised according to the degree of the violation). This allows the search to approach the optimum solutions from both the feasible and infeasible regions of the solution space. For the GA search applied to the *Gessler* problem, a penalty factor of \$50,000 per psi pressure deficit at the critical low pressure node maintained between 20 and 40 infeasible solutions in each new generation of 100 members and often led to the global optimal solution for \$1.7503 million. It was shown that one parent of the optimal solution may be infeasible.

For the New York tunnels problem, a systematic approach determined the most suitable penalty factor as \$30 million per foot of hydraulic head deficit at the critical node. This penalty factor

produced a feasible lowest cost solution and provided a good balance of feasible and infeasible designs in the search. During the analysis of different penalty factors for the New York tunnels problem, a useful infeasible solution was produced using a relatively low penalty factor of \$5 million per foot of hydraulic head deficit for \$33.63 million. The infeasible design represents a cost saving of \$5.2 million for a relatively small deficit in minimum allowable hydraulic head (approximately 1 ft deficit at three downstream nodes).

Parent selection methods and fitness scaling

Proportionate selection chooses strings according to their fitness with respect to the fitness of fellow strings in the population and chance factors. A fitness scaling mechanism was introduced in this research in the form of a variable exponent in the fitness function to be used in conjunction with proportionate selection and the combination was found to be very effective. The magnitude of the exponent was allowed to increase in steps as the GA run progressed to stretch the range of fitness values. Initially, a low value of the exponent is used, to allow the GA time to sort through the potential strengths of the ordinary strings of the early generations. In time, an increased value of the exponent is used to magnify the small differences in fitness values.

Tournament selection has an advantage in that a fitness function is not required to transform the pipe network cost minimisation problem into a coded string fitness maximisation problem. The coded string solution with the lowest cost wins the tournament. Binary tournament selection was very effective and very efficient for suitable combinations of GA parameters (relatively large population sizes and high crossover and mutation rates are recommended). There may be some value in allowing the weaker competitor a small chance of winning the tournament. Ternary tournament selection can be used to determine reasonably good results very quickly.

Crossover mechanisms

The traditional one-point crossover operator was generally used throughout this thesis. Alternative crossover mechanisms including two-point crossover, multi-point and uniform crossover were investigated for the application to the *Gessler* problem in Chapter 6. Crossovers at any bit position linkage in the 24-bit string and crossovers only at the boundaries of the eight 3-bit decision variable substrings were considered for each crossover mechanism. The less disruptive crossover operators (one-point and two-point) were best suited for crossover points at any bit linkage and the more disruptive crossover operators (uniform) were best suited for crossover points at the substring boundaries.

Mutation

The creeping mutation operator was found to be useful for exploring the solution space in the immediate vicinity of the current population. Low rates of creeping mutations with an increased chance of creeping down were found to be most appropriate. Low rates of random bit-wise mutations were necessary to provide occasional useful variations.

The elitist strategy

The power of the elitist strategy was exceptional for the application to the simultaneous optimisation of multiple *Gessler* problems. The maintenance of the elite population of the previous best solutions is designed to keep the GA search on the right track, while allowing for relentless exploitation of past results by selection and extensive exploration of new parts of the solution space by crossover and mutation. The elitist strategy was less effective for the application to the New York tunnels problem.

10.1 GAs for Pipe Network Optimisation

The GA is powerful in its application, and yet flexible enough in its formulation to overcome many of the obstacles which have frustrated traditional optimisation approaches in their application to the pipe network optimisation problem.

In this thesis, the GA was formulated to select pipe sizes for new pipes and pressure settings for PRVs. The GA can potentially consider any combination of design or operational decision variables for any system component, if the feasibility of the decision can be evaluated by hydraulic simulation. The GA can readily consider discrete choices for decision variables such as a list of available pipe sizes for new pipes, tank locations, the on/off status of pumps or the open/closed status of valves. The use of discrete choices provides an advantage over optimisation techniques which determine continuous solutions.

Pipe network designs are usually subjected to multiple demand conditions such as steady-state simulations of projected peak hour demands, maximum day demands or fire fighting flows superimposed on maximum day demands and extended period simulations (EPS) of projected maximum day demands and average day demands. Naturally, there are uncertainties associated with future demands, long-term water use trends, projected demand variations over the day, the magnitude of fire fighting flows and where they will be critical. There are also uncertainties associated with quality of service and system reliability requirements. These issues need to be resolved to design a system, regardless of whether the system will be optimised or not. Once

they are resolved, optimisation can be used to determine the best design for the specified demands, performance requirements and reliability constraints.

The most important performance constraints are minimum allowable pressures at the demand points for all demand conditions. Other performance constraints may include maximum allowable pressures for demand nodes and maximum allowable velocities for pipes. Pumps are required to operate within acceptable operating limits. Regulating valves supplying the system from other zones or systems usually have a maximum capacity. A minimum hydraulic grade may have to be achieved at connections to other zones or systems. Storage tanks have minimum and maximum operating water levels. Fluctuating water levels in tanks monitored during an EPS are usually required to demonstrate a specified amount of exercising and water levels at the end of the demand cycle are required to reset to initial water levels (in preparation for the next demand cycle). In the evaluation of trial solutions generated by the GA, performance constraints such as these are checked by examination of the results of the hydraulic analysis. The GA method considers multiple demand conditions and multiple sets of constraints by calling the simulation model for each demand pattern.

10.2 The Hydraulic Simulation Model

The GA method can optimise any system that can be analysed by the hydraulic simulation model including branched irrigation networks, extensively looped urban networks, gravity-fed systems or pumped systems. The application of the GA approach to the Fort Collins - Loveland system expansion plan in Chapter 9 of this thesis demonstrated the capability of the GA to handle large-scale, complex systems of multiple pressure zones. Expansions to an existing system, the rehabilitation of an existing system or the construction of a new system may be considered. The GA method can potentially optimise other systems such as gas networks or telecommunications networks if the GA model can be coupled with the appropriate network analysis model.

In Chapter 2 of this thesis, a hydraulic simulation model was developed specifically for the purpose of a fast, integrated link to the GA optimisation model. The hydraulic simulation model uses the Newton-Raphson numerical solution technique applied to the loop corrective flow equations. Algorithms were developed for the hydraulic solver in this thesis to define the loop structure (the path of pipes in natural loops and pseudo loops), to compute a set of initial flows in the network such that continuity is satisfied, and to determine a loop numbering scheme that produces a Jacobian matrix of coefficients of near minimum bandwidth. Sparse matrix routines are used to efficiently reduce identical sparse Jacobian matrices at each iteration of the Newton-Raphson technique. Row scaling and a strategy for selecting a suitable sequence of pivot elements are used to maintain numerical stability and minimise the number of row

reduction operations by reducing the amount of fill-in. A convergence test defines the accuracy of the hydraulic solutions required and avoids unnecessary iterations of the technique.

An alternative to developing a hydraulic simulation model is to embed the code of an established simulation model within the code of the genetic algorithm routines. This approach was followed for the application of the GA to the Fort Collins - Loveland system in Chapter 9. Reliable models for hydraulic simulation are now available which are better equipped to handle such complexities as multiple flow and pressure regulating devices.

10.3 Other Possible Applications of GAs to Pipe Networks

There are many potential applications for GAs to problems concerning the design and operation of water distribution systems. This thesis has concentrated on using the GA to determine the optimal system expansion plan for a specified set of demand conditions such as those for ultimate buildout, such that system expansion costs are a minimum.

The optimal staging (or phasing) of future expansions could potentially be considered by first using the GA to identify the ultimate system expansion plan, and then formulating a second GA with the year of construction of facilities as the decision variables. The proposed system improvements would be subject to incrementally increasing demand conditions.

GAs could also be used effectively for the ongoing optimisation of water supply system operation. The pump operation schedule is the set of rules indicating when the individual pumps should be switched on and off over a specified period of time (usually the demand period cycle) such as a day for an urban system. The optimal pump operation schedule is the policy which minimises total operating cost given expected system demands, desired tank water levels and a set of electricity tariffs. Suppose a system operator can efficiently forecast system demands for the next day based on current demands or historical records, the weather forecast, or more specific information such as scheduled watering days or water orders (for irrigation schemes). The operator could potentially run the GA to determine optimum system operation each day.

10.4 Future GA Model Development

Future development of the GA model is expected to proceed through an ongoing cycle of development and testing. The GA model should be applied to a series of water system expansion problems to better understand the practical problems, including the design options and constraints, performance requirements and operational limitations encountered in practice, the various forms of system data that are available and the size and complexity of existing

system models. An appreciation of the practical problems is necessary to determine how the GA model may be applied to solve them. This is the first step towards the ultimate objective of this research: the development of a generic GA pipe network optimisation model capable of anticipating a wide range of systems and possible design conditions. Less common design considerations might include environmental concerns such as the aesthetics of tank heights, inaccessible (overgrown) pipe routes, soil conditions, boring under freeways, various water quality concerns, political issues associated with abandoning existing facilities, etc. A generic GA model (coupled with a flexible, fully developed hydraulic simulation model) would be a useful design tool for water system engineers.

The computational time to perform a GA run depends on the time required to perform a hydraulic simulation of proposed system designs. The GA processes themselves demand negligible computational effort compared to the simulations of network solutions. Solution evaluations are more time consuming if multiple instantaneous and time-dependent simulations are considered for large water distribution systems. Future work should be undertaken to improve the efficiency of the simulation model, or alternatively special techniques should be investigated to reduce the amount of computer time required to perform an individual solution evaluation. Possible time saving techniques might include:

- identify the most critical demand patterns, perform simulations from most critical to least critical demand pattern and stop the evaluation if the solution is shown to be infeasible (for example, tanks drain during an EPS)
- use the balanced pipe flows of the previous simulation as initial flow assumptions for the new simulation to reduce the number of iterations to convergence (this practice may be of value in the GA search as the population of solutions converges and the pipe network designs become similar)
- consider 12 two-hour demand periods or even 6 four-hour demand periods for a 24-hour EPS (final designs can be tested more accurately over 24 one-hour demand periods)
- perform approximate solution evaluations in the early stages of the GA evolution (for example, EPS analyses might only be performed for one in every five new generations, or less iterations might be performed to convergence of the hydraulic solution)
- with some knowledge of the best regions of the solution space (gained from previous GA runs), it may be desirable to bias the GA starting populations towards specific regions of the search space (in contrast to a randomly generated starting population)

For each new water system expansion problem, it may be necessary to adjust and refine the GA formulation and new GA strategies may be investigated to extract the best performance from the model. It has been found that the choice of coding scheme and fitness evaluation functions are

quite problem-dependent and the most appropriate GA operators and parameter values are in turn, dependent on the choice of coding scheme and fitness evaluation functions.

Genetic algorithm technology is advancing at a great rate and there is seemingly no end to the alternative GA formulations that may be investigated. Of particular interest in this area, is the notion of an adaptive genetic algorithm formulation that evolves as the solution evolves. In this case, many of the decisions made by the GA user may be incorporated into the coded string solution, such that the selection, crossover and mutation mechanisms, GA parameters and even the coding scheme itself evolves as the population evolves. This research has shown that there may be value in varying parameters such as the exponent in the fitness scaling mechanism, penalty factors and the probability of creeping mutation as the GA run proceeds, however, the best way to vary these parameters is uncertain. These uncertainties may be best resolved by the GA.

Finally, further research should be undertaken to develop specific elements of the improved GA proposed in this thesis:

- It became evident from the experiments with alternative coding schemes in Chapter 6, that the performance of one arrangement of decision variable substrings within the string was superior to another. Although further investigation of this issue is necessary, it is likely that decision variable substrings which are related in the hydraulics and economics of the solution should be positioned nearby in the string.
- Varying the penalty factor as the GA run proceeds may be of some value, although further tests are required.
- Further research is necessary to determine the best variation of the fitness scaling exponent with time (perhaps in terms of a relationship between current generation fitness and overall search fitness statistics).
- Although initial results using the elitist strategy developed in this thesis are quite promising, further research is necessary to determine the most appropriate values for parameters such as elite population size and probability of an elite mate, and to investigate issues such as the method of selection of elite mates and the method of updating the elite population.

11 References

- Alperovits, A., and Shamir, U. (1977). "Design of Optimal Water Distribution Systems." *Water Resources Research*, 13(6), 885-900
- Barnard, D.T., and Skillicorn, D.B. (1988). *Pascal for Engineers*. Allyn and Bacon, Inc.
- Beasley, D., Bull, D.R., and Martin, R.R. (1993a). "An Overview of Genetic Algorithms: Part 1, Fundamentals." *University Computing*, 15(2), 58-69
- Beasley, D., Bull, D.R., and Martin, R.R. (1993b). "An Overview of Genetic Algorithms: Part 2, Research Topics." *University Computing*, 15(4), 170-181
- Bethke, A.D. (1981). *Genetic Algorithms as Function Optimizers*. Doctoral Dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, Mich., 129pp
- Bhave, P.R. (1985). "Optimal Expansion of Water Distribution Systems." *J. Environmental Engineering*, ASCE, 111(2), 177-197
- Brindle, A. (1981). *Genetic Algorithms for Function Optimization*. Unpublished doctoral dissertation, University of Alberta, Edmonton
- Brooke, A., Kendrick, D., and Meeraus, A. (1988). *GAMS: A User's Guide*, The Scientific Press, Redwood City, Calif., USA
- Calhoun, C.A. (1971). "Optimisation of Pipe Networks by Linear Programming." *Proc., Control of Flow in Closed Conduits*, Colorado State University, Tullis, J.P., Fort Collins, Colorado, 175-192
- Caruana, R.A., and Schaffer, J.D. (1988). "Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms." *Proc., Fifth International Conference on Machine Learning*, University of Michigan, Ann Arbor, Mich., 153-161
- Cembrowicz, R.G., and Krauter, G.E. (1977). "Optimisation of Urban and Regional Water Supply Systems." *Systems Approach for Development, Proc. of the IFAC Conference*, Cairo, Arab Republic of Egypt, 449-454

11 References

- Coombs, S., and Davis, L. (1987). "Genetic Algorithms and Communication Link Speed Design: Constraints and Operators." *Genetic Algorithms and their Applications: Proc., Second International Conference on Genetic Algorithms*, 257-260
- Cross, H. (1936). "Analysis of Flow in Networks of Conduits or Conductors." *Bulletin No. 286*, University of Illinois Engineering Experimental Station, Urbana, Illinois
- Dandy, G.C., Simpson, A.R., and Murphy, L.J. (1996a). "An Improved Genetic Algorithm for Pipe Network Optimisation." *Water Resources Research*, 32(2), February, 449-458
- Davidson, J.W. and Goulter, I.C. (1992a). "Genetic Algorithm for the Design of Rectilinear Branched distribution Systems, Part I: Data Representation and Evaluation Scheme." Dept. Civil Engineering and Building, University of Central Queensland, Australia
- Davidson, J.W. and Goulter, I.C. (1992b). "Genetic Algorithm for the Design of Rectilinear Branched distribution Systems, Part II: Optimisation." Dept. Civil Engineering and Building, University of Central Queensland, Australia
- Davis, L., and Coombs, S. (1987). "Genetic Algorithms and Communication Link Speed Design: Theoretical Considerations." *Genetic Algorithms and their Applications: Proc., Second International Conference on Genetic Algorithms*, 252-256
- Davis, L. (1989). "Adapting Operator Probabilities in Genetic Algorithms." *Proc., Third International Conference on Genetic Algorithms*, edited by J.D. Schaffer, Morgan Kaufmann, 61-69
- Deb, A.K., and Sarkar, A.K. (1971). "Optimization in Design of Hydraulic Network." *J. Sanitary Engineering Division, ASCE*, 97(SA2), 141-159
- DeJong, K.A. (1975). *Analysis of the Behaviour of a Class of Genetic Adaptive Systems*, dissertation presented to the University of Michigan, at Ann Arbor, Mich., in partial fulfillment of the requirements for the degree of Doctor of Philosophy.
- DeJong, K.A. (1985). "Genetic Algorithms: A 10 Year Perspective." *Proc., First International Conference on Genetic Algorithms*, edited by J.J. Grefenstette, Lawrence Erlbaum Associates, 169-177

11 References

- DeJong, K.A., and Spears, W.M. (1990). "An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms." *Parallel Problem Solving from Nature*, edited by H.P. Schwefel and R. Manner, Springer-Verlag, 38-47
- de Neufville, R., Schaake, J., and Stafford, J. (1971). "Systems Analysis of Water Distribution Networks," *J. Sanitary Engineering Division*, ASCE, 97(SA6), Proc., Paper 8575, Dec., 825-842
- Duan, N., Mays, L.W., and Lansey, K.E. (1990). "Optimal Reliability-Based Design of Pumping and Distribution Systems." *J. Hydraulic Engineering*, ASCE, 116(2), February, 249-268
- El-Bahrawy, A.H., and Smith, A.A. (1987). "A Methodology for Optimal Design of Pipe Distribution Networks." *Canadian Journal of Civil Engineering*, Vol.14, 207-215
- Ellis, D.J., and Simpson, A.R. (1996). *Convergence of Iterative Solvers for the Simulation of a Water Distribution Pipe Network*. Research Report No. R138, Dept. Civil and Environmental Engineering, University of Adelaide, August
- Epp, R., and Fowler, A.G. (1970). "Efficient Code for Steady-State Flows in Networks." *J Hydraulics Division*, ASCE, 96(HY1), 43-56
- Eshelman, L.J., Caruana, R.A., and Schaffer, J.D. (1989). "Biases in the Crossover Landscape." *Proc., Third International Conference on Genetic Algorithms*, edited by J.D. Schaffer, Morgan Kaufmann, 10-19
- Even, S. (1979). *Graph Algorithms*. Technion Institute, Computer Science Press, Inc.
- Featherstone, R.E., and El-Jumaily, K.K. (1983). "Optimal Diameter Selection for Pipe Networks." *J. Hydraulic Engineering*, ASCE, 109(2), 221-234
- Fort Collins Engineering Consultant*. (1993). *Water system Master Plan update for Fort Collins - Loveland Water District*, Fort Collins, Colorado
- Fowler, A.J. (1990). *Water*. Municipal Hydraulics, Vancouver, British Columbia
- Fujiwara, O., and Khang, D.B. (1990). "A Two-Phase Decomposition Method for Optimal Design of Looped Water Distribution Networks." *Water Resources Research*, 26(4), 539-549

11 References

- Gessler, J. (1982). "Optimisation of Pipe Networks." *Proc., International Symposium on Urban Hydrology, Hydraulics and Sediment Control*, University of Kentucky, Lexington, KY., 165-171
- Gessler, J. (1985). "Pipe Network Optimisation by Enumeration." *Proc., Computer Applications for Water Resources.*, ASCE, H.C. Torno (editor), Buffalo, N.Y., 572-581
- Gillies, A.M. (1985). *Machine Learning Procedures for Generating Image Domain Feature Detectors*. Unpublished Doctoral Dissertation, University of Michigan, Ann Arbor, reported by Goldberg (1989)
- Goldberg, D.E. (1985). "Optimal Initial Population Size for Binary-coded Genetic Algorithms." *The Clearinghouse for Genetic Algorithms*, Report No. 85001, Tuscaloosa, University of Alabama, 14pp
- Goldberg, D.E., and Kuo, C.H. (1987). "Genetic Algorithms in Pipeline Optimisation." *J. Computing in Civil Engineering*, ASCE, 1(2), 128-141
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc., 412pp
- Goldberg, D.E. (1989b). "Sizing Populations for Serial and Parallel Genetic Algorithms." *Proc., Third International Conference on Genetic Algorithms*, edited by J.D. Schaffer, Morgan Kaufmann, 70-79
- Goldberg, D.E. (1990). "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking." *IlliGAL*, Report No. 90001, Department of General Engineering, University of Illinois at Urbana-Champaign, 18pp
- Goldberg, D.E., and Deb, K. (1991). "A Comparative Analysis of Selection Schemes used in Genetic Algorithms." *Foundations of Genetic Algorithms*, edited by G.J.E. Rawlins, Morgan Kaufmann, 69-93
- Goldberg, D.E., Deb, K., and Clark, J.H. (1992). "Genetic Algorithms, Noise and Sizing of Populations." *Complex Systems*, 6, 333-362
- Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

11 References

Hollstien, R.B. (1971). *Artificial Genetic Adaptation in Computer Control Systems*. (Doctoral Dissertation, University of Michigan). *Dissertation Abstracts International*, 32(3), 1510B

Jacoby, S.L. (1968). "Design of Optimal Hydraulic Networks." *J. Hydraulics Division*, ASCE, 94(HY3), 641-661

Jeppson, R.W. (1976). *Analysis of Flow in Pipe Networks*. Ann Arbor Science Publishers Inc., Ann Arbor, Michigan

Jeppson, R.W. (1985). "Practical Optimization of Looped Water Systems." *Computer Applications in Water Resources*, edited by H.C. Torno, published by ASCE, New York, 723-731

Jowitt, P.W., and Germanopoulos, G. (1992). "Optimal Pump Scheduling in Water-Supply Networks." *J. Water Resources Planning and Management*, ASCE, 118(4), 406-422

Kally, E. (1972). "Computerized Planning of the Least Cost Water Distribution Network." *Water Sewage Works*, R122-127

Kessler, A. (1988). *Optimal Design of Water Distribution Networks using Graph Theory Techniques* (in Hebrew). Doctoral Thesis in Civil Engineering, Technion, Israel Institute of Technology, Haifa, 142pp

Kessler, A., and Shamir, W. (1991). "Decomposition Technique for Optimal Design of Water Supply Networks." *Engineering Optimization*, 17, 1-19

Lansey, K.E., and Mays, L.W. (1989a). "Optimization Model for Water Distribution System Design." *J. Hydraulic Engineering*, ASCE, 115(10), 1401-1419

Lansey, K.E., and Mays, L.W. (1989b). "Optimization Models for Design of Water Distribution Systems." *Reliability Analysis of Water Distribution Systems*, Task Committee on Risk and Reliability Analysis of Water Distribution Systems, edited by L.W. Mays, ASCE, New York

Lansey, K.E., and Mays, L.W. (1989c). "Network Simulation Models." *Reliability Analysis of Water Distribution Systems*, Task Committee on Risk and Reliability Analysis of Water Distribution Systems, edited by L.W. Mays, ASCE, New York

11 References

- Lasdon, L.S. and Waren, A. (1983). *GRG2 User's Guide*. Dept. General Business, University of Texas, Austin, Texas
- Liebman, J.S., Lasdon, L., Schrage, L., and Waren, A. (1986). *Modelling and Optimization with GINO*, The Scientific Press, Palo Alto, Calif., USA
- Loganathan, G.V., Greene, J.J., and Ahn, T.J. (1995). "Design Heuristic for Globally Minimum Cost Water-Distribution Systems." *J. Water Resources Planning and Management*, ASCE, 121(2), 182-192
- Loubser, B.F., and Gessler, J. (1990). "Computer-Aided Optimisation of Water Distribution Networks." *THE CIVIL ENGINEER in South Africa*, October, 413-422
- Mahjoub, Z. (1983). "Contribution a l'etude de l'optimisation des reseaux mailles." These (Docteur d'Etat), l'Inst. Natl. Polytech., de Toulouse, France, 51-142
- Martin, D.W., and Peters, G. (1963). "The Application of Newton's Method to Network Analysis by Digital Computer." *J. Institute of Water Engineers*, Vol. 17, 115-129
- Martin, Q.W. (1980). "Optimal Design of Water Conveyance Systems." *J. Hydraulics Division*, ASCE, 106(HY9), 1415-1433
- Monbaliu, J., Jo, J.H., Fraisse, C.W. and Vadas, R.G. (1990). "Computer Aided Design of Pipe Networks" in *Water Resource Systems Application*, Simonovic, S.P., Goulter, I.C., Burn, D.H., and Lence, B.J. (Editors), Friesen Printers, Winnipeg, Canada
- Morgan, D.R., and Goulter, I.C. (1985). "Optimal Urban Water Distribution Design." *Water Resources Research*, 21(5), 642-652
- Murphy, L.J., Simpson, A.R., and Dandy, G.C. (1993a). "Design of a Pipe Network Using Genetic Algorithms." *Water*, August, 40-42.
- Murtagh, B.A., and Saunders, M.A. (1980). *MINOS/AUGMENTED User's Manual*. Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA.
- Nielsen, H.B. (1989). "Methods for Analysing Pipe Networks." *J. Hydraulic Engineering*, ASCE, 115(2), 139-157

11 References

- Olde, M. (1985). *Watsys User Manual*, Hydraulic Computer Programming Pty Ltd.
- Ormsbee, L.E., Walski, T.M., Chase, D.V., and Sharp, W.W. (1989). "Methodology for Improving Pump Operation Efficiency." *J. Water Resources Planning and Management*, ASCE, 115(2), 148-164
- Ormsbee, L.E., and Wood, D.J. (1986). "Hydraulic Design Algorithms for Pipe Networks." *J. Hydraulic Engineering*, ASCE, 112(12), 1195-1207
- Parisi, A.V. (1981). *Numerical Methods in Sparse Matrix Problems*. Mathematics Department Report No. 3, James Cook University of North Queensland, Townsville, Qld
- Perez, R., Martinez, F., and Vela, A. (1993). "Improved Design of Branched Networks by using Pressure-Reducing Valves." *J. Hydraulic Engineering*, ASCE, 119(2), 164-180
- Quindry, G., Brill, E.D., and Liebman, J.C. (1981). "Optimisation of Looped Water Distribution Systems." *J. Environmental Engineering Division*, ASCE, 107(EE4), 665-679
- Raman, V., and Raman, S. (1966). "New Method of Solving Distribution System Networks Based on Equivalent Pipe Lengths," *J. American Water Works Association*, May, 615-627
- Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stuttgart
- Richardson, J.T., Palmer, M.R., Liepins, G., and Hilliard, M. (1989). "Some Guidelines for Genetic Algorithms with Penalty Functions." *Proc., Third International Conference on Genetic Algorithms*, Schaffer, J.D., San Mateo, California, M. Kaufmann Publishers, 191-197
- Rossman, L.A. (1994). *EPANET Users Manual*. United States Environmental Protection Agency, Office of Research and Development, Risk Reduction Engineering Laboratory, Cincinnati, Ohio, January
- Salgado, R., Todini, E., and O'Connell, P.E. (1988). "Extending the Gradient Method to include Pressure Regulating Valves in Pipe Networks." *Computer Applications in Water Supply, Vol. I, Systems Analysis and Simulation*, edited by B. Coulbeck and C.H. Orr, Research Studies Press Ltd.

11 References

- Schaake, J.C., and Lai, D. (1969). "Linear Programming and Dynamic Programming Applications to Water Distribution Network Design." *Report 116*, Hydrodynamics Laboratory, Department of Civil Engineering, MIT, Cambridge, Massachusetts
- Schaffer, J.D., and Morishima, A. (1987). "An Adaptive Crossover Distribution Mechanism for Genetic Algorithms." *Proc., Second International Conference on Genetic Algorithms*, edited by J.J. Grefenstette, Lawrence Erlbaum Associates, 36-40
- Schrage, L. (1981). *User's Manual for Linear, Integer and Quadratic Programming with LINDO*. The Scientific Press, USA
- Shamir, U., and Howard, C.D.D. (1968). "Water Distribution Systems Analysis." *J. Hydraulics Division*, ASCE, 94(HY1), 219-234
- Siedlecki, W. and Sklansky, J. (1989). "Constrained Genetic Optimization via Dynamic Reward-Penalty Balancing and its use in Pattern Recognition." *Proc., Third International Conference on Genetic Algorithms*, edited by J.D. Schaffer, Morgan Kaufmann, 141-150
- Simpson, A.R., Dandy, G.C., and Murphy, L.J. (1994). "Genetic Algorithms Compared to Other Techniques for Pipe Optimisation." *Journal of Water Resources Planning and Management Division*, ASCE, 120(4), July/August, 423-443.
- Streeter, V.L., and Wylie, E.B. (1981). *Fluid Mechanics*. McGraw-Hill Ryerson Ltd, New York, 562pp
- Swamee, P.K., and Khanna, P. (1974). "Equivalent Pipe Methods for Optimizing Water Networks-Facts and Fallacies." *J. Environmental Engineering Division*, ASCE, 100(EE1), 93-99
- Swamee, P.K., and Jain, A.K. (1976). "Explicit Equations for Pipe-Flow Problems." *J. Hydraulics Division*, Proc., ASCE, 657-664, May
- Syswerda, G. (1989). "Uniform Crossover in Genetic Algorithms." *Proc., Third International Conference on Genetic Algorithms*, edited by J.D. Schaffer, Morgan Kaufmann, 2-9
- Tarquin, A.J., and Dowdy, J. (1989). "Optimal Pump Operation in Water Distribution." *J. Hydraulic Engineering*, ASCE, 115(2), 158-168

11 References

Tewarson, R.P. (1973). *Sparse Matrices*. Mathematics in Science and Engineering, Volume 99, Academic Press, New York and London

Todini, E., and Pilati, S. (1987). "A Gradient Method for the Analysis of Pipe Networks." *International Conference on Computer Applications for Water Supply and Distribution*, Leicester Polytechnic, UK, September 8-10

Torn, A. and Zilinskas, A. (1989). *Global Optimisation*. Lecture Notes in Computer Science, Edited by Goos, G. and Hartmanis, J., Springer-Verlag, Berlin

Vigus, C.L. (1989). "After Computer Water Network Analysis the Next Quantum Step is OPTIMIZATION." *WATERCOMP '89 The First Australian Conference on Technical Computing in the Water Industry*, Melbourne, 165-168

Walski, T.M. (1985). "State-of-the-Art Pipe Network Optimization." *Computer Applications in Water Resources*, edited by H.C. Torno, published by ASCE, New York, 559-568

Walski, T.M., Brill, E.D., Gessler, J., Goulter, I.C., Jeppson, R.M., Lansey, K., Han-Lin Lee, Liebman, J.C., Mays, L., Morgan, D.R., and Ormsbee, L. (1987). "Battle of the Network Models: Epilogue." *J. Water Resources Planning and Management*, ASCE, 113(2), 191-203

Walters, G.A., and Cembrowicz, R.G. (1993). "Optimal Design of Water Distribution Networks." *Water Supply Systems: State of the Art and Future Trends*, edited by E. Cabrera and F. Martinez, Computational Mechanics Publications, Southampton, England

Walters, G.A., and Lohbeck, T. (1993). "Optimal Layout of Tree Networks using Genetic Algorithms." *Engineering Optimization* 22, 47-48

Waters, C. (1989). "Pipe Network Optimisation." *WATERCOMP '89 The First Australian Conference on Technical Computing in the Water Industry*, Melbourne, 169-172

Wood, D.J. (1974). *Users Manual - A Computer Program for the Analysis of Pressure and Flow in Pipe Distribution Systems*. Office of Engineering Continuing Education, University of Kentucky, Lexington, Ky. (Revised 1975, 1977 and 1979)

Wood, D.J., and Charles, C.O.A. (1972). "Hydraulic Network Analysis Using Linear Theory." *J. Hydraulics Division*, ASCE, 98(HY7), Proc. Paper 9031, 1157-1170

11 References

Wood, D.J., and Rayes, A.M. (1981). "Reliability of Algorithms for Pipe Network Analysis." *J. Hydraulics Division*, ASCE, 107(HY10), 1145-1161

Wood, D.J., and Funk, J.E. (1993). "Hydraulic Analysis of Water Distribution Systems." *Water Supply Systems: State of the Art and Future Trends*, edited by E. Cabrera and F. Martinez, Computational Mechanics Publications, Southampton, England

Appendix A Fort Collins - Loveland System Expansion Plan

EPANET hydraulic simulation input data and output results for the Master Plan design subject to the 2015 peak hour demands

System constants

Number of tanks and reservoirs = 10

Number of nodes = 253

Number of pipes = 323 (including 277 existing pipes, 13 new and 33 duplicate pipes)

Number of pump stations = 3

Number of valves = 14 (including 13 PRVs and 1 FCV)

Table A1 Tank and reservoir input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Description	Node	Elevation (ft)	Hydraulic Grade (ft)	Net Outflow (+ve) / Net Inflow (-ve) (gpm)
Soldier Canyon Tanks	1	5217.00	5232.00	4557.30
Existing Zone 1 1.0 MG Tank	19	5327.50	5341.50	1932.23
Existing Zone 2 4.0MG Tank	85	5221.00	5239.00	3556.34
New Zone 3 4.0MG Tank	271	5132.00	5150.00	3197.67
Connection to Loveland system	363	5263.00	5263.00	510.23
Connection to Loveland system	371	5272.00	5272.00	1572.43
Connection to Fort Collins system	450	5103.00	5103.00	1931.95
New McCloughan Hill 4.0MG Tank	624	5075.00	5093.00	930.49
Existing elevated Airport Tank	635	5053.00	5187.00	1673.23
Existing Timnath Tank	701	4958.00	4968.00	-21.03

Table A2 Node input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
3	5111.00	0.00	5193.30	35.66
5	5100.00	132.00	5184.31	36.53
7	5134.00	11.00	5175.79	18.11
9	5107.00	0.00	5173.89	28.98
15	5107.00	53.00	5335.39	98.96
17	5105.00	0.00	5333.53	99.02
21	5121.00	13.00	5330.86	90.93
23	5220.00	9.00	5330.86	48.03
25	5065.00	33.00	5318.23	109.72
27	5144.00	183.00	5316.46	74.73
29	5240.00	0.00	5316.46	33.13
33	5130.00	29.00	5309.99	77.99
35	5090.00	0.00	5305.85	93.53
37	5110.00	176.00	5299.66	82.18
39	5140.00	183.00	5297.28	68.15
41	5127.00	110.00	5294.12	72.41
43	5127.00	4.00	5294.12	72.41
45	5120.00	110.00	5284.60	71.32
47	5110.00	0.00	5283.52	75.19
49	5160.00	183.00	5281.78	52.77
51	5170.00	324.00	5272.38	44.36
53	5170.00	0.00	5271.48	43.97
55	5170.00	0.00	5350.48	78.20
57	5210.00	0.00	5347.17	59.44
59	5240.00	50.00	5346.82	46.28
60	5300.00	141.00	5349.07	21.26
61	5165.00	515.00	5258.45	40.49
63	5145.00	1060.00	5291.15	63.33
68 ^a	5145.00	-7400.00	5259.22	49.49
69	5160.00	7.00	5256.05	41.62
71	5125.00	110.00	5270.09	62.87
73	5090.00	285.00	5228.96	60.21
75	5080.00	0.00	5249.62	73.50
76	5080.00	0.00	5245.34	71.64
77	5150.00	12.00	5243.72	40.61
79	5125.00	0.00	5242.54	50.93
81	5130.00	3.00	5238.25	46.90
83	5130.00	13.50	5238.01	46.80
87	5120.00	34.00	5237.78	51.04
201	5110.00	0.00	5227.70	51.00
203	5082.00	42.00	5222.01	60.66
205	5025.00	87.00	5211.03	80.61
207	5025.00	0.00	5210.89	80.55
209	5090.00	0.00	5221.41	56.94
211	5085.00	28.00	5214.60	56.15
213	5050.00	0.00	5210.29	69.45
215	5040.00	140.00	5194.06	66.75
217	5030.00	0.00	5194.50	71.28
219	5030.00	2.00	5194.46	71.26
221	5027.00	143.00	5193.81	72.28

^a Connection to Fort Collins city system at the Taft Hill source pump station

Table A2 cont. Node input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
222	5000.00	0.00	5191.12	82.81
223	5010.00	288.00	5191.59	78.68
225	5015.00	24.00	5190.61	76.09
226	5000.00	0.00	5189.30	82.03
227	5020.00	0.00	5214.19	84.14
229	5050.00	145.00	5213.78	70.96
230	5050.00	0.00	5211.77	70.10
231	5000.00	117.00	5214.03	92.74
233	4960.00	0.00	5219.36	112.38
235	5000.00	352.00	5206.04	89.28
237	5020.00	84.00	5203.42	79.48
239	5010.00	213.00	5202.63	83.47
241	5010.00	0.00	5223.27	92.41
243	5000.00	0.00	5207.33	89.84
245	5050.00	22.00	5195.87	63.20
247	5040.00	555.00	5151.85	48.47
249	5057.00	0.00	5202.88	63.21
251	5064.00	81.00	5216.92	66.26
253	5055.00	7.00	5224.32	73.37
255	5064.00	12.00	5224.08	69.36
257	5064.00	131.00	5224.07	69.36
259	5090.00	0.00	5221.14	56.82
261	5030.00	769.00	5220.58	82.58
263	5010.00	442.00	5218.97	90.55
264	5000.00	0.00	5217.44	94.22
265	5030.00	0.00	5215.67	80.45
266	5030.00	0.00	5215.64	80.44
267	5086.00	47.00	5192.93	46.33
269	5165.00	0.00	5181.24	7.04
270	5135.00	0.00	5150.58	6.75
273	5030.00	482.00	5205.75	76.15
275	5030.00	55.00	5204.64	75.67
277	5025.00	55.00	5202.09	76.73
279	5010.00	83.00	5200.61	82.59
281	5000.00	55.00	5200.52	86.89
283	5050.00	55.00	5203.04	66.31
285	5010.00	55.00	5201.10	82.80
287	5010.00	55.00	5200.94	82.73
289	5015.00	55.00	5202.22	81.12
290	5010.00	0.00	5109.06	42.92
291	5010.00	55.00	5202.85	83.56
293	5140.00	6.00	5139.80	-0.08
295	5060.00	0.00	5130.41	30.51
297	5060.00	0.00	5203.50	62.18
299	5060.00	0.00	5203.50	62.18
301	5050.00	0.00	5203.30	66.43
303	5030.00	0.00	5203.14	75.02
305	5010.00	6.00	5203.05	83.65
307	5030.00	0.00	5203.01	74.97
309	5005.00	98.00	5202.96	85.78
311	5005.00	0.00	5109.30	45.19

Table A2 cont. Node input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
313	5040.00	0.00	5203.17	70.70
315	5050.00	194.00	5201.77	65.76
317	5135.00	423.00	5202.84	29.40
319	5088.00	22.00	5203.40	50.00
321	5060.00	0.00	5201.77	61.43
323	5050.00	0.00	5206.83	67.95
325	5050.00	0.00	5201.07	65.46
327	5020.00	113.00	5199.29	77.69
329	5030.00	8.00	5199.31	73.36
331	5030.00	0.00	5199.47	73.43
332	5000.00	0.00	5199.47	86.43
333	5017.00	26.00	5202.95	80.57
335	5070.00	0.00	5202.97	57.62
337	5072.00	464.00	5202.53	56.56
339	5072.00	13.00	5203.27	56.88
341	5072.00	0.00	5203.31	56.90
343	5005.00	88.00	5214.14	90.62
345	5005.00	93.00	5211.50	89.48
347	5020.00	88.00	5212.41	83.37
349	5020.00	0.00	5199.31	77.70
351	4990.00	4.00	5209.22	94.99
353	4990.00	93.00	5213.62	96.89
355	5005.00	0.00	5213.62	90.39
357	5005.00	4.00	5173.67	73.09
359	4976.00	29.00	5173.68	85.65
361	4980.00	0.00	5173.56	83.87
365	5050.00	360.00	5202.27	65.98
367	5030.00	79.00	5194.87	71.44
369	4985.00	303.00	5191.03	89.27
373	4980.00	7.00	5173.55	83.87
375	4960.00	0.00	5173.55	92.53
401	5025.00	0.00	5128.85	45.00
403	4985.00	2.00	5121.49	59.14
405	4954.00	40.00	5114.34	69.48
406	5000.00	0.00	5099.24	43.00
407	4990.00	0.00	5097.09	46.40
409	4966.00	96.00	5089.42	53.48
411	4925.00	0.00	5076.93	65.83
413	4925.00	7.00	5076.62	65.70
415	4925.00	2.00	5076.61	65.69
417	4845.00	59.00	5068.68	96.92
419	4845.00	0.00	5068.30	96.75
421	5000.00	114.00	5099.24	43.00
423	4950.00	0.00	5098.19	64.21
425	4940.00	24.00	5098.14	68.52
427	4900.00	452.00	5097.13	85.42
429	4970.00	189.00	5098.07	55.49
431	4920.00	143.00	5096.64	76.54
433	4900.00	560.00	5096.54	85.16
435	5000.00	661.00	5098.07	42.50
437	5000.00	361.00	5099.24	43.00

Table A2 cont. Node input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
439	4980.00	257.00	5095.73	50.15
441	4990.00	123.00	5095.45	45.69
443	4920.00	7.00	5089.79	73.57
445	4920.00	0.00	5089.79	73.57
447	4952.00	36.00	5089.80	59.71
449	4934.00	82.00	5090.11	67.64
451	4917.00	1409.00	5090.66	75.25
453	4886.00	44.00	5090.01	88.40
455	4850.00	0.00	5085.83	102.19
457	4920.00	323.00	5097.18	76.77
459	4910.00	413.00	5097.47	81.23
461	4930.00	235.00	5092.53	70.42
463	4930.00	0.00	5092.43	70.38
465	4950.00	99.00	5092.34	61.68
467	4940.00	99.00	5092.35	66.01
468	5000.00	0.00	5218.69	94.76
469	5000.00	287.00	5218.97	94.88
470	5000.00	0.00	5099.24	43.00
471	4980.00	275.00	5098.52	51.36
473	4909.00	1371.00	5098.32	82.03
474	4969.00	0.00	5106.47	59.56
475	4940.00	243.00	5092.90	66.25
477	4960.00	23.00	5088.90	55.85
479	4923.00	620.00	5090.65	72.64
481	4876.00	29.00	5044.55	73.03
483	4850.00	6.00	5023.79	75.31
485	5005.00	82.00	5200.54	84.73
487	4990.00	0.00	5200.52	91.22
489	4956.00	1143.00	5106.45	65.19
491	4912.00	593.00	5099.71	81.33
493	4870.00	0.00	5096.13	97.98
495	4875.00	6.00	5094.34	95.04
497	4931.00	48.00	5099.48	73.00
498	5000.00	0.00	5099.71	43.20
499	4960.00	26.00	5099.71	60.53
501	4940.00	0.00	5099.42	69.08
503	4939.00	0.00	5099.43	69.51
505	4939.00	23.00	5099.43	69.51
507	4940.00	32.00	5099.22	68.99
509	4940.00	7.00	5099.21	68.99
510	4960.00	0.00	5096.86	59.30
511	4980.00	7.00	5102.32	53.00
512	4960.00	379.00	5086.39	54.76
513	4940.00	25.00	5089.02	64.57
514	4940.00	12.00	5088.97	64.55
515	4940.00	0.00	5088.96	64.55
516	4911.00	0.00	5090.13	77.62
517	4960.00	41.00	5085.55	54.40
519	4930.00	672.00	5086.93	68.00
521	4905.00	44.00	5087.24	78.97
522	4910.00	0.00	5087.24	76.80

Table A2 cont. Node input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
523	4910.00	6.00	5080.03	73.68
525	4900.00	132.00	5087.16	81.10
527	4900.00	61.00	5087.09	81.07
529	4930.00	0.00	5078.43	64.32
531	4950.00	24.00	5079.92	56.30
533	4950.00	10.00	5076.16	54.67
535	4950.00	0.00	5076.15	54.66
537	4952.00	24.00	5075.98	53.72
539	4970.00	40.00	5069.77	43.23
540	4914.00	100.00	5063.28	64.68
541	4875.00	325.00	5087.05	91.88
542	4920.00	2.00	5087.01	72.36
543	4900.00	0.00	5087.01	81.03
545	4835.00	315.00	5087.01	109.19
603	4984.00	0.00	5166.21	78.95
609	4975.00	561.00	5164.22	81.99
611	4990.00	23.00	5163.32	75.10
613	5005.00	143.00	5163.24	68.57
615	5015.00	0.00	5163.17	64.20
617	5030.00	71.00	5162.96	57.61
619	5045.00	187.00	5162.84	51.06
621	5035.00	35.00	5163.25	55.57
623	5035.00	335.00	5163.26	55.57
625	5034.00	0.00	5164.78	56.67
627	5020.00	260.00	5165.47	63.03
629	5015.00	459.00	5168.14	66.36
631	5007.00	388.00	5174.52	72.59
633	5020.00	62.00	5185.27	71.61
637	4990.00	0.00	5164.18	75.47
639	4983.00	10.00	5164.14	78.49
641	4960.00	10.00	5164.14	88.45
643	4960.00	10.00	5164.14	88.45
645	4980.00	10.00	5164.14	79.79
703	4860.00	0.00	4968.01	46.80
705	4870.00	268.00	4968.05	42.49
707	4865.00	47.00	4967.98	44.62
709	4845.00	6.00	4983.47	60.00
711	4842.00	21.00	4978.50	59.14
713	4843.00	0.00	4973.75	56.66
715	4843.00	52.00	4968.02	54.17
717	4840.00	0.00	4967.78	55.37
719	4850.00	34.00	4956.71	46.24
721	4850.00	50.00	4979.24	56.00
723	4850.00	41.00	4967.54	50.93
725	4850.00	37.00	4956.70	46.23
727	4840.00	82.00	4967.64	55.31
729	4850.00	30.00	5021.90	74.48
731	4818.00	50.00	5019.99	87.52
733	4800.00	48.00	5035.91	102.22
735	4847.00	0.00	4967.87	52.37

Table A3 Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
1	1	3	20.00	8100	110	4557.30	4.65	4.78
3	3	5	20.00	2940	140	4557.30	4.65	3.06
5	5	7	20.00	2940	140	4425.30	4.52	2.90
7	7	9	20.00	660	140	4414.30	4.51	2.88
17	15	17	20.00	660	140	4361.30	4.45	2.82
19	19	17	16.00	4300	140	1932.23	3.08	1.85
21	17	21	20.00	480	140	6293.53	6.43	5.56
23	21	23	8.00	2640	140	9.00	0.06	0.00
25	21	25	18.00	5280	140	3024.81	3.81	2.39
27	21	27	18.00	5280	140	3246.72	4.09	2.73
29	29	27	12.00	2160	140	0.00	0.00	0.00
35	27	33	18.00	2640	140	3063.72	3.86	2.45
37	25	35	18.00	5280	140	2991.81	3.77	2.35
39	35	37	18.00	2640	140	2991.81	3.77	2.35
41	33	39	18.00	5280	140	3034.72	3.83	2.41
43	37	43	18.00	2640	140	2815.81	3.55	2.10
45	39	41	18.00	2640	140	2079.38	2.62	1.20
47	41	43	12.00	200	140	-43.10	0.12	0.00
49	43	45	16.00	2640	140	2768.71	4.42	3.61
51	45	47	16.00	500	140	2097.94	3.35	2.16
55	39	49	8.00	1560	140	772.35	4.93	9.94
57	49	51	8.00	1560	140	589.35	3.76	6.03
59	51	53	8.00	1200	140	191.00	1.22	0.75
61	55	57	4.00	1800	140	50.00	1.28	1.84
62	57	59	6.00	1400	140	50.00	0.57	0.25
63	55	60	8.00	3300	140	141.00	0.90	0.43
67	63	51	8.00	1920	140	765.57	4.89	9.78
69	41	63	18.00	2640	140	2012.48	2.54	1.13
71	45	71	8.00	2640	140	560.77	3.58	5.50
73	63	71	4.00	1000	140	186.91	4.77	21.05
75	51	61	8.00	1200	140	839.91	5.36	11.61
77	69	61	8.00	1200	140	-324.91	2.07	2.00
78 ^a	68	76	18.00	7920	140	2555.91	3.22	1.75
79 ^a	76	87	18.00	5280	140	2290.93	2.89	1.43
82	68	69	18.00	2000	140	2422.05	3.05	1.59
83 ^a	68	69	18.00	2000	140	2422.05	3.05	1.59
85	71	73	8.00	5900	140	637.68	4.07	6.97
86	69	75	18.00	3600	140	2581.00	3.25	1.78
87 ^a	69	75	18.00	3600	140	2581.00	3.25	1.78
88	75	76	18.00	2400	140	2581.00	3.25	1.78
89 ^a	75	76	18.00	2400	140	2581.00	3.25	1.78
90	76	77	18.00	1440	140	2010.31	2.53	1.12
91	77	79	18.00	1060	140	2004.31	2.53	1.12
92 ^b	76	241	12.00	5280	140	1406.37	3.99	4.18
93	79	81	18.00	3840	140	2004.31	2.53	1.12
94 ^a	76	77	18.00	1440	140	2010.31	2.53	1.12
95	81	83	18.00	215	140	2002.81	2.53	1.12

^a Duplicate pipes between Taft Hill source pump station (node 68) and Trilby Road

^b Clarendon Hills new pipe

Table A3 cont. Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
96 ^a	77	79	18.00	1060	140	2004.31	2.53	1.12
97	83	87	18.00	200	140	1996.06	2.52	1.11
98 ^a	79	81	18.00	3840	140	2004.31	2.53	1.12
99	85	87	16.00	2640	140	910.09	1.45	0.46
100 ^c	85	87	24.00	2640	140	2646.25	1.88	0.46
102 ^a	81	83	18.00	215	140	2002.81	2.53	1.12
104 ^a	83	87	18.00	200	140	1996.06	2.52	1.11
201	201	203	16.00	2640	140	2097.53	3.35	2.16
203	203	205	16.00	5280	140	2055.53	3.28	2.08
205	205	207	6.00	100	140	125.29	1.42	1.39
211	209	211	6.00	720	140	352.68	4.00	9.46
213	211	213	6.00	3120	140	124.61	1.41	1.38
215	213	215	6.00	1920	140	331.86	3.77	8.45
217	215	221	10.00	960	140	191.86	0.78	0.25
219	205	217	14.00	5080	140	1843.24	3.84	3.25
221	217	219	14.00	10	140	1843.24	3.84	3.27
223	219	221	14.00	200	140	1841.24	3.84	3.25
225	221	223	12.00	1500	140	803.26	2.28	1.48
226	211	227	10.00	1480	140	200.07	0.82	0.28
227	223	225	12.00	1500	140	515.26	1.46	0.65
228	225	226	10.00	900	140	491.26	2.01	1.45
229	227	229	10.00	1500	140	200.07	0.82	0.28
231	229	230	8.00	2300	140	207.25	1.32	0.87
232	230	213	8.00	1700	140	207.25	1.32	0.87
233	221	222	14.00	2200	140	1086.85	2.27	1.23
235	229	231	10.00	1500	140	-152.18	0.62	0.17
237	233	231	10.00	1500	140	797.27	3.26	3.56
239	231	235	6.00	400	140	528.09	5.99	19.96
241	241	233	10.00	1100	140	797.27	3.26	3.55
243	237	235	6.00	1000	140	-176.09	2.00	2.62
245	239	237	6.00	1000	140	-92.09	1.04	0.79
247	241	243	6.00	1200	140	423.61	4.81	13.28
249	243	239	6.00	3600	140	120.91	1.37	1.31
251	241	245	4.00	1320	140	185.48	4.74	20.76
253	243	247	4.00	1080	140	302.70	7.73	51.37
255	245	247	4.00	1200	140	252.30	6.44	36.68
257	249	245	4.00	1320	140	88.82	2.27	5.32
259	251	249	4.00	2640	140	88.82	2.27	5.32
261	257	251	3.00	100	140	169.82	7.71	71.57
263	257	259	10.00	2800	140	411.84	1.68	1.05
264 ^c	257	259	24.00	2800	140	4126.73	2.93	1.05
265	259	261	8.00	500	140	238.50	1.52	1.13
266 ^c	259	261	24.00	500	140	4300.07	3.05	1.13
267	261	263	8.00	2000	140	198.09	1.26	0.80
268	263	469	8.00	10	140	174.86	1.12	0.63
269	469	264	14.00	1400	140	1019.59	2.13	1.09
270 ^c	261	263	24.00	2000	140	3571.48	2.53	0.80

^a Duplicate pipes between Taft Hill source pump station (node 68) and Trilby Road

^c Duplicate pipes along Trilby between the existing Zone 2 Tank (4.0MG) and College Ave

Table A3 cont. Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
271	87	253	18.00	5280	140	3130.19	3.95	2.55
272 ^c	87	253	24.00	5280	140	6675.19	4.73	2.55
273	253	255	10.00	200	140	440.74	1.80	1.19
274 ^c	253	257	24.00	210	140	4410.65	3.13	1.19
275	255	257	10.00	10	140	428.74	1.75	1.12
276 ^c	263	469	24.00	10	140	3152.71	2.24	0.63
277	253	267	18.00	5280	140	4947.00	6.24	5.95
279	266	273	14.00	3960	140	1597.88	3.33	2.50
281	267	269	18.00	2000	140	4900.00	6.18	5.84
284	270	271	18.00	100	140	4900.00	6.18	5.84
285	271	293	18.00	2640	140	3917.85	4.94	3.86
287	293	295	18.00	2440	140	3911.85	4.93	3.85
291	299	297	18.00	200	140	0.00	0.00	0.00
292	275	299	14.00	1710	140	781.53	1.63	0.67
293	273	485	3.00	880	140	44.14	2.00	5.92
294	299	301	14.00	450	140	621.74	1.30	0.44
295	273	275	14.00	930	140	1071.74	2.23	1.19
296	305	307	14.00	1200	140	140.87	0.29	0.03
297	279	485	6.00	450	140	39.69	0.45	0.17
298	301	303	14.00	800	140	413.76	0.86	0.21
299	279	281	6.00	850	140	31.55	0.36	0.11
300	275	277	6.00	570	140	235.21	2.67	4.47
301	277	285	4.00	1800	140	25.97	0.66	0.55
302	277	279	6.00	720	140	154.24	1.75	2.05
303	285	287	4.00	600	140	18.00	0.46	0.28
305	287	281	4.00	1070	140	21.62	0.55	0.39
307	289	285	4.00	680	140	47.03	1.20	1.64
309	289	287	4.00	520	140	58.62	1.50	2.46
310 ^d	295	311	18.00	5480	140	3911.85	4.93	3.85
311	291	289	8.00	1160	140	160.65	1.03	0.54
313	283	291	8.00	1270	140	79.51	0.51	0.15
314	309	333	12.00	1500	140	42.87	0.12	0.01
315	301	283	8.00	675	140	134.51	0.86	0.39
317	303	291	8.00	720	140	136.13	0.87	0.40
318	311	290	20.00	230	140	2549.35	2.60	1.04
319	303	305	14.00	935	140	277.63	0.58	0.10
321	307	309	14.00	1850	140	140.87	0.29	0.03
325	301	313	8.00	1000	140	73.46	0.47	0.13
327	313	305	8.00	1000	140	73.46	0.47	0.13
329	305	315	8.00	1500	140	204.22	1.30	0.85
331	319	299	14.00	2640	140	-159.79	0.33	0.04
333	315	321	8.00	1500	140	10.22	0.07	0.00
335	319	317	14.00	2640	140	423.00	0.88	0.21
337	319	321	6.00	2000	140	93.91	1.07	0.82
339	323	319	12.00	2640	140	747.07	2.12	1.30
341	321	325	6.00	700	140	104.13	1.18	0.99
343	325	327	6.00	1800	140	104.13	1.18	0.99

^c Duplicate pipes along Trilby between the existing Zone 2 Tank (4.0MG) and College Ave

^d Duplicate pipe south of new Zone 3 Tank (4.0MG) between College and Lemay

Table A3 cont. Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
345	327	329	6.00	1800	140	-8.87	0.10	0.01
347	329	331	6.00	4800	140	-16.87	0.19	0.03
349	333	331	3.00	3480	140	16.87	0.77	1.00
350	331	332	6.00	1340	140	0.00	0.00	0.00
353	319	335	14.00	2640	140	367.95	0.77	0.17
355	347	323	12.00	4300	140	747.07	2.12	1.30
357	329	349	3.00	2640	140	0.00	0.00	0.00
359	335	337	14.00	2640	140	367.95	0.77	0.17
361	339	337	6.00	200	140	211.51	2.40	3.67
363	341	339	6.00	10	140	224.51	2.55	4.10
365	343	341	6.00	2640	140	224.51	2.55	4.10
366	371	343	6.00	385	140	1572.43	17.84	150.29
367	365	337	10.00	2640	140	-115.46	0.47	0.10
368	363	365	4.00	450	140	510.23	13.03	134.96
369	365	367	6.00	1320	140	265.68	3.01	5.60
371	367	369	6.00	1320	140	186.68	2.12	2.92
373	343	369	4.00	2640	140	116.32	2.97	8.76
375	343	347	12.00	1000	140	873.64	2.48	1.73
377	347	345	4.00	800	140	38.57	0.98	1.14
379	343	345	6.00	1000	140	176.97	2.01	2.64
381	343	353	8.00	2640	140	93.00	0.59	0.20
383	345	351	6.00	1700	140	122.54	1.39	1.34
387	353	355	8.00	200	140	0.00	0.00	0.00
391	351	359	4.00	3920	140	118.54	3.03	9.07
393	357	359	8.00	3920	140	-4.00	0.03	0.00
395	359	373	4.00	2640	140	7.00	0.18	0.05
397	375	373	4.00	2640	140	0.00	0.00	0.00
401	401	403	6.00	5280	140	125.29	1.42	1.39
403	403	405	6.00	5280	140	123.29	1.40	1.35
405	405	409	4.00	5280	140	83.29	2.13	4.72
407	406	407	10.00	1480	140	491.36	2.01	1.45
409	407	409	10.00	5280	140	491.36	2.01	1.45
411	409	411	10.00	5280	140	639.73	2.61	2.37
413	411	413	6.00	200	140	132.26	1.50	1.54
414 ^e	411	413	10.00	200	140	507.48	2.07	1.54
415	413	415	6.00	10	140	130.81	1.48	1.46
416 ^e	413	415	10.00	10	140	501.92	2.05	1.46
417	415	417	6.00	5280	140	130.39	1.48	1.50
418 ^e	415	417	10.00	5280	140	500.34	2.04	1.50
419	417	419	10.00	200	140	571.73	2.34	1.92
423	421	423	14.00	1050	140	972.84	2.03	1.00
425	423	425	14.00	500	140	289.13	0.60	0.11
427	423	427	12.00	960	140	683.71	1.94	1.10
431	425	429	14.00	700	140	265.13	0.55	0.09
433	429	427	6.00	1800	140	73.72	0.84	0.52
435	435	429	14.00	2000	140	-2.41	0.01	0.00
437	427	431	12.00	2000	140	305.43	0.87	0.25
439	431	433	14.00	2640	140	162.43	0.34	0.04

^e Duplicate pipes along Harmony Road between County Rd 7 and County Rd 9

Table A3 cont. Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
441	409	443	4.00	5280	140	-8.53	0.22	0.07
442 ^f	409	443	12.00	5280	140	-152.55	0.43	0.07
443	433	439	14.00	1525	140	691.87	1.44	0.53
445	439	441	12.00	600	140	434.87	1.23	0.48
447	441	461	8.00	1570	140	311.87	1.99	1.86
448 ⁹	450	451	12.00	1640	140	1931.95	5.48	7.52
449	443	445	4.00	10	140	-9.16	0.23	0.10
450 ^f	443	445	12.00	10	140	-158.92	0.45	0.10
451	445	447	4.00	200	140	-9.16	0.23	0.07
452 ^f	445	447	12.00	200	140	-158.92	0.45	0.07
453	447	449	4.00	2640	140	-11.22	0.29	0.12
454 ^h	447	449	12.00	2640	140	-202.31	0.57	0.12
455	451	449	4.00	2640	140	15.53	0.40	0.21
456 ^h	451	449	12.00	2640	140	280.00	0.79	0.21
457	451	453	3.00	5280	140	5.44	0.25	0.12
458 ^h	451	453	12.00	5480	140	205.04	0.58	0.12
459	453	455	3.00	200	140	87.30	3.96	20.90
462	469	468	12.00	600	140	423.09	1.20	0.45
463	453	481	3.00	2606	140	79.18	3.59	17.45
464	469	265	14.00	1320	140	1597.88	3.33	2.50
465	437	435	14.00	2400	140	658.59	1.37	0.48
466	470	471	12.00	1580	140	423.07	1.20	0.45
467	457	433	20.00	2940	140	1089.44	1.11	0.22
468	471	473	12.00	3100	140	148.07	0.42	0.06
469	461	463	8.00	700	140	76.87	0.49	0.14
470	265	266	14.00	10	140	1597.88	3.33	2.49
471	463	465	8.00	650	140	76.87	0.49	0.14
472	459	457	20.00	820	140	1412.44	1.44	0.35
473	467	465	8.00	685	140	22.13	0.14	0.01
474	473	459	20.00	1520	140	1825.44	1.86	0.56
475	475	467	8.00	1680	140	121.13	0.77	0.32
477	473	475	12.00	2640	140	958.14	2.72	2.06
478 ⁱ	451	479	12.00	8000	140	16.94	0.05	0.00
479	447	477	3.00	2640	140	9.45	0.43	0.34
480 ^j	475	479	12.00	2640	140	594.01	1.69	0.85
481	479	477	3.00	2640	140	13.55	0.61	0.67
482 ^k	271	474	18.00	10000	140	4179.82	5.27	4.35
483	481	483	3.00	2766	140	50.18	2.28	7.50
484 ^l	489	491	16.00	5280	140	1579.66	2.52	1.28
487	479	491	3.00	5280	140	-22.60	1.03	1.72
488	474	473	20.00	3380	140	4006.51	4.09	2.41
489	485	487	3.00	940	140	1.83	0.08	0.02

^f Duplicate pipes along Timberline between Harmony and County Rd 36

⁹ New connection to Fort Collins city system at County Rd 9

^h Duplicate pipes along along County Rd 36 between Timberline and County Rd 7

ⁱ New pipe from County Rd 9 southwest to Trilby/Timberline

^j New pipe from Trilby/Timberline to Paragon Point

^k New pipe from new Zone 3 Tank (4.0MG) east to Lemay

^l New pipe along County Rd 12 between Lemay and Timberline

Table A3 cont. Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
490	487	281	4.00	540	140	1.83	0.05	0.00
491	491	493	3.00	2640	140	19.91	0.90	1.36
492	474	489	20.00	1900	140	173.31	0.18	0.01
493	493	495	3.00	1320	140	19.91	0.90	1.36
495	491	505	3.00	5280	140	3.48	0.16	0.05
496	290	489	20.00	2500	140	2549.35	2.60	1.04
497	311	497	14.00	5280	140	1362.50	2.84	1.86
499	503	505	3.00	10	140	-1.55	0.07	0.00
501	501	503	3.00	200	140	-1.55	0.07	0.01
502	498	499	6.00	1300	140	0.00	0.00	0.00
503	499	501	6.00	1320	140	45.54	0.52	0.21
504 ^m	491	497	12.00	2640	140	175.11	0.50	0.09
505	501	507	6.00	200	140	106.60	1.21	1.04
506 ^m	505	497	12.00	2430	140	-80.57	0.23	0.02
507	507	509	6.00	10	140	74.60	0.85	0.54
508 ^m	503	505	12.00	10	140	-59.51	0.17	0.00
509	509	510	6.00	5280	140	67.60	0.77	0.45
510 ^m	501	503	12.00	200	140	-59.51	0.17	0.01
511	511	499	6.00	5280	140	71.54	0.81	0.49
515	359	361	6.00	200	140	78.54	0.89	0.59
516 ⁿ	491	493	12.00	2640	140	765.56	2.17	1.36
517	497	513	14.00	5280	140	1409.04	2.94	1.98
518 ⁿ	493	495	12.00	1320	140	765.56	2.17	1.36
519	513	514	14.00	200	140	478.93	1.00	0.27
520	513	512	12.00	2588	140	655.38	1.86	1.02
521	514	515	14.00	10	140	466.93	0.97	0.24
522 ^o	495	516	12.00	3000	140	779.47	2.21	1.40
523	515	519	14.00	7920	140	466.93	0.97	0.26
524 ^p	516	513	12.00	2640	140	405.66	1.15	0.42
525	510	531	4.00	5280	140	67.60	1.73	3.21
526 ^q	513	512	12.00	2588	140	655.38	1.86	1.02
527	531	533	4.00	2640	140	43.60	1.11	1.42
528 ^r	516	521	12.00	8000	140	373.81	1.06	0.36
529	533	535	4.00	10	140	33.60	0.86	0.88
531	535	537	4.00	200	140	33.60	0.86	0.88
533	537	539	3.00	2640	140	26.80	1.22	2.35
535	523	537	3.00	3920	140	17.20	0.78	1.04
537	529	539	3.00	2640	140	32.09	1.46	3.28
539	521	519	12.00	2640	140	205.07	0.58	0.12
540	539	540	3.00	5280	140	18.88	0.86	1.23
541	527	523	3.00	3920	140	23.20	1.05	1.80
542	542	540	4.00	5280	140	81.12	2.07	4.49
543	522	521	30.00	920	140	-124.74	0.06	0.00

- ^m Duplicate pipes along Timberline between County Rd 30 and County Rd 32
- ⁿ Duck Lake duplicate pipes east of Timberline
- ^o Duck Lake new pipe east of Timberline
- ^p Duck Lake new pipe south along County Rd 9
- ^q County Rd 9 duplicate pipes between County Rd 30 1/2 and County Rd 28
- ^r Duck Lake new pipe east to Interstate 25

Table A3 cont. Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
544	522	525	30.00	3000	140	1055.23	0.48	0.03
545	527	529	3.00	2640	140	32.09	1.46	3.28
547	527	733	3.00	2640	140	83.82	3.80	19.39
548	525	527	6.00	20	140	200.11	2.27	3.32
549	525	541	24.00	2640	140	723.12	0.51	0.04
551	541	543	24.00	2640	140	398.12	0.28	0.01
553	543	545	24.00	500	140	315.00	0.22	0.01
555	512	517	12.00	1545	140	465.88	1.32	0.54
557	543	542	12.00	200	140	83.12	0.24	0.02
601 ^q	512	517	12.00	1545	140	465.88	1.32	0.54
609	603	609	12.00	4000	140	445.38	1.26	0.50
610 ^q	603	609	12.00	4000	140	445.38	1.26	0.50
611	609	611	12.00	3978	140	289.77	0.82	0.22
613	611	613	12.00	410	140	266.77	0.76	0.19
615	613	615	12.00	1279	140	133.08	0.38	0.05
617	615	617	12.00	1200	140	258.00	0.73	0.18
619	617	619	12.00	1200	140	187.00	0.53	0.10
620 ^s	624	522	16.00	12000	140	930.49	1.48	0.48
621	613	621	8.00	2655	140	-9.31	0.06	0.00
623	615	623	12.00	1771	140	-124.92	0.35	0.05
625	621	623	12.00	1200	140	-44.31	0.13	0.01
627	623	625	12.00	2425	140	-504.23	1.43	0.63
629	625	627	12.00	1100	140	-504.23	1.43	0.63
631	629	627	12.00	1975	140	764.23	2.17	1.35
633	631	629	12.00	1975	140	1223.23	3.47	3.23
635	633	631	12.00	2000	140	1611.23	4.57	5.38
637	635	633	12.00	300	140	1673.23	4.75	5.76
639	609	637	10.00	2640	140	40.00	0.16	0.01
641	637	639	10.00	2640	140	40.00	0.16	0.01
643	639	641	10.00	1000	140	18.11	0.07	0.00
645	641	643	10.00	1000	140	8.11	0.03	0.00
647	645	643	10.00	1000	140	1.89	0.01	0.00
649	639	645	10.00	2640	140	11.89	0.05	0.00
701	701	703	12.00	3800	140	-21.03	0.06	0.00
703	705	703	12.00	2640	140	70.49	0.20	0.02
705	703	707	12.00	2640	140	49.46	0.14	0.01
707	713	705	8.00	2640	140	338.49	2.16	2.16
709	709	711	10.00	2640	140	565.73	2.31	1.88
711	711	713	10.00	2700	140	544.73	2.23	1.76
713	713	715	3.00	100	140	150.67	6.84	57.36
715	715	717	3.00	3960	140	3.70	0.17	0.06
717	715	719	3.00	3000	140	34.59	1.57	3.77
719	713	707	4.00	2584	140	55.57	1.42	2.23
721	717	723	3.00	3960	140	3.70	0.17	0.06
723	721	723	3.00	2700	140	37.30	1.69	4.33
725	719	725	3.00	3000	140	0.59	0.03	0.00
727	727	725	3.00	2640	140	36.41	1.65	4.15

^q County Rd 9 duplicate pipes between County Rd 30 1/2 and County Rd 28

^s New pipe up to the new McCloughan Hill Tank

Table A3 cont. Pipe input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
729	729	731	3.00	2640	140	14.18	0.64	0.72
731	733	731	3.00	3960	140	35.82	1.63	4.02
733 ^t	715	707	12.00	2640	140	60.38	0.17	0.01
735 ^t	707	735	12.00	2640	140	118.41	0.34	0.04
737 ^t	735	727	12.00	5280	140	118.41	0.34	0.04

^t Loop of new pipes at County Rd 3

Table A4 Pump station (PS) input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Pump station description	Pipe	Start node	End node	No. of pumps operating	Rated flow (gpm)	Rated head (ft)	Flow (gpm)	Power output (hp)	Pump lift (ft)
Overland Trail source PS	9	9	15	3	4252	165.8	4414.30	180	161.50
Westridge booster PS	60	53	55	1	191	79	191.00	4	79.00
Airport booster PS	603	517	603	1 ^a	620	125	890.77	18	80.66

^a ESP-1 operating curve

Table A5 Pressure reducing valve (PRV) and flow control valve (FCV) input data and output results for the EPANET simulation of the Master Plan design subject to the 2015 peak hour demands

Valve	Pipe	Start node	End node	Diameter (inches)	Pressure setting (psi)	Status	Flow (gpm)	Velocity (fps)	Head loss (ft)
PRV 1	53	47	201	8.00	51	Operating	2097.94	13.39	55.82
PRV 3	207	207	401	4.00	45	Operating	125.29	3.20	82.04
PRV 2	209	73	209	4.00	60	Operating	352.68	9.00	7.55
FCV 32	283	269	270	18.00	4900 ^a	Operating	4900.00	6.18	30.66
PRV 29	351	332	498	6.00	43	Closed	0.00	-	-
PRV 23	389	355	357	4.00	43	Closed	0.00	-	-
PRV 28	404	226	406	6.00	43	Operating	491.26	5.57	90.07
PRV 27	420	222	421	10.00	43	Operating	1086.85	4.44	91.88
PRV 8	421	419	709	3.00	60	Operating	571.73	25.95	84.82
PRV 31	460	264	437	10.00	43	Operating	1019.59	4.17	118.20
PRV 9	461	455	721	2.00	56	Operating	87.30	8.92	106.59
PRV 30	476	468	470	12.00	43	Operating	423.09	1.20	119.46
PRV 15	485	483	729	2.00	90	Not operating	44.18	4.51	1.90
PRV 14	513	361	511	3.00	53	Operating	78.54	3.56	71.24

^a Flow setting for the FCV 32

Appendix B Fort Collins - Loveland System Expansion Plan

EPANET hydraulic simulation input data and output results for the Genetic Algorithm design subject to the 2015 peak hour demands

System constants

Number of tanks and reservoirs = 10

Number of nodes = 253

Number of pipes = 299 (including 277 existing pipes, 13 new and 9 duplicate pipes)

Number of pump stations = 3

Number of valves = 14 (including 13 PRVs and 1 FCV)

Table B1 Tank and reservoir input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Description	Node	Elevation (ft)	Hydraulic Grade (ft)	Net Outflow (+ve) / Net Inflow (-ve) (gpm)
Soldier Canyon Tanks	1	5217.00	5232.00	4647.48
Existing Zone 1 1.0 MG Tank	19	5327.50	5341.50	2480.94
Existing Zone 2 4.0MG Tank	85	5221.00	5239.00	1566.26
New Zone 3 4.0MG Tank	271	5132.00	5150.00	2644.07
Connection to Loveland system	363	5263.00	5263.00	963.03
Connection to Loveland system	371	5272.00	5272.00	2433.76
Connection to Fort Collins system	450	5103.00	5103.00	1813.90
New McCloughan Hill 4.0MG Tank	624	5075.00	5093.00	1390.74
Existing elevated Airport Tank	635	5053.00	5187.00	1753.91
Existing Timnath Tank	701	4958.00	4968.00	153.82

Table B2 Node input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
3	5111.00	0.00	5191.87	35.04
5	5100.00	132.00	5182.55	35.77
7	5134.00	11.00	5173.71	17.21
9	5107.00	0.00	5171.73	28.05
15	5107.00	53.00	5330.78	96.96
17	5105.00	0.00	5328.84	96.99
21	5121.00	13.00	5325.65	88.68
23	5220.00	9.00	5325.65	45.78
25	5065.00	33.00	5310.38	106.33
27	5144.00	183.00	5308.58	71.31
29	5240.00	0.00	5308.58	29.71
33	5130.00	29.00	5300.83	74.02
35	5090.00	0.00	5295.39	89.00
37	5110.00	176.00	5287.90	77.08
39	5140.00	183.00	5285.59	63.08
41	5127.00	110.00	5281.24	66.83
43	5127.00	4.00	5281.12	66.78
45	5120.00	110.00	5265.41	63.01
47	5110.00	0.00	5263.17	66.37
49	5160.00	183.00	5272.96	48.94
51	5170.00	324.00	5265.80	41.51
53	5170.00	0.00	5264.90	41.12
55	5170.00	0.00	5343.90	75.35
57	5210.00	0.00	5340.60	56.59
59	5240.00	50.00	5340.24	43.43
60	5300.00	141.00	5342.49	18.41
61	5165.00	515.00	5257.93	40.27
63	5145.00	1060.00	5278.64	57.91
68 ^a	5145.00	-7400.00	5260.55	50.07
69	5160.00	7.00	5257.65	42.31
71	5125.00	110.00	5257.40	57.37
73	5090.00	285.00	5232.66	61.81
75	5080.00	0.00	5252.04	74.54
76	5080.00	0.00	5248.29	72.92
77	5150.00	12.00	5245.59	41.42
79	5125.00	0.00	5243.61	51.39
81	5130.00	3.00	5236.45	46.12
83	5130.00	13.50	5236.05	45.95
87	5120.00	34.00	5235.68	50.12
201	5110.00	0.00	5206.93	42.00
203	5082.00	42.00	5195.14	49.02
205	5025.00	87.00	5172.15	63.76
207	5025.00	0.00	5171.99	63.69
209	5090.00	0.00	5193.85	45.00
211	5085.00	28.00	5191.52	46.15
213	5050.00	0.00	5180.89	56.71
215	5040.00	140.00	5134.97	41.15
217	5030.00	0.00	5135.35	45.65
219	5030.00	2.00	5135.27	45.61
221	5027.00	143.00	5133.83	46.29

^a Connection to Fort Collins city system at the Taft Hill source pump station

Table B2 cont. Node input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
222	5000.00	0.00	5123.43	53.48
223	5010.00	288.00	5131.18	52.51
225	5015.00	24.00	5129.90	49.79
226	5000.00	0.00	5128.18	55.54
227	5020.00	0.00	5191.53	74.33
229	5050.00	145.00	5191.55	61.33
230	5050.00	0.00	5185.42	58.68
231	5000.00	117.00	5194.30	84.19
233	4960.00	0.00	5204.97	106.14
235	5000.00	352.00	5187.46	81.23
237	5020.00	84.00	5185.89	71.88
239	5010.00	213.00	5185.64	76.11
241	5010.00	0.00	5212.79	87.87
243	5000.00	0.00	5193.86	84.00
245	5050.00	22.00	5183.20	57.72
247	5040.00	555.00	5138.80	42.81
249	5057.00	0.00	5189.28	57.32
251	5064.00	81.00	5201.44	59.55
253	5055.00	7.00	5213.08	68.49
255	5064.00	12.00	5208.32	62.53
257	5064.00	131.00	5208.09	62.43
259	5090.00	0.00	5185.74	41.49
261	5030.00	769.00	5181.75	65.76
263	5010.00	442.00	5161.33	65.57
264	5000.00	0.00	5161.25	69.87
265	5030.00	0.00	5160.97	56.75
266	5030.00	0.00	5160.97	56.75
267	5086.00	47.00	5181.68	41.46
269	5165.00	0.00	5170.00	2.17
270	5135.00	0.00	5150.58	6.75
273	5030.00	482.00	5160.12	56.38
275	5030.00	55.00	5160.14	56.39
277	5025.00	55.00	5157.82	57.55
279	5010.00	83.00	5156.40	63.43
281	5000.00	55.00	5156.34	67.74
283	5050.00	55.00	5159.89	47.62
285	5010.00	55.00	5157.35	63.85
287	5010.00	55.00	5157.18	63.77
289	5015.00	55.00	5158.89	62.35
290	5010.00	0.00	5113.42	44.81
291	5010.00	55.00	5159.66	64.85
293	5140.00	6.00	5150.00	4.33
295	5060.00	0.00	5150.00	39.00
297	5060.00	0.00	5160.43	43.52
299	5060.00	0.00	5160.43	43.52
301	5050.00	0.00	5160.19	47.75
303	5030.00	0.00	5159.99	56.32
305	5010.00	6.00	5159.87	64.94
307	5030.00	0.00	5159.82	56.25
309	5005.00	98.00	5159.76	67.06
311	5005.00	0.00	5113.30	46.92

Table B2 cont. Node input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
313	5040.00	0.00	5160.03	52.01
315	5050.00	194.00	5158.26	46.91
317	5135.00	423.00	5162.99	12.13
319	5088.00	22.00	5163.55	32.74
321	5060.00	0.00	5158.20	42.55
323	5050.00	0.00	5174.49	53.94
325	5050.00	0.00	5155.52	45.72
327	5020.00	113.00	5148.62	55.73
329	5030.00	8.00	5146.86	50.64
331	5030.00	0.00	5142.83	48.89
332	5000.00	0.00	5140.68	60.96
333	5017.00	26.00	5159.73	61.85
335	5070.00	0.00	5163.83	40.66
337	5072.00	464.00	5164.11	39.91
339	5072.00	13.00	5166.28	40.85
341	5072.00	0.00	5166.40	40.90
343	5005.00	88.00	5196.86	83.13
345	5005.00	93.00	5192.45	81.22
347	5020.00	88.00	5192.31	74.66
349	5020.00	0.00	5146.86	54.97
351	4990.00	4.00	5190.06	86.68
353	4990.00	93.00	5196.34	89.41
355	5005.00	0.00	5196.34	82.91
357	5005.00	4.00	5152.55	63.93
359	4976.00	29.00	5152.55	76.50
361	4980.00	0.00	5152.42	74.71
365	5050.00	360.00	5166.43	50.45
367	5030.00	79.00	5160.83	56.69
369	4985.00	303.00	5158.28	75.08
373	4980.00	7.00	5152.42	74.71
375	4960.00	0.00	5152.42	83.38
401	5025.00	0.00	5151.93	55.00
403	4985.00	2.00	5143.17	68.53
405	4954.00	40.00	5134.64	78.27
406	5000.00	0.00	5115.39	50.00
407	4990.00	0.00	5112.56	53.10
409	4966.00	96.00	5102.43	59.12
411	4925.00	0.00	5095.21	73.75
413	4925.00	7.00	5091.92	72.32
415	4925.00	2.00	5091.76	72.26
417	4845.00	59.00	5007.84	70.56
419	4845.00	0.00	5007.64	70.47
421	5000.00	114.00	5115.53	50.06
423	4950.00	0.00	5111.02	69.77
425	4940.00	24.00	5110.25	73.77
427	4900.00	452.00	5109.22	90.65
429	4970.00	189.00	5109.21	60.32
431	4920.00	143.00	5108.19	81.54
433	4900.00	560.00	5107.87	90.07
435	5000.00	661.00	5107.02	46.37
437	5000.00	361.00	5106.64	46.21

Table B2 cont. Node input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
439	4980.00	257.00	5107.09	55.07
441	4990.00	123.00	5106.82	50.62
443	4920.00	7.00	5070.70	65.30
445	4920.00	0.00	5070.65	65.27
447	4952.00	36.00	5069.60	50.96
449	4934.00	82.00	5068.82	58.42
451	4917.00	1409.00	5076.32	69.03
453	4886.00	44.00	5021.85	58.86
455	4850.00	0.00	5018.59	73.05
457	4920.00	323.00	5108.34	81.61
459	4910.00	413.00	5108.57	86.04
461	4930.00	235.00	5104.18	75.47
463	4930.00	0.00	5104.12	75.45
465	4950.00	99.00	5104.06	66.75
467	4940.00	99.00	5104.08	71.10
468	5000.00	0.00	5161.25	69.87
469	5000.00	287.00	5161.25	69.87
470	5000.00	0.00	5108.65	47.08
471	4980.00	275.00	5108.65	55.74
473	4909.00	1371.00	5109.28	86.78
474	4969.00	0.00	5118.07	64.59
475	4940.00	243.00	5104.77	71.39
477	4960.00	23.00	5060.36	43.48
479	4923.00	620.00	5059.31	59.06
481	4876.00	29.00	4991.35	49.98
483	4850.00	6.00	4980.80	56.68
485	5005.00	82.00	5156.31	65.56
487	4990.00	0.00	5156.33	72.07
489	4956.00	1143.00	5114.76	68.79
491	4912.00	593.00	5038.83	54.96
493	4870.00	0.00	5066.03	84.94
495	4875.00	6.00	5079.62	88.66
497	4931.00	48.00	5097.22	72.02
498	5000.00	0.00	5096.93	42.00
499	4960.00	26.00	5094.85	58.43
501	4940.00	0.00	5091.10	65.47
503	4939.00	0.00	5087.62	64.40
505	4939.00	23.00	5087.45	64.32
507	4940.00	32.00	5090.90	65.39
509	4940.00	7.00	5090.90	65.38
510	4960.00	0.00	5088.65	55.74
511	4980.00	7.00	5097.70	51.00
512	4960.00	379.00	5073.48	49.17
513	4940.00	25.00	5081.93	61.50
514	4940.00	12.00	5081.89	61.48
515	4940.00	0.00	5081.89	61.48
516	4911.00	0.00	5080.89	73.61
517	4960.00	41.00	5070.93	48.07
519	4930.00	672.00	5080.36	65.15
521	4905.00	44.00	5080.89	76.21
522	4910.00	0.00	5080.89	74.05

Table B2 cont. Node input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Node	Elevation (ft)	Demand (gpm)	Hydraulic Grade (ft)	Pressure (psi)
523	4910.00	6.00	5073.21	70.72
525	4900.00	132.00	5080.81	78.34
527	4900.00	61.00	5080.73	78.31
529	4930.00	0.00	5071.84	61.46
531	4950.00	24.00	5072.47	53.07
533	4950.00	10.00	5068.97	51.55
535	4950.00	0.00	5068.96	51.55
537	4952.00	24.00	5068.80	50.61
539	4970.00	40.00	5062.95	40.28
540	4914.00	100.00	5056.71	61.84
541	4875.00	325.00	5080.70	89.13
542	4920.00	2.00	5080.66	69.61
543	4900.00	0.00	5080.66	78.28
545	4835.00	315.00	5080.66	106.44
603	4984.00	0.00	5166.47	79.06
609	4975.00	561.00	5160.44	80.35
611	4990.00	23.00	5159.95	73.64
613	5005.00	143.00	5159.91	67.12
615	5015.00	0.00	5159.88	62.78
617	5030.00	71.00	5159.67	56.18
619	5045.00	187.00	5159.54	49.63
621	5035.00	35.00	5160.02	54.17
623	5035.00	335.00	5160.04	54.18
625	5034.00	0.00	5162.04	55.48
627	5020.00	260.00	5162.95	61.94
629	5015.00	459.00	5166.16	65.50
631	5007.00	388.00	5173.34	72.08
633	5020.00	62.00	5185.11	71.54
637	4990.00	0.00	5160.40	73.84
639	4983.00	10.00	5160.37	76.85
641	4960.00	10.00	5160.36	86.82
643	4960.00	10.00	5160.36	86.82
645	4980.00	10.00	5160.36	78.15
703	4860.00	0.00	4967.74	46.68
705	4870.00	268.00	4967.72	42.34
707	4865.00	47.00	4967.62	44.46
709	4845.00	6.00	4975.72	56.64
711	4842.00	21.00	4973.08	56.80
713	4843.00	0.00	4970.63	55.30
715	4843.00	52.00	4967.62	54.00
717	4840.00	0.00	4964.58	53.98
719	4850.00	34.00	4951.87	44.14
721	4850.00	50.00	4967.70	51.00
723	4850.00	41.00	4961.54	48.33
725	4850.00	37.00	4951.22	43.86
727	4840.00	82.00	4958.70	51.43
729	4850.00	30.00	4980.00	56.33
731	4818.00	50.00	4980.02	70.20
733	4800.00	48.00	5010.85	91.36
735	4847.00	0.00	4964.64	50.97

Table B3 Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
1	1	3	20.00	8100	110	4647.48	4.75	4.95
3	3	5	20.00	2940	140	4647.48	4.75	3.17
5	5	7	20.00	2940	140	4515.48	4.61	3.01
7	7	9	20.00	660	140	4504.48	4.60	2.99
17	15	17	20.00	660	140	4451.48	4.55	2.93
19	19	17	16.00	4300	140	2480.94	3.96	2.94
21	17	21	20.00	480	140	6932.42	7.08	6.64
23	21	23	8.00	2640	140	9.00	0.06	0.00
25	21	25	18.00	5280	140	3350.80	4.22	2.89
27	21	27	18.00	5280	140	3559.62	4.49	3.23
29	29	27	12.00	2160	140	0.00	0.00	0.00
35	27	33	18.00	2640	140	3376.61	4.26	2.93
37	25	35	18.00	5280	140	3317.80	4.18	2.84
39	35	37	18.00	2640	140	3317.80	4.18	2.84
41	33	39	18.00	5280	140	3347.61	4.22	2.89
43	37	43	18.00	2640	140	3141.80	3.96	2.57
45	39	41	18.00	2640	140	2473.11	3.12	1.65
47	41	43	12.00	200	140	491.88	1.40	0.60
49	43	45	16.00	2640	140	3629.69	5.79	5.95
51	45	47	16.00	500	140	3112.94	4.97	4.48
55	39	49	8.00	1560	140	691.50	4.41	8.10
57	49	51	8.00	1560	140	508.50	3.25	4.59
59	51	53	8.00	1200	140	191.00	1.22	0.75
61	55	57	4.00	1800	140	50.00	1.28	1.84
62	57	59	6.00	1400	140	50.00	0.57	0.25
63	55	60	8.00	3300	140	141.00	0.90	0.43
67	63	51	8.00	1920	140	623.43	3.98	6.69
69	41	63	18.00	2640	140	1871.23	2.36	0.98
71	45	71	8.00	2640	140	406.74	2.60	3.03
73	63	71	4.00	1000	140	187.80	4.79	21.24
75	51	61	8.00	1200	140	616.93	3.94	6.56
77	69	61	8.00	1200	140	-101.93	0.65	0.23
78 ^a	68	76	24.00	7920	140	5095.17	3.61	1.55
79 ^a	76	87	18.00	5280	140	3021.70	3.81	2.39
82	68	69	18.00	2000	140	2304.83	2.91	1.45
83 ^a	68	69	0.0	2000	Duplicate pipe not required			
85	71	73	8.00	5900	140	484.55	3.09	4.19
86	69	75	18.00	3600	140	2399.76	3.03	1.56
87 ^a	69	75	0.0	3600	Duplicate pipe not required			
88	75	76	18.00	2400	140	2399.76	3.03	1.56
89 ^a	75	76	0.0	2400	Duplicate pipe not required			
90	76	77	18.00	1440	140	2654.97	3.35	1.88
91	77	79	18.00	1060	140	2642.97	3.33	1.86
92 ^b	76	241	12.00	5280	140	1818.25	5.16	6.72
93	79	81	18.00	3840	140	2642.97	3.33	1.86
94 ^a	76	77	0.0	1440	Duplicate pipe not required			
95	81	83	18.00	215	140	2639.97	3.33	1.86

^a Duplicate pipes between Taft Hill source pump station (node 68) and Tribby Road

^b Clarendon Hills new pipe

Table B3 cont. Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
96 ^a	77	79	0.0	1060	Duplicate pipe not required			
97	83	87	18.00	200	140	2626.47	3.31	1.84
98 ^a	79	81	0.0	3840	Duplicate pipe not required			
99	85	87	16.00	2640	140	1566.26	2.50	1.26
100 ^c	85	87	0.0	2640	Duplicate pipe not required			
102 ^a	81	83	0.0	215	Duplicate pipe not required			
104 ^a	83	87	0.0	200	Duplicate pipe not required			
201	201	203	16.00	2640	140	3107.73	4.96	4.46
203	203	205	16.00	5280	140	3065.73	4.89	4.35
205	205	207	6.00	100	140	137.90	1.56	1.67
211	209	211	6.00	720	140	197.78	2.24	3.24
213	211	213	6.00	3120	140	203.05	2.30	3.41
215	213	215	6.00	1920	140	582.21	6.61	23.91
217	215	221	10.00	960	140	442.21	1.81	1.19
219	205	217	14.00	5080	140	2840.83	5.92	7.25
221	217	219	14.00	10	140	2840.83	5.92	7.23
223	219	221	14.00	200	140	2838.83	5.92	7.24
225	221	223	12.00	1500	140	883.20	2.51	1.77
226	211	227	10.00	1480	140	-33.28	0.14	0.01
227	223	225	12.00	1500	140	595.20	1.69	0.85
228	225	226	10.00	900	140	571.20	2.33	1.92
229	227	229	10.00	1500	140	-33.28	0.14	0.01
231	229	230	8.00	2300	140	379.16	2.42	2.66
232	230	213	8.00	1700	140	379.16	2.42	2.66
233	221	222	14.00	2200	140	2254.84	4.70	4.73
235	229	231	10.00	1500	140	-557.44	2.28	1.83
237	233	231	10.00	1500	140	1159.93	4.74	7.11
239	231	235	6.00	400	140	485.49	5.51	17.09
241	241	233	10.00	1100	140	1159.93	4.74	7.11
243	237	235	6.00	1000	140	-133.49	1.51	1.57
245	239	237	6.00	1000	140	-49.49	0.56	0.25
247	241	243	6.00	1200	140	464.98	5.28	15.78
249	243	239	6.00	3600	140	163.51	1.86	2.28
251	241	245	4.00	1320	140	193.34	4.94	22.42
253	243	247	4.00	1080	140	301.47	7.70	50.98
255	245	247	4.00	1200	140	253.53	6.47	37.01
257	249	245	4.00	1320	140	82.18	2.10	4.60
259	251	249	4.00	2640	140	82.18	2.10	4.60
261	257	251	3.00	100	140	163.18	7.41	66.49
263	257	259	10.00	2800	140	1234.28	5.04	7.98
264 ^c	257	259	8.00	2800	140	685.97	4.38	7.98
265	259	261	8.00	500	140	685.97	4.38	7.98
266 ^c	259	261	10.00	500	140	1234.28	5.04	7.98
267	261	263	8.00	2000	140	783.74	5.00	10.21
268	263	469	8.00	10	140	709.25	4.53	8.50
269	469	264	14.00	1400	140	0.00	0.00	0.00
270 ^c	261	263	6.00	2000	140	367.52	4.17	10.21

^a Duplicate pipes between Taft Hill source pump station (node 68) and Trilby Road

^c Duplicate pipes along Trilby between the existing Zone 2 Tank (4.0MG) and College Ave

Table B3 cont. Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
271	87	253	18.00	5280	140	4142.38	5.22	4.28
272 ^c	87	253	16.00	5280	140	3038.05	4.85	4.28
273	253	255	10.00	200	140	2226.44	9.09	23.76
274 ^c	253	257	0.0	210	Duplicate pipe not required			
275	255	257	10.00	10	140	2214.44	9.05	23.54
276 ^c	263	469	0.0	10	Duplicate pipe not required			
277	253	267	18.00	5280	140	4947.00	6.24	5.95
279	266	273	14.00	3960	140	422.25	0.88	0.21
281	267	269	18.00	2000	140	4900.00	6.18	5.84
284	270	271	18.00	100	140	4900.00	6.18	5.84
285	271	293	18.00	2640	140	6.00	0.01	0.00
287	293	295	18.00	2440	140	0.00	0.00	0.00
291	299	297	18.00	200	140	0.00	0.00	0.00
292	275	299	14.00	1710	140	-375.43	0.78	0.17
293	273	485	3.00	880	140	37.27	1.69	4.33
294	299	301	14.00	450	140	691.11	1.44	0.53
295	273	275	14.00	930	140	-97.02	0.20	0.01
296	305	307	14.00	1200	140	163.67	0.34	0.04
297	279	485	6.00	450	140	42.61	0.48	0.19
298	301	303	14.00	800	140	464.25	0.97	0.25
299	279	281	6.00	850	140	25.45	0.29	0.07
300	275	277	6.00	570	140	223.41	2.54	4.07
301	277	285	4.00	1800	140	17.35	0.44	0.26
302	277	279	6.00	720	140	151.06	1.71	1.97
303	285	287	4.00	600	140	18.25	0.47	0.28
305	287	281	4.00	1070	140	31.67	0.81	0.79
307	289	285	4.00	680	140	55.90	1.43	2.26
309	289	287	4.00	520	140	68.42	1.75	3.28
310 ^d	295	311	0.0	5480	Duplicate pipe not required			
311	291	289	8.00	1160	140	179.32	1.14	0.67
313	283	291	8.00	1270	140	88.76	0.57	0.18
314	309	333	12.00	1500	140	65.67	0.19	0.01
315	301	283	8.00	675	140	143.76	0.92	0.44
317	303	291	8.00	720	140	145.57	0.93	0.45
318	311	290	20.00	230	140	-1778.15	1.82	0.53
319	303	305	14.00	935	140	318.68	0.66	0.13
321	307	309	14.00	1850	140	163.67	0.34	0.04
325	301	313	8.00	1000	140	83.11	0.53	0.16
327	313	305	8.00	1000	140	83.11	0.53	0.16
329	305	315	8.00	1500	140	232.12	1.48	1.07
331	319	299	14.00	2640	140	1066.54	2.22	1.18
333	315	321	8.00	1500	140	38.12	0.24	0.04
335	319	317	14.00	2640	140	423.00	0.88	0.21
337	319	321	6.00	2000	140	178.22	2.02	2.68
339	323	319	12.00	2640	140	1399.62	3.97	4.14
341	321	325	6.00	700	140	216.34	2.45	3.83
343	325	327	6.00	1800	140	216.34	2.45	3.83

^c Duplicate pipes along Trilby between the existing Zone 2 Tank (4.0MG) and College Ave

^d Duplicate pipe south of new Zone 3 Tank (4.0MG) between College and Lemay

Table B3 cont. Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
345	327	329	6.00	1800	140	103.34	1.17	0.98
347	329	331	6.00	4800	140	95.33	1.08	0.84
349	333	331	3.00	3480	140	39.67	1.80	4.86
350	331	332	6.00	1340	140	135.01	1.53	1.60
353	319	335	14.00	2640	140	-290.14	0.60	0.11
355	347	323	12.00	4300	140	1399.62	3.97	4.14
357	329	349	3.00	2640	140	-0.00	0.00	0.00
359	335	337	14.00	2640	140	-290.14	0.60	0.11
361	339	337	6.00	200	140	379.66	4.31	10.84
363	341	339	6.00	10	140	392.66	4.46	11.57
365	343	341	6.00	2640	140	392.66	4.46	11.54
366	371	343	6.00	385	140	1810.95	20.55	195.16
367	365	337	10.00	2640	140	374.47	1.53	0.88
368	363	365	4.00	450	140	655.60	16.74	214.60
369	365	367	6.00	1320	140	228.57	2.59	4.24
371	367	369	6.00	1320	140	149.57	1.70	1.93
373	343	369	4.00	2640	140	153.43	3.92	14.61
375	343	347	12.00	1000	140	1473.21	4.18	4.56
377	347	345	4.00	800	140	-14.41	0.37	0.18
379	343	345	6.00	1000	140	233.44	2.65	4.41
381	343	353	8.00	2640	140	93.00	0.59	0.20
383	345	351	6.00	1700	140	126.03	1.43	1.41
387	353	355	8.00	200	140	0.00	0.00	0.00
391	351	359	4.00	3920	140	122.03	3.12	9.57
393	357	359	8.00	3920	140	-4.00	0.03	0.00
395	359	373	4.00	2640	140	7.00	0.18	0.05
397	375	373	4.00	2640	140	0.00	0.00	0.00
401	401	403	6.00	5280	140	137.67	1.56	1.66
403	403	405	6.00	5280	140	135.67	1.54	1.62
405	405	409	4.00	5280	140	95.67	2.44	6.10
407	406	407	10.00	1480	140	571.06	2.33	1.92
409	407	409	10.00	5280	140	571.06	2.33	1.92
411	409	411	10.00	5280	140	475.81	1.94	1.37
413	411	413	6.00	200	140	475.81	5.40	16.46
414 ^e	411	413	0.0	200	Duplicate pipe not required			
415	413	415	6.00	10	140	468.81	5.32	16.02
416 ^e	413	415	0.0	10	Duplicate pipe not required			
417	415	417	6.00	5280	140	466.81	5.30	15.89
418 ^e	415	417	0.0	5280	Duplicate pipe not required			
419	417	419	10.00	200	140	407.81	1.67	1.03
423	421	423	14.00	1050	140	2140.84	4.46	4.29
425	423	425	14.00	500	140	1228.97	2.56	1.54
427	423	427	12.00	960	140	911.87	2.59	1.88
431	425	429	14.00	700	140	1204.97	2.51	1.48
433	429	427	6.00	1800	140	-6.03	0.07	0.00
435	435	429	14.00	2000	140	-1022.00	2.13	1.09
437	427	431	12.00	2000	140	453.84	1.29	0.52
439	431	433	14.00	2640	140	310.84	0.65	0.12

^e Duplicate pipes along Harmony Road between County Rd 7 and County Rd 9

Table B3 cont. Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
441	409	443	4.00	5280	140	94.91	2.42	6.01
442 ^f	409	443	0.0	5280	Duplicate pipe not required			
443	433	439	14.00	1525	140	675.84	1.41	0.51
445	439	441	12.00	600	140	418.84	1.19	0.45
447	441	461	8.00	1570	140	295.84	1.89	1.68
448 ^g	450	451	10.00	1640	140	1813.90	7.41	16.27
449	443	445	4.00	10	140	87.91	2.24	5.22
450 ^f	443	445	0.0	10	Duplicate pipe not required			
451	445	447	4.00	200	140	87.91	2.24	5.22
452 ^f	445	447	0.0	200	Duplicate pipe not required			
453	447	449	4.00	2640	140	18.68	0.48	0.30
454 ^h	447	449	0.0	2640	Duplicate pipe not required			
455	451	449	4.00	2640	140	63.32	1.62	2.84
456 ^h	451	449	0.0	2640	Duplicate pipe not required			
457	451	453	3.00	5280	140	59.60	2.71	10.32
458 ^h	451	453	4.00	5480	140	124.58	3.18	9.94
459	453	455	3.00	200	140	76.37	3.47	16.32
462	469	468	12.00	600	140	0.00	0.00	0.00
463	453	481	3.00	2606	140	63.81	2.90	11.70
464	469	265	14.00	1320	140	422.25	0.88	0.21
465	437	435	14.00	2400	140	-361.00	0.75	0.16
466	470	471	12.00	1580	140	0.00	0.00	0.00
467	457	433	20.00	2940	140	924.99	0.94	0.16
468	471	473	12.00	3100	140	-275.00	0.78	0.20
469	461	463	8.00	700	140	60.84	0.39	0.09
470	265	266	14.00	10	140	422.25	0.88	0.20
471	463	465	8.00	650	140	60.84	0.39	0.09
472	459	457	20.00	820	140	1247.99	1.27	0.28
473	467	465	8.00	685	140	38.16	0.24	0.04
474	473	459	20.00	1520	140	1660.99	1.70	0.47
475	475	467	8.00	1680	140	137.16	0.88	0.41
477	473	475	12.00	2640	140	867.66	2.46	1.71
478 ⁱ	451	479	6.00	8000	140	157.39	1.79	2.13
479	447	477	3.00	2640	140	33.24	1.51	3.50
480 ^j	475	479	6.00	2640	140	487.49	5.53	17.22
481	479	477	3.00	2640	140	-10.24	0.46	0.40
482 ^k	271	474	24.00	10000	140	7538.07	5.35	3.19
483	481	483	3.00	2766	140	34.81	1.58	3.81
484 ^l	489	491	6.00	5280	140	442.27	5.02	14.38
487	479	491	3.00	5280	140	35.13	1.59	3.88
488	474	473	20.00	3380	140	4174.65	4.26	2.60
489	485	487	3.00	940	140	-2.12	0.10	0.02

^f Duplicate pipes along Timberline between Harmony and County Rd 36

^g New connection to Fort Collins city system at County Rd 9

^h Duplicate pipes along along County Rd 36 between Timberline and County Rd 7

ⁱ New pipe from County Rd 9 southwest to Trilby/Timberline

^j New pipe from Trilby/Timberline to Paragon Point

^k New pipe from new Zone 3 Tank (4.0MG) east to Lemay

^l New pipe along County Rd 12 between Lemay and Timberline

Table B3 cont. Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
490	487	281	4.00	540	140	-2.12	0.05	0.01
491	491	493	3.00	2640	140	-59.56	2.70	10.30
492	474	489	20.00	1900	140	3363.42	3.43	1.74
493	493	495	3.00	1320	140	-59.56	2.70	10.30
495	491	505	3.00	5280	140	-56.05	2.54	9.21
496	290	489	20.00	2500	140	-1778.15	1.82	0.54
497	311	497	14.00	5280	140	1778.15	3.71	3.05
499	503	505	3.00	10	140	79.05	3.59	17.38
501	501	503	3.00	200	140	79.05	3.59	17.40
502	498	499	6.00	1300	140	134.98	1.53	1.60
503	499	501	6.00	1320	140	184.00	2.09	2.84
504 ^m	491	497	0.0	2640	Duplicate pipe not required			
505	501	507	6.00	200	140	104.95	1.19	1.01
506 ^m	505	497	0.0	2430	Duplicate pipe not required			
507	507	509	6.00	10	140	72.95	0.83	0.49
508 ^m	503	505	0.0	10	Duplicate pipe not required			
509	509	510	6.00	5280	140	65.95	0.75	0.43
510 ^m	501	503	0.0	200	Duplicate pipe not required			
511	511	499	6.00	5280	140	75.02	0.85	0.54
515	359	361	6.00	200	140	82.03	0.93	0.64
516 ⁿ	491	493	0.0	2640	Duplicate pipe not required			
517	497	513	14.00	5280	140	1730.15	3.61	2.90
518 ⁿ	493	495	0.0	1320	Duplicate pipe not required			
519	513	514	14.00	200	140	411.68	0.86	0.20
520	513	512	12.00	2588	140	1230.09	3.49	3.26
521	514	515	14.00	10	140	399.68	0.83	0.20
522 ^o	495	516	6.00	3000	140	-65.56	0.74	0.42
523	515	519	14.00	7920	140	399.68	0.83	0.19
524 ^p	516	513	6.00	2640	140	-63.39	0.72	0.40
525	510	531	4.00	5280	140	65.95	1.68	3.06
526 ^q	513	512	0.0	2588	Duplicate pipe not required			
527	531	533	4.00	2640	140	41.95	1.07	1.33
528 ^r	516	521	6.00	8000	140	-2.17	0.02	0.00
529	533	535	4.00	10	140	31.95	0.82	0.78
531	535	537	4.00	200	140	31.95	0.82	0.80
533	537	539	3.00	2640	140	25.95	1.18	2.21
535	523	537	3.00	3920	140	18.00	0.82	1.13
537	529	539	3.00	2640	140	32.54	1.48	3.37
539	521	519	12.00	2640	140	272.32	0.77	0.20
540	539	540	3.00	5280	140	18.48	0.84	1.18
541	527	523	3.00	3920	140	24.00	1.09	1.92
542	542	540	4.00	5280	140	81.52	2.08	4.54
543	522	521	30.00	920	140	318.49	0.14	0.00

^m Duplicate pipes along Timberline between County Rd 30 and County Rd 32

ⁿ Duck Lake duplicate pipes east of Timberline

^o Duck Lake new pipe east of Timberline

^p Duck Lake new pipe south along County Rd 9

^q County Rd 9 duplicate pipes between County Rd 30 1/2 and County Rd 28

^r Duck Lake new pipe east to Interstate 25

Table B3 cont. Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
544	522	525	30.00	3000	140	1072.25	0.49	0.03
545	527	529	3.00	2640	140	32.54	1.48	3.37
547	527	733	3.00	2640	140	99.19	4.50	26.47
548	525	527	6.00	20	140	216.73	2.46	3.83
549	525	541	24.00	2640	140	723.52	0.51	0.04
551	541	543	24.00	2640	140	398.52	0.28	0.01
553	543	545	24.00	500	140	315.00	0.22	0.01
555	512	517	12.00	1545	140	851.09	2.41	1.65
557	543	542	12.00	200	140	83.52	0.24	0.02
601 ^q	512	517	0.0	1545	Duplicate pipe not required			
609	603	609	12.00	4000	140	810.09	2.30	1.51
610 ^q	603	609	0.0	4000	Duplicate pipe not required			
611	609	611	12.00	3978	140	209.09	0.59	0.12
613	611	613	12.00	410	140	186.09	0.53	0.10
615	613	615	12.00	1279	140	82.53	0.23	0.02
617	615	617	12.00	1200	140	258.00	0.73	0.18
619	617	619	12.00	1200	140	187.00	0.53	0.10
620 ^s	624	522	16.00	12000	140	1390.74	2.22	1.01
621	613	621	8.00	2655	140	-39.44	0.25	0.04
623	615	623	12.00	1771	140	-175.47	0.50	0.09
625	621	623	12.00	1200	140	-74.44	0.21	0.02
627	623	625	12.00	2425	140	-584.91	1.66	0.82
629	625	627	12.00	1100	140	-584.91	1.66	0.82
631	629	627	12.00	1975	140	844.91	2.40	1.63
633	631	629	12.00	1975	140	1303.91	3.70	3.63
635	633	631	12.00	2000	140	1691.91	4.80	5.89
637	635	633	12.00	300	140	1753.91	4.98	6.29
639	609	637	10.00	2640	140	40.00	0.16	0.01
641	637	639	10.00	2640	140	40.00	0.16	0.01
643	639	641	10.00	1000	140	18.11	0.07	0.00
645	641	643	10.00	1000	140	8.11	0.03	0.00
647	645	643	10.00	1000	140	1.89	0.01	0.00
649	639	645	10.00	2640	140	11.89	0.05	0.00
701	701	703	12.00	3800	140	153.82	0.44	0.07
703	705	703	12.00	2640	140	-32.78	0.09	0.00
705	703	707	12.00	2640	140	121.03	0.34	0.04
707	713	705	8.00	2640	140	235.22	1.50	1.10
709	709	711	10.00	2640	140	401.81	1.64	1.00
711	711	713	10.00	2700	140	380.81	1.56	0.91
713	713	715	3.00	100	140	106.46	4.83	30.17
715	715	717	3.00	3960	140	14.63	0.66	0.77
717	715	719	3.00	3000	140	41.37	1.88	5.25
719	713	707	4.00	2584	140	39.14	1.00	1.17
721	717	723	3.00	3960	140	14.63	0.66	0.77
723	721	723	3.00	2700	140	26.37	1.20	2.28
725	719	725	3.00	3000	140	7.37	0.33	0.22
727	727	725	3.00	2640	140	29.63	1.34	2.83

^q County Rd 9 duplicate pipes between County Rd 30 1/2 and County Rd 28

^s New pipe up to the new McCloughan Hill Tank

Table B3 cont. Pipe input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pipe	Start node	End node	Diameter (inches)	Length (ft)	Hazen-Williams C	Pipe flow (gpm)	Velocity (fps)	Headloss /1000ft
729	729	731	3.00	2640	140	-1.19	0.05	0.01
731	733	731	3.00	3960	140	51.19	2.32	7.79
733 ^t	715	707	6.00	2640	140	-1.54	0.02	0.00
735 ^t	707	735	6.00	2640	140	111.63	1.27	1.13
737 ^t	735	727	6.00	5280	140	111.63	1.27	1.13
800 ^u	371	343	4.00	385	140	622.81	15.90	195.16
801 ^u	363	365	3.00	450	140	307.43	13.95	214.60

^t Loop of new pipes at County Rd 3

^u Duplicate pipes which connect to the adjacent City of Loveland system

Table B4 Pump station (PS) input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Pump station description	Pipe	Start node	End node	No. of pumps operating	Rated flow (gpm)	Rated head (ft)	Flow (gpm)	Power output (hp)	Pump lift (ft)
Overland Trail source PS	9	9	15	3	4252	165.8	4504.48	181	159.04
Westridge booster PS	60	53	55	1	191	79	191.00	4	79.00
Airport booster PS	603	517	603	1 ^a	620	125	810.09	20	95.53

^a ESP-1 operating curve

Table B5 Pressure reducing valve (PRV) and flow control valve (FCV) input data and output results for the EPANET simulation of the Genetic Algorithm design subject to the 2015 peak hour demands

Valve description	Pipe	Start node	End node	Diameter (inches)	Pressure setting (psi)	Status	Flow (gpm)	Velocity (fps)	Head loss (ft)
PRV 1	53	47	201	8.00	42	Operating	3112.94	19.87	56.24
PRV 3	207	207	401	4.00	55	Operating	137.90	3.52	20.06
PRV 2	209	73	209	4.00	45	Operating	199.55	5.09	38.80
FCV 32	283	269	270	18.00	4900 ^a	Operating	4900.00	6.18	19.41
PRV 29	351	332	498	6.00	42	Operating	135.01	1.53	43.75
PRV 23	389	355	357	4.00	43	Closed	0.00	-	-
PRV 28	404	226	406	6.00	50	Operating	571.20	6.48	12.78
PRV 27	420	222	421	10.00	50	Operating	2254.84	9.21	7.90
PRV 8	421	419	709	3.00	75	Not operating	407.81	18.51	31.92
PRV 31	460	264	437	10.00	41	Closed	0.0	-	-
PRV 9	461	455	721	2.00	51	Operating	76.37	7.80	50.89
PRV 30	476	468	470	12.00	43	Closed	0.0	-	-
PRV 15	485	483	729	2.00	100	Not operating	28.81	2.94	0.81
PRV 14	513	361	511	3.00	51	Operating	82.03	3.72	54.72

^a Flow setting for the FCV 32