MESSY GENETIC ALGORITHMS FOR OPTIMISATION OF WATER DISTRIBUTION SYSTEMS INCLUDING WATER HAMMER

BY

ZHENG YI WU

B.Sc., Civil Engineering, Guizhou Institute of Technology, Guizhou, China, 1983

M.SC., Structural Engineering, Guizhou Institute of Technology, Guizhou, China, 1986

M.Sc., Hydroinformatics, International Institute for Infrastructural, Hydraulic & Environmental Engineering, Delft, The Netherlands, 1994

THESIS

Submitted in fulfillment of the requirements for

the degree of Doctor of Philosophy in Civil & Environmental Engineering

in Department of Civil & Environmental Engineering

at the University of Adelaide, 1998

Adelaide, South Australia

DECLARATION

This thesis 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 published or written 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.

Zheng Yi Wu
Date: 4/3/1999

ABSTRACT

A standard and/or improved GA, using a fixed-length genotype and crossover operation, have been widely used in science and engineering disciplines. It is well-known that the GA is robust in searching for the optimal combination of diameter and rehabilitation actions of water distribution systems but requires a large number of evaluations. A characteristic, which has been observed from the studies, is that the optimal solution of the design and rehabilitation of water distribution systems is located at the boundary of the feasible and infeasible regions of the search space. Previous research has not considered the sizing of pipe wall thicknesses and water hammer protection measures, which are always required in reality. In this research, first of all, the original messy genetic algorithm is applied to the optimisation of design and rehabilitation of water distribution systems. It has been found that the messy GA is more efficient and effective than the standard and improved GAs at solving the optimisation of water distribution systems, but it requires a huge initial population size. This has been overcome by introducing the fast messy GA. Secondly, a scheme of co-evolutionary and self-adaptive penalty has been proposed for the GA solving a constrained boundary optimisation problem. It is purposely designed to guide the GA to search the boundary of the feasible and infeasible regions of the search space. It has been shown that this approach is very effective and efficient for the optimisation of water distribution systems. Finally, the hydraulic network solver (EPANET) for steady state simulation has been incorporated into a transient model for simulation of water hammer in water distribution systems. A methodology for comprehensive optimisation of pipe diameters, pipe classes and surge tanks of the water distribution systems has been developed by carefully integrating the steady state hydraulic solver, the water hammer simulation model, the fast messy GA and the boundary search strategy.

ACKNOWLEDGMENT

Looking at the past and the present, I appreciate all the people who have contributed to my life and have helped me having been through all the way of my education. I am where I am now because of them. I take this chance to express my gratitude to some of them.

First and foremost, I just wish that I can share this moment with my parents and tell them that my success, both my previous success and my future success, belongs to them more than to myself. They sacrificed their life to support their children's education. I have never heard any grievance for that. I am deeply indebted to their love and support in my life.

I gratefully acknowledge all the international financial support, particularly, the Overseas Postgraduate Research Scholarship (OPRS) provided by Australia government, Scholarship of the University of Adelaide and Frank Perry Scholarship provided by the University of Adelaide. It is these financial aids that enabled me to pursue my Ph.D research.

During the past three years, my supervisor, Dr. Angus R. Simpson, is the person I have learned so much from in many aspects. His weekly meeting has been the most helpful to clarify my progress of where I was and where I should go. His consistently constructive and insightful suggestions have encouraged me to overcome the difficulties on the way of the research work. His contribution to my study helps me identifying my own identity. I appreciate his invaluable contribution to my Ph.D research work.

This research would not been carried out without the support from many organisations. I would like to express my gratitude to the following people and organisations for free access information and software. They are:

- Illinois Genetic Algorithms Laboratory, University of Illinios at Urbana-Champaign, Urbana, for free use of computer source code of the original messy genetic algorithm, developed by Deb K. and Goldberg D.E.;
- U.S. Environmental Protection Agency, Cincinnati, for free use of computer source code of EPANET, a hydraulic network solver developed by Rossman L.A.;
- Department of Civil & Environmental Engineering and Department of Electrical and Electronic Engineering, the University of Adelaide, for free use of computer

source code for transient modelling of hydro-electric systems, developed by Angus R. Simpson, John McPheat and Michael J. Gibbard;

- James Hardie Pipelines, South Australia, for providing the wall thickness information of Hobas pipes;
- Dragan A. Savic from School of Engineering, University of Exeter and Halhal D. from Water and Electricity Distribution Co. (RAID), Tangier, Morocco, for providing the data of a real Moroccan water distribution network;
- Paul Doherty from South Australia Water Cooperation for providing pipe and surge tank cost information for Loveday irrigation network in South Australia and also for the insightful discussions about the transient modelling of Loveday network.

Department of Civil & Environmental Engineering at the University of Adelaide is a nice place to work. The Departmental secretaries have been great at helping me to get paper work done. The computing officer is always available either for solving a computer network problem or for helping at computer programming. Many other academic staff and postgraduate students have been really wonderful. I thank you all for everything.

My thanks are also extended to anonymous examiners for spending their valuable time on reviewing my thesis.

Last and the most important, I am deeply indebted to my family, my wife Feng Hua and my daughter Xuan and Annie. My wife has been through all the ups and downs with me since we met in 1986. She sacrificed her career to support me completing my degree. I have been blessed by ever sweetly smiling Xuan and Annie when I am home after working on computer for long hours every day. I would not be able to complete my Ph.D without my wife and my daughters' love and support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF NOTATION

a	exponent of the fitted cost function;
$a^{d}_{n,dex}$	*dex*-th gene bit representing diameter of pipe *n;*
$a^{e}_{r,edex}$	*edex*-th gene bit representing rehabilitation event *r;*
b	coefficient of the fitted cost function;
c_f	fitted cost function of continuous pipe diameters;
$c_{h,k}$	unit cost of a rehabilitation event e^0_h with diameter of d^0_k;
c_n	unit length cost of pipe size *n;*
$c_n(d_n)$	cost of per unit length of new pipe *n* with diameter d_n;
$c_{0,k}$	unit cost of a new pipe with a diameter of d^0_k;
$c_r(d_r, e_r)$	cost of unit length of rehabilitated pipe *r* with diameter d_r and event e_r;
c_{tank}	cost of unit perimeter area of surge tank;
c_{rep}	unit cost (dollars/break) of pipe break repair;
cls_n	pipe class variable of pipe *n;*
cls^0_{ci}	*ci*-th available pipe class;
C	a set of corresponding unit length costs;
C^n_f	cost of pipe *n* evaluated by the fitted cost function;
C_n	Hazen-Williams coefficient of pipe *n;*
C_{nn}	cost of design solution *nn*;
C_{new}	cost of new pipes added to a water distribution system;
C_{pum}	energy cost of pumps in the operation period for its design life (in years);
C_{reh}	cost of rehabilitation pipes;
C_{rep}	cost of pipe repair;
C^n_s	cost of pipe *n* by the split pipe cost evaluation;
C_{tank}	cost of surge tank;
C_{total}	total cost of water distribution network;
CI	number of available pipe classes
CLS^0	set of commercially available pipe class;
CLS	set of pipe class variables;

C_{total}	total cost of a water distribution system;
d^{max}	maximum diameter of split pipe model;
d^{min}	minimum diameter of split pipe model;
d_n	diameter of pipe *n;*
d_n^d	diameter of downstream pipe segment;
$d_n^{\max}$	maximum diameter of pipe *n;*
$d_n^{\min}$	minimum diameter of pipe *n;*
d_k^0	diameter of *k*-th commercially available pipe;
d_n^u	diameter of upstream pipe segment;
d_{tn}^0	*tn*-th possible diameter of surge tank;
dd	number of bits representing the diameter of an expanded or rehabilitated pipe;
$\vec{D}$	vector of pipe diameters;
D^0	set of commercially available discrete pipe size diameters;
D_{tank}	diameter of surge tank;
D_{tank}^0	set of possible diameters of surge tank;
e_h^0	*h*-th rehabilitation event that may applied to the existing pipe *r*;
e_r	rehabilitation action of pipe *r*;
ee	number of bits representing a rehabilitation action for an existing pipe;
$\vec{E}$	vector of pipe rehabilitation actions and associated pipe diameters;
E^0	set of possible rehabilitation actions;
E_k	efficiency of pump *k*;
$g_j(x)$	*j*-th constraint of nonlinear programming problem;
$\vec{g}_{nn}(t)$	genotype *nn;*
$G(t)$	population of genotypes at generation *t;*
h_{fn}	head loss of pipe *n*;
h_{fn}^d	head loss of downstream segment of pipe *n*;
h_{fn}^u	head loss of upstream segment of pipe *n*;
hh	number of the rehabilitation events applicable to pipes;

H_j	hydraulic grade at node *j*;
H_j^{min}	minimum allowable hydraulic grade at node *j*;
H_j^{max}	maximum allowable hydraulic grade at node *j*;
H_n^{max}	maximum allowable transient pressure head of pipe *n*.
$H_{tr,n}^{srg}$	maximum transient pressure head of pipe *n* under transient loading *tr*;
HP_k	horse power of pump *k*;
$\underline{HP}_k$	minimum horse power of pump *k*;
I	number of steady state loading cases;
J	number of nodes in system (excluding fixed grade nodes);
$J(t)$	break rate in year *t*;
J_0	break rate in year 0;
k	order of deceptive nonlinearities;
kk	number of the commercially available pipe sizes;
L_n	length of new pipe *n*;
L_n^d	length of downstream pipe segment;
L_n^u	length of upstream pipe segment;
l	problem length;
l'	length of initial strings;
l^k	length of order *k* strings;
M_k	unit cost (dollars/kWh) of electricity for pump *k*;
N	number of new pipes added to a water distribution system;
NN	population size;
$\vec{P}_{nn}$	population of phenotype representing one design solution;
pp	number of bits coding a pipe undergoing rehabilitation;
P_c	probability of *cut* operator;
P_b	expected break rate of the unit length pipe *n*;
P_k	bitwise cut probability;
P_s	probability of *splice* operator;
Q_n	discharge of pipe *n*;
R	number of existing pipes to be rehabilitated;

R^n	n-dimensional Euclidean space;
r	interest rate;
s	tournament selection pressure;
TR	number of possible diameters of surge tank;
V	flow velocity;
x	vector of variables x_i;
x_i	i-th nonlinear optimisation variable
x_i^{max}	upper bound of variable x_i;
x_i^{min}	lower bound of variable x_i
Y	planning period in years;
λ	length of the string;
θ	number of common genes in the two strings;
Φ_{nn}	fitness of genotype nn;
ΔT_{tk}	time of pump k in operation period t during one day;
γ	penalty factor;
τ	cooling *temperature;*

1. INTRODUCTION

"Engineering is defined as the art or science of making practical application of the knowledge of pure science such as physics, chemistry, biology, etc. This implies that the task of engineering is to synthesise, or put together, useful systems by applying knowledge and methods derived from the pure science."

by Edward J. Haug and Jasbir S. Arora, 1979.

Water is essential for each community in our society. People have been building water distribution systems for thousands of years yet it is not possible to say with any degree of confidence that a particular distribution system is the least cost system that could have been implemented. Archaeologists have shown that ceramic pipes and brick aqueducts were built 5000 years ago in Sindhu-valley civilisation. Rome had a well-developed water supply system 2000 years ago. The system was built from a wide range of materials such as clay, bored-stone, lead and bronze. Modern civilisation started designing and constructing public water supply systems for population centers in late 19th century. Since then a water distribution system has become one of the major requirements in urban and regional economic development.

With the economic development of a modern society, the design and construction of water distribution systems requires hydraulic analysis of the systems. The hydraulic analysis is usually carried out by establishing a numerical model that simulates the flow within the water distribution system. Modelling of a water distribution system, as hydraulic modelling in general, has been facilitated by advent of computer. It is the wide application of computer technology that the hydraulic modelling, in general, and water distribution system modelling, in particular, has been become every day tool of hydraulic engineers for analysis of water distribution systems.

Computer modelling of water distribution systems has provided a practical possibility of *what-if* decision-making of the system. The numerical model for simulation of a water distribution system has been developed as a product which encapsulates the hydraulics equations that govern the flow behaviour of a physical system. The product has usually been presented as a menu-driven and user-friendly computer software. It is easy to use. Thus it becomes a standard tool. It is the tool, by which the numerical model of a large-scale network system is established, that makes the network data that is originally only *available* to become readily *accessible* through the numerical model (Abbott 1992). The accessibility of the large set of data of the physical system provides the possibility for engineers and/or decision-maker to understand the flow behaviour of complicated water systems. Thus the model enables the engineer to answer the *what-if* questions for design and construction of the water systems. It is practically impossible, however, that a cost effective design can be found by trial and error of answering the *what-if* questions.

In profit-driven industry, as well as in government authorities, the objective is to maximise the value or to reduce the cost of the system, while satisfying constraints on resources, performance, and human limitations. Once the cost function or measure of value is chosen and the constraints are identified, the system designer would like to have a method by which optimum design is found. The development of hydraulic modelling computer software systems has laid down a sound basis for optimisation of water distribution system. Thus the development of a computer-aided tool for optimisation of water distribution networks serves as a way of achieving cost-effective networks. This is one of places *where the waters of the world and our current informational revolution come together* (Abbott 1992).

1.1 The Need for Robust and Efficient Optimisation

With the development of high speed digital computers and improved optimisation techniques, optimisation of designs for water distribution networks has been investigated since the 1960's. Walski (1985) and Goulter (1987) both predicted that within the next decade water distribution optimisation models should become everyday tools of practicing water engineers. However, it has been found (Goulter 1992) that optimisation models have not been widely used, or even show signs of being accepted. This implies that there is still a need, in general, for developing optimisation models and techniques for water distribution systems, and in particular, developing a methodology that is of practical use from an engineering point of view.

It is difficult, however, that the problem of optimisation of water distribution system is solved without compromising efficiency, accuracy and completeness of the problem. The difficulty arises from many aspects such as

1. nonlinear constraints i.e. hydraulic heads in relation to pipe sizes;
2. mixed continuous and discrete decision variables;
3. nonconvexity of the feasible solution regions;
4. existence of multiple local optimal solutions;
5. high dimensionality of optimisation problems.

First of all, the optimisation of a water distribution system is a nonlinear-constrained optimisation problem. It is usually subjected to satisfying the minimum hydraulic pressure heads at nodes in the system. The constraint of the pressure head is given by a head loss equation and the energy conservation law for looped network systems, and the head loss is a nonlinear function of decision variables such as pipe diameters and velocities. Secondly, optimisation of the water system requires sizing not only the pipe diameters, but also pump capacities, valve locations and settings, and also the storage tanks. The pipe diameters, locations of valves, pump stations and storage tanks are discrete variables while the pump capacities and valve settings are continuous. Thus the problem becomes a mixed continuous and discrete optimisation problem. Furthermore, the research in literature has shown that optimisation of water distribution systems is a nonconvex optimisation problem, and that there exists plenty of local optima in a solution region. In addition to the nonconvexity, multiple local optimal solutions, the mixed decision variables and the constraint nonlinearity, optimisation of water distribution system, in reality, often involves hundreds of pipes and dozen of valves and pumps. The problem, therefore, is a highly dimensional optimisation problem. No single model or algorithm is able to solve the problem effectively and efficiently. Although not every case of optimisation of water distribution networks contains all the characteristics mentioned above, it is essential to develop an optimisation methodology that is able to handle these characteristics more efficiently and effectively than previous methods. In this research, a state-of-the-art evolutionary computation technique is employed to develop an efficient and effective methodology for optimisation of water distribution systems including water hammer loadings. Consideration of transient loadings allows not only pipe diameters, but also pipe wall thicknesses (pipe classes) and surge pressure protection devices to be optimised. This is the first time, to the best of my knowledge, that water hammer loadings has been included in the optimisation of water distribution systems.

1.2 Objectives and Thesis Overview

The objectives of this research are:

1. To develop efficient and effective algorithms for optimal design of water distribution systems taking account of water hammer loadings (e.g. high pressures due to a rapid operation of an outlet valve).
2. To apply messy genetic algorithm to optimisation of water distribution systems.
3. To compare the performance of GA paradigms including standard GA, improved GA and the messy GA for optimisation of pipeline networks.
4. To evaluate the improvement in design when large water distribution networks are optimised taking into account both water hammer loadings and steady state conditions simultaneously.

To achieve these goals, the research focuses on developing an insight into the methodology. The thesis is organised into 10 Chapters including the introduction presented in this Chapter. Chapter 2 contains a review of previous research in the area of the optimisation of water distribution systems. The review has classified previous optimisation methods into traditional methods and evolutionary optimisation techniques, and also discussed their drawbacks and the motivation for developing a more efficient and robust algorithm for the optimisation of water distribution systems. It eventually outlines the scope of this research. Chapter 3 proposes a discrete model and a generalised formulation for applying a genetic algorithm (GA) for optimisation of water distribution systems. The original messy GA, using complete enumerative initialisation scheme, is employed to solve the problem. Chapter 4 presents an analysis of a continuous optimisation model using a fitted cost function. It leads to a formulation of a split pipe optimisation of water distribution systems to avoid fitting the cost function from a set of discrete cost data.

Chapter 5 extends the discrete model introduced in Chapter 3 to a model for optimal rehabilitation of water distribution systems. The original messy GA is applied to the problem of the optimal rehabilitation. Chapter 6 applies a fast messy GA to improve the original messy GA by introducing a probabilistically complete initialisation and gene filtering scheme to replace the complete enumerative initialisation in the original messy GA. The fast messy GA is applied to large-scale optimisation of water distribution systems. Chapter 7 proposes an approach for a boundary search within genetic algorithm optimisation by using a self-adaptation and a co-evolutionary penalty factor. This approach has been applied to the optimisation of water distribution systems. It is shown that the boundary search genetic algorithm improves the efficiency and effectiveness of the optimisation procedure. Chapter 8 discusses the method of characteristic (MOC) and implements the MOC for transient analysis of water distribution systems. A comprehensive transient analysis of a low head irrigation system is carried out to evaluate critical water hammer loadings. The water hammer model is verified by using LIQT, a widely-used commercial program for transient analysis of pipeline systems. This paves the way for considering transient loadings in the optimisation model. Chapter 9 suggests a model for optimal transient design of water distribution systems including water hammer. It optimises not only pipe diameters but also pipe wall thicknesses and pressure surge protection devices. Finally, conclusions from the research are drawn and the recommendations for the future work are made in Chapter 10.

2. LITERATURE REVIEW

Optimisation of water supply systems is undertaken to find out the most cost effective pipeline diameters, components and configurations while meeting water demands at different locations. The total cost of water distribution systems is minimised by searching for optimum pipe diameters and the preferred layout. This problem has been solved by using different methods in the past. They are generally classified as traditional optimisation approaches and evolutionary optimisation techniques. A review of both approaches is given as follows.

2.1 Traditional Optimisation Approaches

Comprehensive reviews of optimisation of water distribution systems have been made by different researchers (Walski 1985; Goulter 1987; Walters 1988; Lansey and Mays 1989b; Goulter 1992 and Dandy et al. 1993). In this section, a summary of optimisation of water distribution systems published up through 1993 is given by using previous reviews as a guide and then subsequent works in this area are reviewed in detail. The review is followed by classifying traditional approaches for optimisation of water distribution systems as the methods without using a hydraulic network solver and the methods using a hydraulic network solver.

2.1.1 Optimisation methods without using a network solver

For methods that do not use a hydraulic network solver, early research works assumed that the flow distribution in water distribution systems was fixed. A set of length variables, each of them corresponding to a discrete diameter, was applied to one pipe between a pair of nodes. The cost of the pipeline was a linear function of the length. This enabled linear programming (LP) technique to be applied to optimisation of water distribution systems. The assumption of the fixed flow distribution, however, is only valid for branched systems. Water distribution systems commonly are found to be looped systems. The flow in a pipe of a looped system does not only depend on the adjacent pipe but also other pipes in the system. A breakthrough was made for optimisation of looped water distribution systems by introducing a method called linear programming gradient (LPG) (Alperovits and Shamir 1977). The LPG method decomposed the optimisation problem into two stages: (1) flow variables are kept constant while the network is optimised by using LP; and (2) a search technique is employed to determine how the flow variables held constant in the first stage should be changed so that the solution improves. The flow variables are initially assumed by engineering judgment and modified according to the gradient of the objective function (GOF) with respect to the flows. The two-stage iterative procedure is continued until no further reduction of network cost occurs. The problem is elegantly solved at the first stage by using LP for a fixed flow distribution. The optimality of the final solution is governed by the computation of GOF. In other words, how to achieve the optimal flow of the network in the second stage is the key for the LPG methods.

Since Alperovits and Shamir (1977) proposed the original LPG method, a great deal of work has been done to improve the LPG method by introducing different schemes for GOF computation. Alperovits and Shamir (1977) suggested the use of the simple gradient search in determining the direction of the GOF to modify the link flows, and a fixed step size in the selected gradient direction. Quindry et al. (1979) corrected the GOF expression suggested by Alperovits and Shamir (1977) by considering the interaction of the paths and loops in the head constraints for the demand nodes. Quindry et al. (1981) also proposed an approach, similar to that of Alperovits and Shamir (1977), using assumed nodal heads instead of assumed link flows. The nodal heads are subsequently corrected by using the GOF with respect to the assumed nodal heads, and an optimal solution is obtained. Fujiwara et al. (1987) presented a full derivation of the correct gradient expressions, and suggested the use of (1) a quasi-Newton method to determine the direction of the flow revision; and (2) a backtracking line search method to determine the step size. They observed that the final solution was sensitive to the step size, perhaps because there existed many local optimal solutions, and therefore they suggested the use of several step sizes before a final solution was accepted. An analysis of LPG methods was carried out by Kessler and Shamir (1989). They introduced the projection of the GOF onto the constraint surface, and thus it guaranteed that a truly local optimal solution could be obtained. The philosophy of the modified LPG method was analysed by Bhave and Sonak (1992). It is shown that the LPG method is inefficient, even for the optimisation of illustrative two-loop network, as compared to a heuristic method that initially identified logically good branching configuration for obtaining the local optimal solution.

Some efforts have been made to improve the LPG methods to achieve the global optimal solution. Eiger et al. (1994) extended the LPG method by applying a branch and bound algorithm, using non-smooth optimisation and duality theory. A methodology was developed for reducing the duality gap that enabled the computation of a tight lower bound of the global optimal solution. The global optimality of the final solution is measured by the difference between the tight lower bound and the solution. It is served as an optimality criteria for optimisation of water distribution systems. This improved LGP method demonstrated its superiority over previous methods in literature. Loganathan et al. (1995) followed the same philosophy of the two-stage decomposition model of the LPG method, but employed two global-search schemes, MULTISTART and simulated ANNEALING, for changing the flows in the second stage (outer loop). The linear programming was used to optimise the pipe size in the first stage (inner loop). MULTISTART and simulated ANNEALING permit the search to migrate among various local optimal solutions. This approach yielded lower cost designs than previously reported designs in literature for two well-studied test problems of a two-loop network and the New York city tunnel network.

In summary, the methods of the two-stage decomposition model for the optimisation of design of water distribution systems are able to

(1) apply the well-developed LP technique to solving for the optimal length of pipe segments corresponding to the discrete commercially available pipe sizes;

(2) guarantee a local optimal solution;

(3) define the tight lower bound of global optimal solution;

(4) handle pumps and multiple loadings.

The methods, however, have difficulty performing well for a real world problem. This is due to the methods

(1) have difficulty considering other optimisation variables such as tank locations and sizes, and also rehabilitation actions (eg. cleaning), which are discrete in solution space and hard to cope with in the LP model;

(2) require considerable amount of mathematical sophistication, particularly for the computation of gradients of objective function and the tight lower bound of global optimal solution. These mathematical complexities may create some difficulty for the engineering community to accept the methodology;

(3) generate a rather high dimensionality of the optimisation model even for a network of moderate size (the dimension = the number of pipes x the number of available discrete pipe sizes);

(4) are not able to take advantage provided by the user-friendly simulation models that have been accepted as basic tools for network analysis by engineers.

Some of these drawbacks can be overcome by incorporating a hydraulic network solver into the optimisation of water distribution systems.

2.1.2 Optimisation methods using a network solver

Development of hydraulic network simulation models has provided some new opportunities for developing optimisation model for design of water distribution systems. A hydraulic network solver is usually used to calculate flows in pipes and the pressure heads at nodes while optimisation techniques are employed to optimise networks. A number of researchers have reported their works in optimisation of water distribution systems using a hydraulic network solver.

Morgan and Goulter (1985) adopted the two-stage decomposition formulation (Alperovits and Shamir 1977) and integrated the inner LP model with a Hardy Cross solver. This algorithm was able to analyse networks under multiple loads and determine the optimum layout and pipe sizes of looped water distribution systems. It is not applicable to large networks because the Hardy Cross solver is only effective and efficient at solving small network. Su et al. (1987) developed an optimisation model for design of water distribution systems. A nonlinear programming algorithm called the generalised reduced gradient (GRG) technique was integrated with a steady state simulation model KYPIPE (Wood 1980). The GRG was used to optimise the network while KYPIPE was employed to calculate hydraulic pressure heads throughout the system. A reliability constraint was added to the model and guaranteed to produce a looped network. The reliability was defined as the probability of the design pressure being maintained at appropriate nodes in the system for a given probability of some pipes being unavailable e.g. a pipe breakage. The model could not handle other system components such as pumps, valves and storage tanks. Lansey and Mays (1989a) used the same techniques as Su et al. (1987), namely

GRG and KYPIPE. In their model, an augmented Lagrangian method was used to handle the minimum head and other constraints. It was able to optimise not only pipe sizes but also the pumps and tanks under multiple loadings. The model often resulted in a branch network because no reliability restriction was included in the model. Duan et al. (1990) improved the work of Lansey and Mays (1989a). A separate model for computing system reliability was incorporated into the optimisation model integrating GRG and KYPIPE. Their formulation decomposed the optimisation procedure into two levels the master problem level and the subproblem level. At the master problem level, the number and locations of pumps and tanks are identified. At the subproblem level, GRG, KYPIPE and reliability model were employed to optimise the pipe sizes for the pump and tank locations specified at the master level.

The methods integrating a hydraulic network solver and a nonlinear optimisation technique provided more flexibility for considering some discrete variables than the methods that do not use the network solver. Kim and Mays (1994) proposed a mixed-integer nonlinear programming formulation for optimal rehabilitation of water distribution systems. In their model, two integer variables (each takes one value of either 0 or 1) were used for each pipe within the system for replacement and rehabilitation, respectively. The other variables such as pipe diameters and pump horsepower are continuous variables. The solution methodology included an implicit enumeration scheme for integer variables and an integration of the GRG and KYPIPE for optimising the continuous variables of pipe diameter and pump horsepower.

More recently, Taher and Labadie (1996) presented a water distribution system optimisation program (WADSOP), interfacing a simulation model and the LP optimisation technique with geographic information system (GIS). GIS provides an efficient spatial data management and analysis tool for preparation of accurate spatial information for input to network simulation and optimisation models, and also for post-optimisation analysis graphical output display. It also provides a graphical environment and functions for layout design of the networks. In their optimisation model, the optimisation of a water distribution system was formulated as a two-stage decomposition procedure as originally proposed by Alperovits and Shamir in 1977. A nonlinear programming algorithm, the Frank-Wolfe algorithm, was employed as the network solver for steady state simulation. The LP model determined the optimal pipe diameter and pump head for the flows and heads from the network solver. An iterative optimisation procedure continues until no significant change of objective cost occurs. As the model employed LP for optimising the network, it avoided the calculation of gradients of the design variables as required by the GRG method, and consequently reduced the computer CPU time. To enable the LP to be applied to solving the optimisation problem, however, some assumption, such as the pumping energy cost being a linear function of pumping heads, was made. This assumption may reduce the optimality of the final solution.

The methods developed by using hydraulic network solvers offset the drawbacks of the methods that do not use solvers as follows.

(1) The methods employ hydraulic simulation models as network solvers and nonlinear programming algorithms as optimisers. This reduces not only the mathematical complexity of optimisation formulation for design and rehabilitation of water distribution systems, but also ease the implementation of the optimisation model by integrating well-developed and off-shelf computer packages of network solvers and optimisation algorithms.

(2) The methods are usually more flexible than the methods that do not use a solver, and thus they enable more system components such as tanks and valves, and also

enable some discrete variables such as rehabilitation actions to be considered in the model.

These methods usually employ a non-linear programming algorithm for optimising continuous variables of the system components and enumerative scheme for optimising discrete variables of rehabilitation actions. The main drawbacks with such methods are:

(1) It only guarantees a local optimal solution because the optimisation is a non-convex nonlinear programming problem.

(2) It requires that the cost of the pipelines be expressed as continuous and preferably differentiable function of the pipe diameters. The available pipe sizes, in reality, are discrete. This means that a cost function needs to be fitted from cost information for a set of discrete pipe sizes.

(3) The final optimal pipe sizes are a set of continuous pipe diameters between the prespecified maximum and minimum diameter. Thus the optimal diameters need to be rounded up or down to an adjacent available discrete diameter size. This reduces the optimality of the solution and sometime may generate an infeasible solution.

(4) It lacks robustness and requires large computation times because of the gradient calculation.

Some other methods have been employed for optimisation of water distribution systems in literature. The optimisation algorithms include direct search techniques (Ormsbee and Contractor 1981), dynamic programming (Yang 1975; Kally 1971), integer programming (Rowell 1982; Oron and Karmeli 1979), and also enumerative methods (Gessler 1982). These optimisation models have been reviewed in detail by Lansey and Mays (1989b).

Traditional optimisation techniques, such as linear programming, non-linear programming and enumerative methods, have been investigated by many researchers in the past. It has been found (Murphy & Simpson 1992; Simpson et al. 1994) that genetic algorithms (GAs), a general search algorithm based on natural evolutionary principles and biological reproduction, outperform their counterparts for water distribution system optimisation.

2.2 Evolutionary Optimisation Techniques

Natural evolution has long been visualised as an optimisation process by Rechenberg (1973) and Schwefel (1981) in Germany and in a parallel development by Holland (1975) in the United States. The principle of natural selection and biological reproduction is simplified in both of approaches into two similar but different optimisation techniques, namely evolutionary strategy and genetic algorithm (GA). The genetic algorithm was successfully applied to optimal operation of a gas pipeline by Goldberg and Kou in 1987. Since then the GA has been applied to many optimisation problems in different areas (Goldberg 1989). Cembrowicz and Krauter (1977) applied evolutionary strategy along with graph theory and linear programming to solving the optimisation of the design of a sewer system. It was noted that the optimisation technique based on the evolutionary principles offered a very effective solution method. This evolutionary technique, and in particularly the genetic algorithm technique, has been further explored and applied to optimisation of water distribution system in literature. A brief overview of the GA is given in the following section.

2.2.1 Standard genetic algorithms: a brief overview

Genetic algorithms (GAs) (Holland 1975, Goldberg 1989) are fundamentally different from traditional optimisation approaches in terms of search process. Genetic algorithms, generally, initiate a population of points in the search space. Each point is represented by either a n-bit string or a set of real numbers encoding the information of the point (one solution of the problem). The string is usually called a *chromosome* and evaluated by a measure called *fitness* which quantifies the merit of the chromosome (one individual) adapting itself to the dynamic environment (solution space). In other words, the fitness quantifies the degree of the optimality of the solution. The parents are selected from the population by performing simulation of Darwin's *survival of the fittest* principle. These individuals reproduce their offspring by mimicking gene operations such as crossover and mutation, and thus the reproduction of a new generation is completed. After a number of generations, the population is expected to evolve artificially, and the optimum (hopefully the global optimum) will be found in the evolved population.

2.2.2 A standard GA based approach

In the early work on genetic algorithms applied to the optimisation of design of water distribution systems (Murphy & Simpson 1992), pipe diameters of new pipes and duplicated pipes were considered and encoded as a string. The GA was developed and successfully applied to a typical water supply network. The optimal solution, a set of diameters for new and duplicated pipes was found and compared with other methods. The GA based approach was able to find lower cost solutions for the problem. However, it is observed that the simple GA requires a large number of evaluation and is not able to reach the global optimal in every run.

Murphy et al. (1993) and Dandy et al. (1996) improved the simple GA with fitness scaling using exponentiation, adjacency or creeping mutation and Gray coding. The improved GA has been used to solve a well-studied problem for the New York City Tunnel water supply network. The optimal solution given by the improved GA was found to be better than that by any other method. However, it involved considerable effort to tune the

GA parameters, and dozens of runs to find the optimal solution. Simpson and Goldberg (1994) compared the effects on finding the known global optimal solution for a two reservoir problem for different coding schemes (binary coding, Gray coding and natural number coding), and different selection operators such as roulette selection and tournament selection, different crossover operators such as one point crossover, two-point crossover and uniform crossover, and also for different population sizes. It was found (Simpson and Goldberg 1994) that the selection scheme and population size were the most critical aspects of applying standard GAs to pipeline system design. Tournament selection was recommended. Savic and Walters (1997) proposed an approach by integrating the standard GA with a steady state hydraulic solver EPANET Rossman 1994). They identified that the optimal solution was sensitive to the Hazen William coefficient in the head loss equation.

A general methodology (Murphy, Dandy & Simpson 1994) was developed for applying the improved GA (Murphy et al. 1993) to optimise the *Anytown* water supply system (Walski et al. 1987), which has served as a benchmark problem for testing pipe network optimisation algorithms. The *Anytown* problem considered the expansion of the existing water supply network of an old central city area to meet water demands of several new residential and industrial areas. The expansion covers almost all the system components such as pipes, tanks and pumps. The total cost is the sum of the pipe costs (a function of the pipe lengths and diameters), the tank costs (a function of elevations and volumes) and the pump cost (a function of capacity). An improved design, in terms of cost, was found using the improved GA, but it was observed to be difficult to compare to the previous design because of different network reliability criteria and interpretations. This suggests that a robust GA-based approach to a comprehensive optimisation design for any water distribution system, particularly when all the components must be considered in the design, is still needed to be developed.

2.2.3 Lessons from the GA based approach

The application of GAs in water supply system optimisation has shown that the GA is a promising technique for the development of a practical design tool for water resource engineers. The GA based approach for optimisation of water distribution systems belongs to the category of the methods using the network solver. In addition to the advantages mentioned earlier in this Section 2.1.2, The GA optimisation of water distribution systems has been proven to be more effective at solving this type of problems than the traditional methods, particularly in the following aspects:

1. The GA works on either a discrete or continuous space.
2. The GA produces a set of near-optimal solutions instead of just one optimal solution by the traditional methods.
3. The GA has a global search mechanism. It increases the chance to reach the global optimal solution.
4. The GA is more robust than the traditional methods.

The GA that uses either a binary string coding or other alphabet string coding works on a discrete space. It enables more discrete variables of water system components, such as tanks and rehabilitation actions, to be included in the optimisation model without losing the generality of the water distribution system optimisation. The GA searches the solution space by using a population of genotypes. Each of them represents one alternative in the search space, consequently, it produces a set of final near-optimal solutions instead of just one final optimal solution by the traditional methods. The GA optimisation is guided by using the information of the objective function only. It avoids the calculation of the gradient of objective function with respect to the decision variables. Thus the GA is more robust than traditional methods. It increases chances of finding the optima solution by mimicking the principles of natural evolution and biological reproduction. In previous applications of the standard GA to optimisation of water distribution systems, however, the following drawbacks have being noted.

1. It is sometimes "hard" for standard GAs to find the global optimum design for a practical water distribution system (although, in general, it can never be guaranteed that the global optimum solution has been determined).
2. The standard GA requires a large number of objective evaluations to reach the optimal or near-optimal solutions.

Different GA runs may reach different solutions with different convergence rates. It is particularly difficult for the GA to reach the same optimal solution in every GA run as the dimension of the problem increases. Genetic algorithm optimisation, in theory, lacks convergence (optimality) criteria, thus an *exhaustive* search (a computer run using computation resources as much as possible) is usually performed to increase the probabilities for the GA to achieve optimal solution. Furthermore the standard GA is not able to solve a *linkage* problem where the genes coding the optimal solution are far apart instead of being coded tightly within one chromosome. This is often found in a high dimensional optimisation problem. Thus a more efficient GA technique such as a messy genetic algorithm (mGA) needs to be investigated.

2.2.4 Motivation for using messy genetic algorithms

A messy genetic algorithm, developed by Goldberg, Korb & Deb (1989), was seen as particularly suitable for problems which are "hard" to solve with standard GAs. It was designed to solve bounded difficult search problems by developing a tight linkage between artificial genes that improve the chances of leading to the global optimal. It is the tight linkage of the genes that help the messy GA to solve the linkage problem more efficiently and more effectively. In this research, a messy GA is employed to develop an approach for optimisation of water distribution systems. This methodology hopefully will be able to improve the existing GA based optimisation for water distribution systems and develop a sound foundation and provide insight into how the messy GA based approach can be developed for a comprehensive approach to the optimisation design for water distribution networks.

Finally, a pipe system design, in reality, includes not only sizing diameters but also selecting pipe wall thickness (pipe class) and surge or water hammer protection devices such as surge tanks. Consequently, the solution domain may become "hard" for standard GAs to search for the optimal or near-optimal solutions. This also encourages a more efficient and effective GA — the messy GA to be employed for the optimisation of water distribution systems including the consideration of water hammer loadings. To enable the pipe wall thickness and surge devices to be included in the optimisation formulation, a computer model simulating water hammer loadings is used to calculate the maximum and/or the minimum transient pressure heads for each of the links in water distribution systems.

2.3 Simulation Models

Computer modelling of water distribution system has been a basic tool for hydraulic analysis, design and operation of those systems. In practical applications, the simulation models are usually classified as steady state model and water hammer model. The steady state model is used to simulate flow condition which does not vary with time, namely instantaneous simulation or is governed by a temporary change of its boundary conditions, namely extended period simulation. Water hammer model takes into account the inertia of the water and the changes of the water velocity in the pipeline. It is employed to analyse the unsteady state flow condition generated by sudden change of water velocity within the system. Both the steady state model and the water hammer model are employed to develop a comprehensive methodology for optimisation of water distribution system in this research.

2.3.1 Steady state model

Steady state flow in pipeline networks is governed by the energy conservation law and mass balance law. The distribution of flows through the network under a certain demand pattern is given by solving the energy conservation equations namely loop equations and hydraulic head loss equations, and also mass conservation equations namely continuity equations at nodes. Different methods have been developed for solving the network equations in literature. They include Hardy Cross method (Cross 1936); Newton-Raphson techniques (Warga 1954); linear theory method (Wood and Charles 1972), Gradient method (Todini and Pilati 1987, 1988) and optimisation techniques (Collins et al. 1987). All the methods are theoretically capable of solving the set of system equations and were compared by Lansey and Mays (1989c). Hardy-Cross method is attractive for hand calculation and easily coded, however, it requires more computation time than the other methods, and converges to the solution slowly for complex networks. The Newton-Raphson method converges more quickly than the linear method for small systems. It requires less storage but converges more slowly than the linear method for large networks. The optimisation technique was introduced to solve the network equations by Collins et al. (1978). Three techniques including the Franke-Wolfe algorithm, piece-wise linear programming and convex Simplex algorithm were compared for solving the problem. It was found that the piece-wise approximation technique produced the most accurate solution for the least effort.

Todini and Pilati (1987) developed an efficient algorithm which took advantage of the special structure of the pipe network governing equations. The basis for the technique is that the solution to be found is the complete set of unknown heads and unknown flows, for which a unique solution is known to exist. This technique was implemented in a hydraulic network solver EPANET by Rossman (1994). Main advantages of the method are as follows (Rossman 1994).

1. The system equations to be solved at each iteration of the algorithm is sparse, symmetric, and positive-definite. This allows highly efficient sparse matrix techniques to be used for their solution.
2. The method maintains flow continuity at all nodes after its first iteration.
3. It can readily handle pumps and valves without having to change the structure of the equation matrix when the status of these components changes.

The EPANET is a public domain software and will be employed in this research.

2.3.2 Transient model

When a steady state flow condition in pipeline networks is disturbed by sudden change the flow velocity a unsteady state flow occurs. The flow situation before the unsteady state flow becomes steady again is known as a transient flow (Wylie and Streeter 1993). The transient flow is governed by two differential equations. One of them is unsteady continuity equation describing the mass balance of the flow within the system. The other is called unsteady momentum equation describing the balance between the friction force and the gravity force, caused by a rapid change of boundary conditions in the pipeline network. Both of the equations are classified as quasi-linear hyperbolic partial differential equations. Because of the nonlinearity of the equations along with nonlinearities introduced by boundary conditions e.g. surge tanks and air chambers, no analytic solutions of the transient flow are available. Only an approximate solution can be obtained by numerical methods. The methods that have been used to solve the partial differential equations are method of characteristics, finite difference method and finite element method.

The finite element method is very well developed. It can accurately represent a complex boundary and handle high-dimensional nonlinearity of a physical domain, but requires more computation than other two methods. Since the transients in a pipeline are usually represented as one-dimensional waves it is highly unlike that the finite element method can perform better than other methods.

There are two types of finite difference methods namely explicit difference methods and implicit difference methods. The explicit finite difference schemes usually have significant restrictions on the maximum time step to maintain the solution stability. A number of implicit finite difference schemes have been applied to solving the transient equations. The partial derivatives are approximated by the finite difference scheme, the unknowns of the problem are solved simultaneously at each time step. Although the simultaneous resolution of the unknowns increases the computation effort, efficiency of the transient simulation may be improved by using much larger time step because Courant number (wave propagation velocity times the ratio of the time step to the space step) is allowed to be greater than the unity. Arfaie and Anderson (1990) found that the centred difference scheme produced the most accurate results, but it generated other errors, in particular numerical phase errors associated with wave dispersion when the Courant number is small. The phase errors can be avoided by dividing the scheme over more points in space, but it results in a lower accuracy at the boundaries. A space-compact high-accuracy scheme was proposed by Verwey and Yu (1993). This scheme is defined between two points in space and over three levels in time. The advantages of the scheme in practical implementation are (1) the distance steps of the computation grid can be varied along pipeline without affecting the accuracy of the adjacent pipe sections and (2) the Courant number in different pipes or pipe sections can be varied while the high accuracy is maintained. This implicit finite difference scheme, however, needs to be further developed for the different boundary conditions, such as pumps, valves, surge tanks and air chambers, which are often found in water distribution systems.

The method of characteristics (MOC) exploits the hyperbolic nature of the governing equations to convert the two partial differential equations into four ordinary differential equations. The integration of the four ordinary differential equations gives rise to a set of explicit equations for solving for the discharge and head of internal sections of a pipe. This explicit solution equation offers a distinct advantage of the MOC being easily implemented. A number of boundary conditions of the MOC have been developed by Wylie and Streeter (1993), One boundary condition can be solved independently from the others. The accuracy of the method has also been verified by physical experiment results (Chaudhry 1987; Wylie and Streeter 1993). Finally, the boundary conditions of the MOC for water distribution systems have been further simplified and developed by Karney and McInnis (1992). An efficient and comprehensive representation of pressure surge protection devices and valves has been introduced and verified by applying it to a real world water system. It improves the efficiency of the MOC for transient analysis in water distribution systems. In this research, this approach will be used to develop a transient model that will be integrated into optimisation model for optimal design of water distribution system under water hammer loadings.

2.4 Scope of This Research

The review of optimisation of water distribution systems as given in Section 2.1 and 2.2 indicates that no optimisation technique, up to date, is efficient and effective at comprehensively solving the optimisation problem. This has prompted many researchers to develop new methods and apply state-of-the-art optimisation techniques to solving the problem. In this research, the messy genetic algorithm—a new generation of genetic algorithms will be applied to optimisation of water distribution systems. The efficiency and effectiveness of GA optimisation of water distribution system will be further improved by developing a technique particularly for solving a constrained nonlinear optimisation problem where the optimal solution is located at the boundary of the feasible and infeasible regions in the search space.

As indicated in Section 2.2.4, the formulation of optimal design of the water distribution system needs to be extended to take into account sizing pipe wall thickness and pressure surge protection devices. Water hammer can damage the pipe in short term by generating dangerously high pressures that often cause pipe bursts, and it also may cause column separation. Current water hammer design and selection of water hammer control devices for pipeline systems are based on a trial and error procedure. The transient analysis is often referred to a specialist because of the sophistication of the transient analysis for the water distribution systems. It is highly unlikely that a comprehensive analysis of water hammer events in a water pipeline network is carried out to maintain the safety of pressure surge protection and at the same time to achieve a least cost design of the system. Development of optimisation model to include water hammer, to the best of my knowledge, will be the first time to provide engineer with a methodology for completely sizing the pipeline network such that the total cost of the network is minimised subject to satisfying water supply demands and water hammer protection requirements. The specific tasks of this research are:

1. Application of the original messy GA using a complete enumerative initialisation to optimisation of water distribution systems just under steady state loadings.

2. Development and application of a fast messy GA to optimisation of design and rehabilitation of water distribution systems.
3. Development of a GA technique searching the boundary of the feasible and infeasible regions of the search space. This will be further improving the efficiency and effectiveness of optimisation of water distribution systems.
4. Development of a transient model for simulation of water hammer in water distribution systems.
5. Development of a generalised methodology for optimal design of water distribution system. It will be able to size not only pipe diameters but also pipe wall thicknesses (pipe classes) and surge (water hammer) protection devices.

The work was initiated by investigating the application of the original messy GA to optimisation of water distribution systems. This is discussed in the next Chapter.

3. MESSY GENETIC ALGORITHM FOR OPTIMAL DESIGN OF WATER DISTRIBUTION SYSTEMS

3.1 Problem Description

An overall design of a water distribution system involves selection of the network layout including the pipeline routes, the locations of many components such as pump stations, valves and tanks. In addition, other aspects that need to be selected include the sizes of all the components including water hammer control devices and pipe wall thicknesses. The design is subject to constraints such as geometric limitations and water availability. To date it appears that no optimisation technique can handle the overall optimisation design problem in one model. In fact, practicing engineers subdivide the problem into several different stages and solve each of them as a separate stage, instead of attempting to solve the problem in its entirety. The planning and design process for water distribution systems has been classified into four stages (Walski 1995) including (1) master planning; (2) preliminary engineering design; (3) subdivision design and (4) rehabilitation. Most of the research up to date in the area of optimisation of water distribution systems has fallen into later two categories. Examples include sizing of new pipes added to an existing network and/or the pipes replacing or duplicating existing pipes; and choosing rehabilitation actions such as cleaning or lining of old pipes to improve system supply performance.

This type of optimisation design problem has been well studied and specified in the following way (Alperovits and Shamir 1977; Gessler 1985; Walski et al. 1987; Dandy, Simpson & Murphy 1992; Simpson, Dandy & Murphy 1994). For a given layout of a network and prescribed water demands at certain nodes, the total cost of full system components (mainly pipes) is minimised by searching for the optimum combination of pipes sizes and rehabilitation actions. The search process is subjected to two types of constraints. The first type involves component dimension limitations such as minimum and maximum pipe diameters. The second type of constraint includes hydraulic requirements such as minimum pressure heads at certain nodes. The hydraulic analysis is generally carried out by pipe flow simulation models.

3.2 Discrete Model Formulation

There are two types of models for optimal design of water distribution systems. These include a discrete model and a continuous model. The continuous model is based on fitting curves to cost functions. A pipe diameter d_n is treated as a continuous variable between a minimum and maximum diameter in a continuous optimisation model. Commercially available pipe sizes, however, are discrete. For a gradient based optimisation approach, a formulation with continuous diameters is used and a cost curve (function) is usually fitted by using the discrete pipe sizes and unit cost data. Most of the continuous formulation approaches (Lansey and Mays 1989; Fujiwara & Khang 1990; Ahn 1993; Loganathan, Greene & Ahn 1995) up to date have adopted a fitted cost function in the search process. An optimisation model using discrete pipe sizes is formulated as follows.

The task is to select pipe diameters $\vec{D}=\{d_n, n=1,..., N\}$ for all new pipes added to a water system, and to choose rehabilitation actions (e.g. duplicating or replacing a pipe or cleaning a pipe) and the associated pipe sizes $\vec{E}=\{e_r, d_r, r=1,...R\}$ for all rehabilitated pipes. The objective is to ensure that the total cost of pipe materials and rehabilitation actions $C_{total} = (\vec{D}, \vec{E})$ is minimised subject to minimum allowable hydraulic pressure heads at nodes and pipe size limits.

In this model, all pipe diameters d_n, $n = 1, \ldots, N$, where N is the number of new pipes, take values directly from commercially available discrete sizes. The cost of pipes is assumed to be only a function of its length. Therefore, the cost of all new pipes may be written as:

$$C_{new} = \sum_{n=1}^{N} c_n(d_n) L_n \tag{3.1}$$

where L_n = the length of new pipe n; d_n = the diameter of new pipe n; $c_n(d_n)$ = the cost of per unit length of new pipe n with diameter d_n.

The cost of a rehabilitated pipe (e.g. removing, replacing, duplicating or cleaning of a pipe) is a function of the rehabilitation event and associated diameter (e_r, d_r), where r = $1, 2, 3, \ldots, R$. Thus the cost function of the rehabilitation actions is given as:

$$C_{reh} = \sum_{r=1}^{R} c_r(d_r, e_r) L_r \tag{3.2}$$

where R = number of rehabilitated pipes; L_r = length of rehabilitated pipe r; d_r = diameter of rehabilitated pipe r; e_r = rehabilitation action taking place at pipe r; $c_r(d_r, e_r)$ = cost of unit length of rehabilitated pipe r with diameter d_r and event e_r.

The total cost of the water system is:

$$C_{total} = C_{new} + C_{reh} \tag{3.3}$$

or more generally :

$$C_{total} = C(\vec{D}, \vec{E}) \tag{3.4}$$

A discrete optimisation model may be written as:

search for $(\vec{D},\vec{E})$ such that

$$\textit{minimise} \qquad C(\vec{D},\vec{E}) = \sum_{n=1}^{N} c_n(d_n)L_n + \sum_{r=1}^{R} c_r(d_r,e_r)L_r \qquad (3.5)$$

subject to

$$\forall d_n, d_r \in D^0 = \{d_k^0, k = 1, \ldots kk\}$$
$$\forall e_r \in E^0 = \{e_h^0, h = 1, \ldots, hh\}$$
$$H_{i,j} \geq H_j^{min}, j = 1, \ldots\ldots, J; i = 1, \ldots\ldots, I$$

where d_k^0 = k-th commercially available pipe diameter; kk = number of the commercially available pipe sizes; hh = number of the rehabilitation events applicable to pipes; e_h^0 = h-th rehabilitation event that may applied to the existing pipe r; $H_{i,j}$ = hydraulic grade at node j for steady state loading I; I = number of steady state loadings; $H_j^{\min}$ = minimum requirement of hydraulic grade at node j and J = number of nodes in system (excluding fixed grade nodes).

```
Each solution {D1,D2,...,Dn} from GA initialisation;
Update the values of decision variables;
Call hydraulic simulation model (e.g. EPANET);
Evaluate the cost of the solution;
Generation = 1;
While (Generation < specified maximum generation)
     { /*performing GA search*/
     Each solution {D1,D2,...,Dn} from optimisation process;
     Update the values of the decision variables ;
     Call hydraulic simulation model (e.g. EPANET);
     Evaluate the cost of the solution;
     Generation = Generation + 1;
     }
```

Figure 3-1 Pseudo-code for Optimal Rehabilitation and Pipe Sizing

3.3 A General Optimisation Procedure

In order to find low cost solutions, a hydraulic solver EPANET (Rossman 1994) is employed and coupled with a messy GA (mGA) search algorithm. EPANET is used to simulate the water flow in the pipe network system and gives hydraulic pressure head (grade) at nodes. The mGA is used to search for optimal pipe sizes and the best rehabilitation policy. A conceptual procedure of the approach is illustrated in Figure 3-1.

3.4 Why Messy Genetic Algorithms?

As genetic algorithms are receiving more and more attention in combinatorial optimisation problems, it has been found that standard GAs, with fixed-length strings and various genetic operators, are very unlikely to solve a difficult optimisation problem. A difficult search problem is a problem where a great number of near-global optimal points are present in a search space. The standard GA search process favours a suboptimal solution instead of the global optimal solution when searching a difficult problem. This type of problem is usually a so-called *deceptive problem* (Goldberg 1987). Simply stated, *deception* (for GAs) means that low-order building blocks lead the search away from the global optimum. The optimal genetic features (genetic configuration or building blocks) for this type of problem are not coded tightly within the chromosome, instead they are located far apart. This is referred to as *loose linkage* or a so-called *linkage problem* for genetic search approach. It has been recognised as one of the biggest challenges for GA based applications, especially for highly dimensional problems such as optimisation of a large-scale water distribution network.

A standard GA is not able to deal with the *linkage problem*. The standard GA considers only a small fraction of building blocks defined by the representation. The GA with one-point crossover favours those building blocks for which positions in sequence space are closer to each other and neglect those building blocks that contain positions far apart. The *inversion* operator was suggested as a possible solution to this problem, but it is very slow and unlikely to solve the problem efficiently (Thierens & Goldberg 1993). Uniform crossover does not have any preference bias toward the closely spaced partitions. Random mutation of the strings also results in disrupting proper evaluation of the building blocks. The standard GA is not able to accomplish a proper search in the building block space if the problem is deceptive.

A new genetic algorithm, called messy genetic algorithm (mGA)) (Goldberg, Korb & Deb 1989) was purposely designed and developed to solve search problems of bounded difficulty. A carefully designed 30-bit deceptive problem has been solved to global optimality by using a mGA without any prior knowledge of the structure of the problem. Furthermore, the mGA has been shown (Goldberg, Deb & Korb 1990) to converge to the global optima in a time that grows as a polynomial function of the number of decision variables on a serial machine. Although the optimisation for water distribution systems has not been theoretically shown to be a *deceptive problem*, carefully tuning of the GA and a large number computer runs have been required to search for the optimal or near-optimal solutions (Murphy & Simpson 1992; Murphy et al. 1993). It suggests that optimal sizing and rehabilitation of water distribution system may be a difficult task for GAs. The outstanding performance of the mGA on the carefully designed deceptive problem by Goldberg et al. (1989) strongly suggests that it be able to solve bounded problems that a standard GA is not able to solve.

An approach called the structured messy genetic algorithm (SMGA) (Halhal et al. 1997) was developed and applied to the optimal improvement of water distribution systems.

The SMGA employed fixed-length strings in one generation. The length of the strings increases with a fixed step size over generations, thus the chromosomes in each generation have the same length or a "tidy" structure, which enables a standard GA crossover to be applied in a GA process. However, the messy genetic algorithm (mGA) (Goldberg, Korb & Deb 1989) was originally developed by using variable-length strings, a threshold genetic selection and messy operators of *cut* and *splice* to solve search problems of bounded difficulty. The length of strings can be varied not only over generations but also within each generation. The messy GA provides a flexible coding representation which enables a linkage problem to be solved. Thus it is expected to be able to solve a combinatorial optimisation problem such as optimal sizing and rehabilitation of a water supply network more easily with a mGA than a standard GA. If this is the case, it will make it more practical to employ genetic search methodology in the development of a comprehensive methodology for the optimisation design and rehabilitation of water distribution systems.

3.5 Genotype Representation and Fitness

In order for a GA to be used to optimise the pipe sizes and rehabilitation actions for a network, variables such as the pipe size (diameter) for each new pipe, the rehabilitation action for an existing pipe and its size associated with the action must be coded into a n-bit string, usually a binary string. The string is reproduced in the GA process by mimicking genetic reproduction rules, and thus is called a genotypical representation or a genotype of the optimisation variables. A genotype or one string corresponds to a set of values of the variables, namely one solution of the problem. This solution is the so-called phenotype of the string.

In a canonical GA, a population is followed over a time t (generations) by its genotypical representation as a sequence $G(t) = (\vec{g}_1(t), \dots \vec{g}_{nn}(t), \dots, \vec{g}_{NN}(t))$, where NN is the population size, composed of a population of elements $\vec{g}_{nn}(t)$ called chromosomes. Each chromosome is represented by a string of binary digits. The total length of the chromosome is divided into N intervals of the same length dd for a new pipe and R intervals of the same length pp for a rehabilitated pipe. Thus each genotype $\vec{g}_{nn}(t)$ has a general form as:

$$\vec{g}_{nn}(t) = (a^d_{1,1}, \dots a^d_{1,dd}, \dots a^d_{N,1}, \dots, a^d_{N,dd}, a^e_{N+1,1}, \dots, a^e_{N+1,ee}, a^d_{N+1,ee+1} \dots, a^d_{N+1,ee+dd}$$

$$a^e_{N+R,1}, \dots, a^e_{N+R,ee}, a^d_{N+R,ee+1} \dots, a^d_{N+R,ee+dd}) \qquad (3.6)$$

$a^d_{n,dex}$ is a gene bit, taking the value either 1 or 0 for a binary string, where n = 1, ..., $N+R$, N is the number of new pipes to be added to a distribution system; R is the number of existing pipes that may be rehabilitated; dex = 1, ... dd, dd is the number of bits representing the diameter of an expanded or rehabilitated pipe. Similarly, $a^e_{r,edex}$ represents a rehabilitation event, where r = $N+1$, ..., $N+R$, $edex$ = 1, ..., ee, where ee is the number of bits representing a rehabilitation action for an existing pipe. The number pp of bits coding a pipe undergoing rehabilitation is the sum of the bits for coding the new pipe diameters and the bits for coding the rehabilitation actions namely

$$pp = dd + ee \qquad (3.7)$$

Each chromosome, the genotypical individual $\vec{g}_{nn}$ with the length of ($dd*N$ + $pp*R$) bits, defines a corresponding phenotypical representation or a phenotype namely one design alternative given as:

$$\vec{p}_{nn} = (\vec{D}, \vec{E}) = (d_1, \dots, d_N, e_1, d_{N+1}, \dots, e_R, d_{N+R}) \qquad (3.8)$$

One example of a genotype and a phenotype representation is given as follows. Suppose that in a water distribution system, there are 5 new pipes (N = 5) which are to be added to the system, 3 existing pipes (R = 3) which are to be rehabilitated by taking one of the actions such as cleaning, duplicating or leaving, and that there are 8 commercially available pipe sizes which can be used for new and duplicated pipes. Three binary bits (dd = 3) are needed to represent pipe size alternatives for each new or duplicated pipe ($2^3 = 8$ choices), 2 binary bits (ee = 2) are needed to represent rehabilitation action for each of 3 choices for actions associated with existing pipes (this actually provides 4 choices, 2^2, however, 2 genotype values can be set to the same rehabilitation action). Thus the total length of a genotype for this example problem is 30 bits. Figure 3-2 shows one genotype and phenotype of this example network.

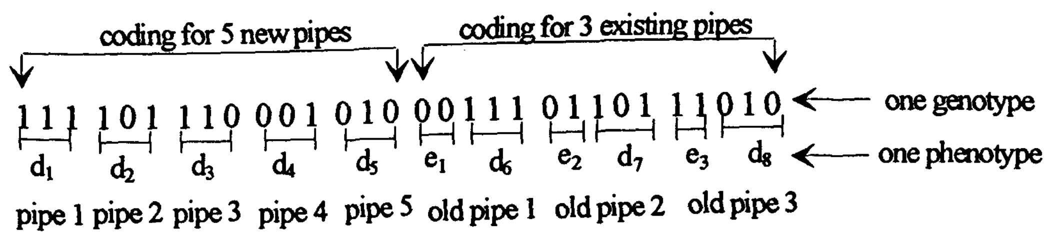


Figure 3-2 An Example of a Genotype Coding Scheme

One phenotype, namely one alternative design configuration, is composed of a sequence of diameters for new pipes, and diameters associated with a rehabilitation policy for duplicated or replaced pipes. Each such alternative gives rise to a cost for new pipes and rehabilitated pipes. The fitness can be defined for a genotype representation as (Wu 1994) :

$$\Phi_{nn} = 1 - \frac{C_{nn}(\vec{D}_t, \vec{E}_t)}{\underset{nn=1,\ldots,NN}{Max} C_{nn}(\vec{D}_t, \vec{E}_t)} \tag{3.9}$$

where NN = the population size. This has a desirable property that the fitness is in the range $0 \le \Phi_{nn} \le 1.0$ and that the cost $C(\vec{D}, \vec{E})$ will be minimised over generations t.

A methodology is needed for transcribing the genotypical representation of a chromosome, a n-bit string, into the phenotypical representation of pipe diameters and rehabilitation policies (the rehabilitation actions allocated to existing pipes). Generally, this is done by using a mapping from a genotype to phenotype of M: $\vec{g}_{nn}(t) \mapsto \vec{p}_{nn}(t)$. The mapping function can be defined in many ways, but the following definitions are given for the discrete pipe model.

3.6 Mapping From Genotype to Phenotype

For the discrete optimisation formulation, a pipe size (for a new or duplicated or replaced pipe) d_n takes the value from a commercially available diameter pipe list of kk+1 choices as $D^0 = (d_0^0, d_1^0, \ldots d_k^0, \ldots, d_{kk}^0)$, and a rehabilitation action is selected from an action set of hh+1 options as $E^0 = (e_0^0, e_1^0, \ldots, e_h^0, \ldots, e_{hh}^0)$. The unit cost (the cost per unit length) for rehabilitating an existing pipe is associated with a specific rehabilitation event (such as cleaning a pipe) and its corresponding diameter. Thus a unit cost matrix $c = (c_{h,k})_{(hh+1)\times(kk+1)}$, where h=0,1,2,...hh and k=0,1,2,...,kk, is generally formed as shown in Table 3-1. Each $c_{0,k}$, k =0, 1, 2, ..., kk, represents the unit cost of a new pipe with a diameter of d_k^0, while $c_{h,k}$ (h = 0,1, ..., hh; k = 0,1, ..., kk) represents the unit cost of a rehabilitation event e_h^0 with diameter of d_k^0.

One genotypical segment $(a^d_{n,1}, a^d_{n,2} \ldots a^d_{n,dd})$, where *dd* is the number of bits for coding a pipe diameter, can be mapped into its phenotypical representation d_n, $n=1,\ldots,N+R$, a pipe diameter as follows:

$$index = \sum_{dex=1}^{dd} a^d_{n,dex} b^{dex-1} \tag{3.10}$$

$$d_n = \begin{cases} d^0_{kk}; & for \quad index \geq kk; \\ d^0_{index}; & for \quad index < kk; \end{cases} \tag{3.11}$$

Table 3-1 The Cost per Unit Length for a New Pipe in a Network Expansion or Rehabilitation

Pipe Diameter list	d^0_0	d^0_1	……	d^0_k	……	d^0_{kk}
New pipe added (e^0_0)	$C_{0,0}$	$C_{0,1}$	……	$C_{0,k}$	……	$C_{0,kk}$
Leave the pipe(e^0_1)	$C_{1,0}$	$C_{1,1}$	……	$C_{1,k}$	……	$C_{1,kk}$
Cleaning a pipe (e^0_2)	$C_{2,0}$	$C_{2,1}$	……	$C_{2,k}$	……	$C_{2,kk}$
……	……	……	……	……	……	……
…… (e^0_h)	$C_{h,0}$	$C_{h,1}$	……	$C_{h,k}$	……	$C_{h,kk}$
……	……	……	……	……	……	……
…… (e^0_{hh})	$C_{hh,0}$	$C_{hh,1}$	……	$C_{hh,k}$	……	$C_{hh,kk}$

Similarly, genotypical representation $(a^e_{r,1}, a^e_{r,2} \ldots a^e_{r,ee})$, where $r = N+1, \ldots, N+R$, coding a rehabilitation policy for an existing pipe, is mapped into a phenotypical representation e_r, a rehabilitation action as follows:

$$index = \sum_{edex=1}^{ee} a^e_{n,edex} b^{edex-1} \tag{3.12}$$

$$e_r = \begin{cases} e^0_{hh}; & for \quad index \geq hh; \\ e^0_{index}; & for \quad index < hh; \end{cases} \tag{3.13}$$

where $b = 2$ for binary strings and $b = 10$ for natural number strings.

For instance, two sub-strings 101 for a new pipe n and 11 011 for an existing pipe r to be rehabilitated for a coding example in Table 3-2 can be mapped by using the index given by the equations from (3.10) to (3.13) as shown in Table 3-2.

Table 3-2 An Example of Mapping from Genotype to Phenotype.

	new pipe	existing pipe	
variables	diameter(d_n)	action(e_r)	diameter(d_r)
genotype (substring)	101	11	011
index by Eq. (3.10) or (3.12)	5	3	3
phenotype (solution)	d^0_5	e^0_3	d^0_3

3.7 Messy GA Components

What is a messy GA (mGA)? A messy genetic algorithm uses variable-length strings, a threshold genetic selection (also called genic selective crowding) and messy operators of cut and splice (Goldberg et al. 1989). The components have been designed based on an analysis of the GA schema theorem, which guarantees that all the *building blocks* —short, highly fit combinations of bits are growing and can combine to give improving solutions. The basic components of the mGA as applied to optimisation pipe-sizing and rehabilitation as part of this Ph.D research in this work are outlined as follows.

3.7.1 Messy genotype

The mGA uses variable-length strings in the genetic based search. Each bit of a string consists of a bit value, 0 or 1 for binary coding, and a name or a tag of the bit. A bit name or a tag is usually the order of the bit in a complete string. The variable-length string gives rise to an under- and/or over-specification of the genotypes. For example, a 3-bit coding can be represented either by (1,1), (3,0) or by (1,1), (2,1), (3,0), (3,1), where the first number within the bracket refers to the tag and the second number refers to the value of the bit i.e. zero or one. The former coding is called underspecification because the bit with tag 2 is missing. The latter is called overspecification because of more than one value for the bit with tag 3. To evaluate the underspecified string, Goldberg, Korb & Deb (1989) suggested that the best way to fill in the missing gene locus is to use a *competitive template*. The competitive template can be a random solution initially and then is updated with the best solution found in each generation. For the overspecified strings, the redundant bits are removed by following a *first-come-first served* rule scanning from left to right. In this way, the messy GA is able to evaluate the variable-length strings. It is this length-relaxed genetic representation that makes it possible for weakly linked building blocks, which cause a *linkage problem,* to be sought out and to be ordered to assemble a genetic structure that is more likely to reach the global optimum.

3.7.2 Thresholding selection

A specifically designed selection scheme—thresholding selection is required for comparing the variable-length genotypes in the mGA. In a standard GA, the individuals have very 'tidy' structure—i.e. each individual has the same number of bits. In contrast, individuals from a mGA have 'untidy' structures—i.e. individuals have different numbers of bits. For example, for a 5-bit problem, the strings ((1,1) (2,0)) and ((3,1) (5,0) (4,1)) can be selected, but comparing both strings is meaningless because there are no bits specified for matching tags. Comparing strings makes sense only when they have some genes in common, in other words, only comparing the same classes of building blocks is meaningful. Thresholding selection (Goldberg, Deb & Korb 1990) was introduced to ensure that the strings compete with each other only when they contain some genes from the same locus. A similarity measure θ is used to denote the number of common genes in the two strings. Two strings are allowed to compete with each other if the θ is greater than a threshold value given as (Goldberg et al. 1990):

$$\theta = \left\lceil \frac{l_1 l_2}{l} \right\rceil \tag{3.14}$$

where the operator $\lceil\ \rceil$ denotes a ceiling operator that calculates the nearest integer greater than the operand; l_1 = the length of the first string, l_2 = the length of the second string and l = the problem length. For example, for a 10-bit problem $l = 10$, the string ((1,0) (2,0) (4,1)) ($l_1 = 3$) and the string (((1,1) (3,0) (5,0) (6,1)) ($l_2 = 4$) can be selected, and a threshold value $\theta = \lceil 12/10 \rceil = 2$ is required by Eq. (3.14), but the number of the common gene in these two strings is 1, thus they are not allowed to compete each other by the thresholding selection.

In practice, a tournament selection is held, except that genotypes are forced to compete with those with some genes in common with them. Thresholding selection (Goldberg, Deb & Korb 1990) starts by picking up the first random candidate from a permutation list, and then the second candidate is chosen by checking the next *shuffle number*, n_{sh}, candidates in the list. A conceptual procedure of thresholding selection is shown in Figure 3-3.

```
{
choose a shuffle number n_sh > 2;
create a random permutation list including all the genotypes;
while  (selection is required)
     {
          pick the first candidate randomly from the list
          without replacement;
          check the next n_sh  candidates, one at a time;
          if(the candidate with at least θ genes in common
          with the first is found)
          then(compare both, select the better one)
          else(the first one is selected);
     } /*end of selection */
}
```

Figure 3-3 Pseudo-code for Thresholding Selection

3.7.3 Messy operators

The main genetic operator of crossover used in the standard GA cannot be used for the mGA due to overspecified and/or underspecified strings. Two messy operators, *cut* and *splice*, were designed by Goldberg et al. (1989) and are used for messy genetic reproduction instead of crossover. *Cut* acts to cut a chromosome into two, while *splice* links or concatenates two chromosomes to form one individual. The cut operator is activated by the *cut* probability as

$$P_C = P_k(\lambda - 1) \qquad (3.15)$$

where P_k is specified bitwise cut probability, and λ is the length of the string. Splicing is initiated by a prescribed probability P_S. If *cut* and *splice* are called in turn and applied to two strings, both operators work in a similar way to one point crossover operator in traditional or standard GAs. If two strings are cut, and then just the splice operation is called for the second string, it works like an inversion operator. Figure 3-4 shows *cut* and *splice* operations.

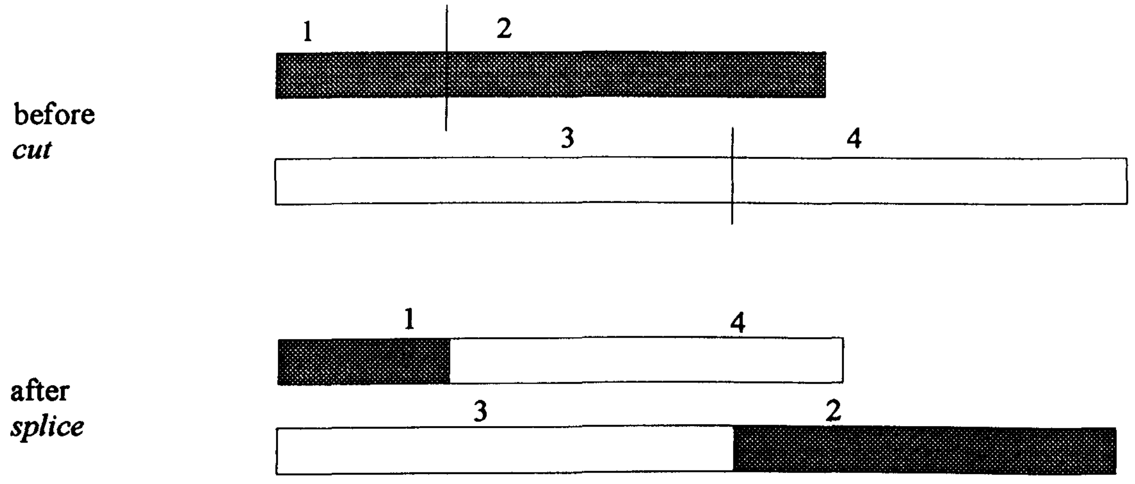


Figure 3-4 ***Cut*** **and** ***Splice*** **Operations of the Messy GA (*Goldberg et al. 1989)***

There are two phases to the mGA including the *primordial phase* and the *juxtapositional phase.* As the string length grows, the *cut* probability P_C increases due to the dependence on string length as shown in Eq. (3.15). Consequently, disruption of the genetic expression becomes highly likely. It means that the probability of forming a complete optimal genetic representation is reduced as string length grows longer and longer. One straightforward idea is to form the optimal genetic expression before the genotype grows so long that the cut operation will disrupt the genetic structure. The *primordial* phase of the mGA is designed for this purpose. It allows highly fit strings to be enriched in a population prior to the messy genetic reproduction phase, the so-called a *juxtapositional* phase. Thus the probability of having erroneous bits (misleading bits) in the search process is reduced, and there is a higher probability of forming optimal or near-optimal structures at an early stage of the search process.

3.8 Two-Phase Evolution

The original messy genetic algorithm proceeds with an initial population by using a complete enumerative initialisation scheme, thereafter, the primordial phase is called to select good building blocks, and in a juxtapositional phase the *cut* and *splice* operators combine the good building blocks to form the optimal or the near-optimal solutions.

In a complete enumerative initialisation, at least one copy of all possible building blocks of a specified length, *k*, is provided. A population size of $NN = 2^k \binom{l}{k}$ is required for capturing deceptive nonlinearities of order *k*. Thus, the initialisation of the messy GA is not exactly random. Rather, it provides all possible order-*k* equivalence classes (building blocks) in the initial population.

A primordial selection phase follows the complete enumerative initialisation for an era of order-*k*. During this phase, selection of highly fit strings is performed only, i.e. without activating any messy genetic reproduction. As a result, no extra fitness evaluations are required. As selection proceeds, the population size is reduced by halving the size every generation. The primordial phase continues until the population size reaches a prescribed size which is then used in the juxtapositional phase.

In the juxtapositional phase, the mGA invokes the cut and splice operators and other operators like mutation operators to juxtapose the good building blocks. It is similar to the action of a standard GA during this phase except that cut and splice are used. The population size is kept constant from generation to generation in the juxtapositional phase, and the operator probabilities are also kept constant.

The length of the strings increases over the messy GA optimisation process. The strings have the same length of the order of the building blocks (e.g. $l = k$, where $k = 1, 2,$

3, ...) in the initial population and the primodial phase. In justapositional phase, cut and splice genetic operations are applied to each of the strings (according to the cut and splice probabilities). At early stage of the justapositional phase, the length of the strings is small, the cut probability is low as indicated by Eq.(3.15). The splice operation is more likely applied to generating the next generation than the cut operation. Thus the strings grow in length. However, the longer the strings are, the higher the cut probability becomes, the more likely the strings are cut. The length of the strings stays at a certain level while the cut probability is about the same as the splice probability.

```
{
    era = 0; /*the order of building block*/
    while ( messy GA termination is not true ){
        era = era + 1;
        complete enumerative initialisation;
        evaluation; /*network solver is called */
        t = 0;
        while (primordial phase is true ){
            threshold tournament selection;
            t = t + 1;
        }
        while (juxtapositional phase is true ){
            threshold tournament selection;
            cut;
            splice;
            mutation;
            evaluation; /*network solver is called */
            t = t + 1;
        }
    }
```

Figure 3-5 A Conceptual Framework of the Messy GA

The mGA components described above are combined into this two-phase evolution process, are shown in Figure 3.5. The search process is followed over a certain number of eras. The eras are taken as k = 1, 2, 3, ..., etc (where k is order of the building blocks). In each era, an initial population—a complete enumeration of *era*-order building blocks—strings of *era* fixed positions are initialised, and evaluated by using a randomly generated template. For the first era, order one building blocks are initialised. For example, the entire set of order one building blocks of 3-bits string are 1 * *, * 1 *, * * 1, 0 * *, * 0 * and * * 0, where * is *don't care* character (Goldberg 1989), and are filled in by the template. After the initialisation, the population is enriched with fitter strings in the primordial phase, and evolved in the juxtapositional phase. Thus the first era search is completed. The second era search is commenced by initialising the order two building blocks (for example 1 1 *, 1 * 1, * 1 1, 0 0 *, 1 * 0, * 1 0), which are evaluated by using the best string from the first era as the template. Over a number of eras (user-specified), the optimum solution can be found in the evolved population.

3.9 Implementation of the Messy Genetic Algorithm for Optimisation of Pipe Networks

The computer program mGANET for optimisation of the design and/or rehabilitation of water distribution systems has been implemented in this research by integrating the mGA (Deb and Goldberg 1991) with the hydraulic network solver EPANET (Rossman 1994). The mGA searches for new low cost designs or rehabilitation alternatives for water distribution systems. EPANET simulates the water flow in the system and gives information on flow rates for pipes, hydraulic grades or pressures at nodes. Integrating the mGA with EPANET required an interface program for linking the subroutines together, evaluating the costs of the alternatives, and checking the actual pressures against the minimum allowable pressures at nodes. This interface program has been developed and coupled with mGA and

EPANET as a program named mGANET for the optimal design and/or rehabilitation of water distribution networks.

3.9.1 The messy genetic algorithm

The mGA provides a general optimisation programming tool. It allows users to add their own optimisation functions, specify genotypical encoding bits and decoding formats. This program was originally developed at the Genetic Algorithm Laboratory at University of Illinios, and has been employed in this research project.

The mGA consists of four main components including `get_input()`, `initialize()`, `generate()` and `functions.c`, which are summarised as follows.

Functions	**Purpose**
`get_input()`	This function reads all user specified parameters for controlling the messy genetic algorithm, sets up the template and gets information on objective function from input files.
`initialise()`	This function calls routines to create the initial population and to report population statistics. It also invokes a function that sets up the population reduction schedule in the primordial phase.
`generate()`	This function performs one generation of the messy GA operation. If the generation counter, **gen,** is less than **prim_gen**, the primordial phase is called, otherwise the juxtapositional phase is called. Population statistics of the new generation are calculated and reported. The routine terminates by assigning the new generation to the old population.
`functions.c`	The file contains all optimisation functions that are used as subfunctions. Users define the objective function(s) in this file.

Two functions `file8()` in the file `functions.c` has been defined respectively for the discrete pipe model described earlier. The cost function begins by extracting the genes for each rehabilitated pipe and decoding the genes as a rehabilitation event and a pipe size for each old pipe, and then extracts the genes for each new pipe and decodes the genes as the new pipe diameters. The conceptual procedure for the cost function is given in Figure 3-6 and it terminates by calling the `totalcost()` from the interface program to calculate the total cost of rehabilitated and/or expanded water distribution systems.

```
{
        for each old pipe {
                extract gene for rehabilitation policy;
                decode the policy;
                extract gene for rehabilitation pipe size;
                decode the size;
                        }
        for each new pipe {
                extract gene for pipe size;
                decode the size;
                        }
        call totalcost();
}
```

Figure 3-6 Framework of `file8()` for Linking mGA Search with Pipe Cost Function

3.9.2 Hydraulic solver

As discussed before, a hydraulic network solver was needed to simulate the water flow in distribution systems. A program called EPANET (Rossman 1994), a tool for water quantity and quality simulation, has been used in this research project. EPANET is written in C and developed by U.S. Environmental Protection Agency. It contains files which include `main.c`, `smatrix.c`, `hydaul.c`, `quality.c`, `input1.c` and `input2.c`.

Since water quality is not being considered in this research, the file `quality.c` was left out of the integration. In order to compare the simulation results from the final optimisation from mGANET with a pure EPANET simulation, the EPANET input format is used in the integrated program mGANET. A function `hydsolve()` links routines from `hydraul.c` and `smatrix.c`, and performs the water flow simulation. It is this function that is called for each chromosome representing one alternative of optimal rehabilitation and design of the water distribution system under consideration.

3.9.3 The Interface program

The interface file `pipe.c` has been written for linking the routines in mGA and EPANET, and performing the optimisation of the rehabilitation and pipe-sizing for the distribution systems. The `pipe.c` includes functions as described below.

Functions	**Purpose**
`ReadPipeInfo()`	the unit costs for new pipes, unit rehabilitation costs for old pipes, demand(s) and minimum hydraulic grade (or pressure) for each node are read from a data file.
`Oldpipecost()`	calculate costs for rehabilitating old pipes by using discrete model.
`newpipecost()`	calculate costs for new expansion pipes by using discrete model
`maxHeadDeficit()`	finds the maximum hydraulic grade (or pressure) deficit in the system considering all demands
`updatediam()`	updates the diameters for old and new pipes.
`updatecoff()`	updates roughnesses for cleaned and/or lined pipes.
`totalcost()`	calculates the total cost of rehabilitation and pipe-sizing by using a discrete pipe model

All the routines or functions are linked by the function `totalcost()` for the discrete model. The cost function has the programming structure for the total cost computation as given in Figure 3-7. The cost evaluation starts by updating friction coefficients due to the rehabilitation such as cleaning and lining of an old pipe, and is followed by assigning the new set of diameters, decoded from `file8()` to diameter variables in EPANET. The network solver is called for each demand loading case, and the maximum hydraulic pressure (or grade) deficit, found for each of the demand cases, is used for the penalty cost computation. The total cost, evaluated for one solution, is the sum of the network cost and the penalty cost.

```
{
        updatecoeff();
        updatediam();
        for each demand case {
                hydsolve(); /*performs EPANET hydraulic simulation*/
                maxHeaddeficit();
                                }
        penaltycost = maxdeficit*penaltyfactor;
        networkcost = Oldpipecost() + newpipecost(); /*for discrete model*/
        totalcost = penaltycost + networkcost;
}
```

Figure 3-7 Pseudo Code for `totalcost()`

3.9.4 The structure of mGANET

The mGANET program provides an overall structure that initiates the messy GA, links optimisation models (the interface program), and activates the hydraulic network solver EPANET. The main structure of the mGANET is given as in Figure 3-8. It starts by getting the data for messy GA, optimisation models and EPANET from input files as listed in Table

3-3. The mGANET terminates by outputting the best individual pipe network design from the messy GA. Table 3-4 gives more detail about the outputs.

The mGA is invoked by initialising a population of chromosomes for each era. Each era contains a certain order of building block, missing genes are filled by using a random or user-specified template for the initial population. The template will be replaced by the best string (i.e. lowest cost) found whenever the generation best string is found. One complete representation of each chromosome is evaluated by calling function `file8()`, which then calls cost functions `totalcost()` in the interface program. It is the cost functions that activate the hydraulic network solver EPANET and that contribute to the fitness, if there is a penalty cost, for the genotypical representation. After the initialisation, the messy GA enters the primordial phase within which individuals are selected by performing just the threshold tournament competition, in the meantime, the population size is reduced to a prescribed level. The juxtapositional phase performs genetic operations similar to a standard GA, but using messy operators such as cut and splice, and mutation operations if necessary as described earlier. This phase is executed until the specified maximum permitted number of generations for this era is reached. The best string found at the end of the first era is used as the template for the initial population in the next era. Over several eras, the best individual is found to be the optimum solution.

Table 3-3 Input Files for mGANET

Input Files	*Description*
`Parameter`	user-specified messy GA global parameters such as number of eras (levels), probabilities for genetic operations, problem sizes etc.
`Subfunc`	Contains number of subfunctions, genes specifying each function.
`Era`	Includes population size and the maximum generations for each juxtapositional phase.
`Template`	contains template information for a complete chromosome.
`Pipeinfo`	required for optimisation models, it involves unit cost information, penalty factors, demand loading cases and minimum allowable pressures at nodes.
`*.inp`	Information required by EPANET.

Table 3-4 Output Files for mGANET

Output Files	*Description*
`output`	Population records for each generation, it includes the best fitness, average fitness and the worst fitness for each generation
`solution.opt`	Optimum rehabilitation policy and pipe sizes, corresponding cost, and hydraulic grade lines at nodes.

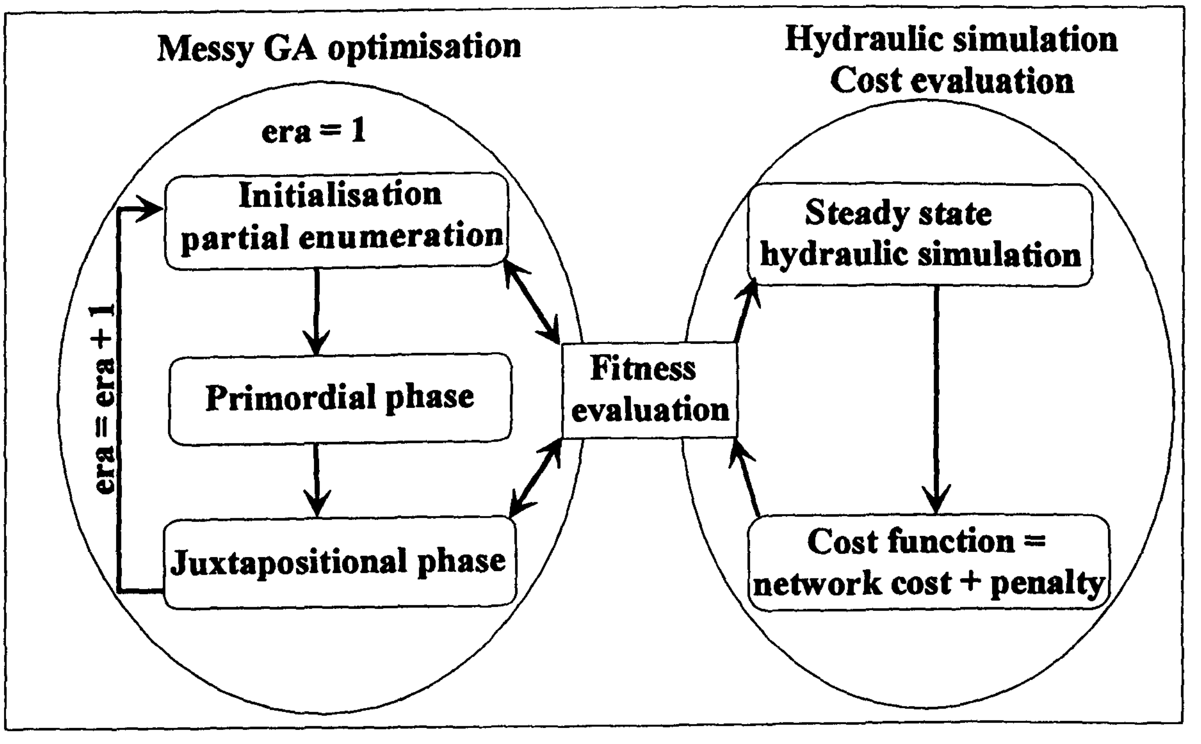


Figure 3-8 The mGANET Integration Structure

3.10 Application and Results

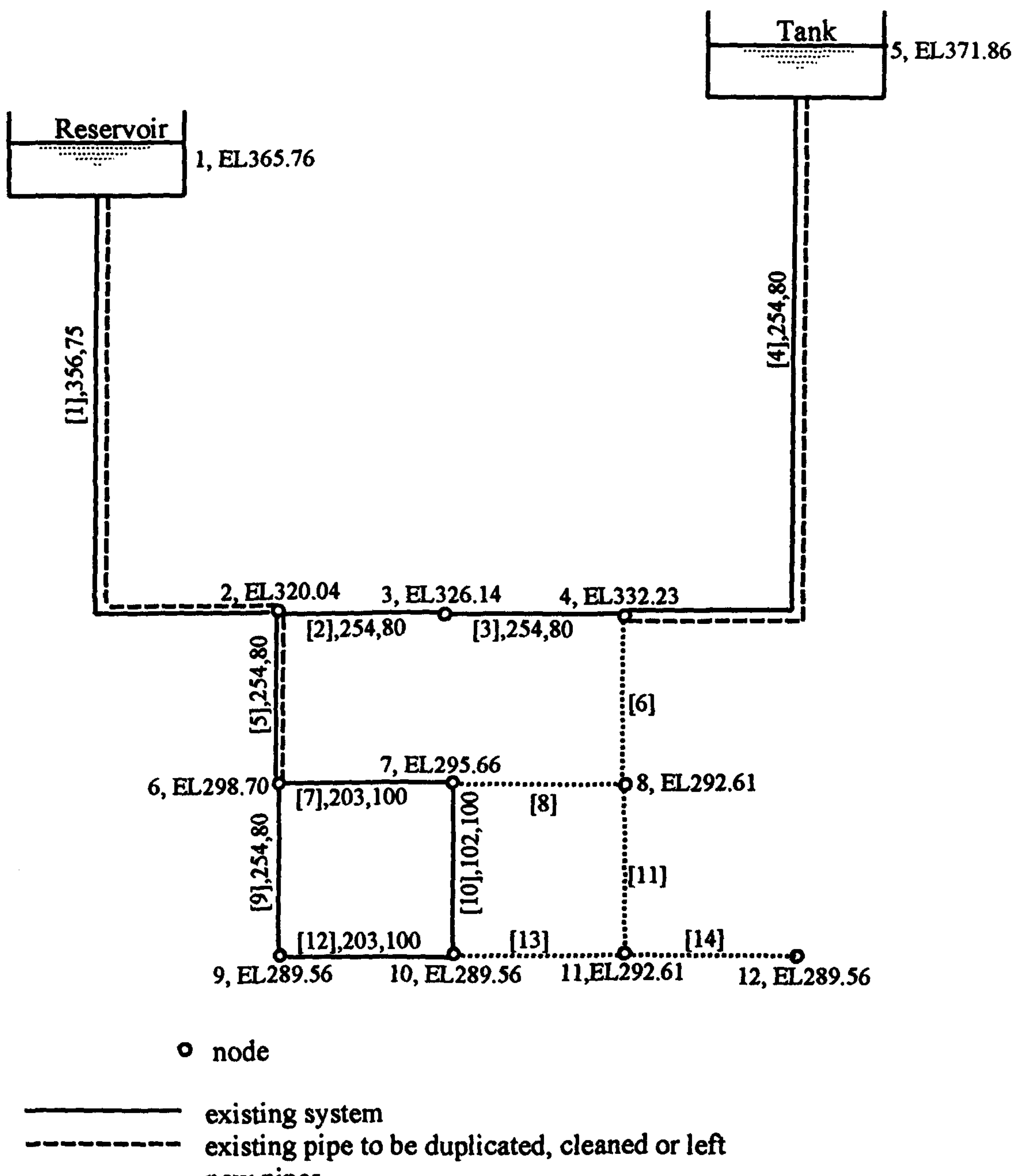
Two problems, a two-reservoir network problem and the New York City water supply tunnels, have been chosen for testing the messy GA based approach for optimal rehabilitation and pipe-sizing of water supply systems. Both problems have been previously studied in literature and are good examples for testing the program mGANET.

3.10.1 Case study I: the two reservoir network

A network with two water supply sources and fourteen pipes has been studied by Simpson et al. (1994) and is shown in Figure 3-9. A complete analysis has been given by applying complete enumeration, linear programming, non-linear programming and the standard GA (Murphy & Simpson 1992; Simpson, Dandy & Murphy 1994). The global optimum solution and a set of ranked solutions for this problem was found by using discrete pipe sizes (equivalent to discrete optimisation model proposed in section 3.2) using complete enumeration of every alternative. The results from the previous studies provide an excellent example for testing the program mGANET and comparing the performance of the messy GA approach with the standard GA approach.

The layout of the network is given in Figure 3-9. It contains two reservoirs, five new pipes to be sized, and nine existing pipes, three of which may be rehabilitated by a pipe in parallel (referred to duplication although a different diameter may actually be selected), cleaning or alternatively left as they are. In Figure 3-9, solid lines represent the existing system, and dashed lines represent the parts of the system where pipe links [1], [4] and [5] may be rehabilitated, and pipe links [6], [8], [11], [13] and [14] that are to be sized with at least a minimum diameter pipe. Table 3-5 gives the pipe costs and available diameters.

Three demand cases including two fire loading cases and one peak day loading case and the associated minimum allowable pressure heads are shown in Table 3-6.



Pipes: [1],356,75 [pipe number], diameter(mm), Hazen-Williams roughness C

Note. 1. All pipe lengths are 1609m, except pipe[1]=4828m and pipe[4]=6437m.

2. C=120 for new pipes and cleaned pipes.

Nodes: 2, EL320.04 node number, node elevation(m)

Figure 3-9 The Two Reservoir Network *(from Simpson et al. 1994)*

Table 3-5 Available Pipe Sizes and Associated Costs for the Two Reservoir Network

Pipe Diameter (mm)		Cost for a new pipe ($/m)	Cost for duplicating a pipe ($/m)	Cost for cleaning a pipe ($/m)
152	d_0^0	49.54	49.54	47.57
203	d_1^0	63.32	63.32	51.51
254	d_2^0	94.82	94.82	55.12
305	d_3^0	132.87	132.87	58.07
356	d_4^0	170.93	170.93	60.70
407	d_5^0	194.88	194.88	63.00
458	d_6^0	232.94	232.94	-
509	d_7^0	264.10	264.10	-

Table 3-6 Demand Patterns and Associated Minimum Allowable Pressures for the Two Reservoir Network

Node	Demand Pattern 1		Demand Pattern 2		Demand Pattern 3	
	Demand (L/s)	Minimum allowable pressure head (m)	Demand (L/s)	Minimum allowable pressure head (m)	Demand (L/s)	Minimum allowable pressure head (m)
2	12.62	28.18	12.62	14.09	12.62	14.09
3	12.62	17.61	12.62	14.09	12.62	14.09
4	0	17.61	0	14.09	0	14.09
6	18.93	35.22	18.93	14.09	18.93	14.09
7	18.93	35.22	82.03	10.57	18.93	14.09
8	18.93	35.22	18.93	14.09	18.93	14.09
9	12.62	35.22	12.62	14.09	12.62	14.09
10	18.93	35.22	18.93	14.09	18.93	14.09
11	18.93	35.22	18.93	14.09	18.93	14.09
12	12.62	35.22	12.62	14.09	50.48	10.57

mGA Coding, Decoding and Parameters

To apply messy GA to discrete optimisation of two reservoir network problem, 3 binary bits have been used to represent each pipe size variable of the five new pipes and three existing pipes. Each of the 3 binary bits represents 8 possible choices of pipe sizes. Two binary bits have been used for each existing pipe to represent 3 possible choices of rehabilitation actions that include cleaning, leaving or duplicating an existing pipe. Thus 30 bits are needed for solving the problem by using the discrete formulation. A binary coding and decoding scheme for the all possible pipe sizes and rehabilitation actions of the two reservoir network are given in the Table 3-7 and Table 3-8. The penalty factor for pressures which do not meet the minimum allowable pressure constraint for this problem was chosen to be $5000/m of deficit to match the value taken by Simpson et al. (1994). The other parameters used as follows:

Splice probability	1.0	Maximum number of eras	3
Cut probability	0.017	Juxtapositional phase population size	150
Mutation Probability	0.01	Maximum generations	10

Table 3-7 Coding and Decoding Scheme of Available Pipe Sizes for the Two Reservoir Network

Pipe Diameter (mm)	Binary substring corresponding to the pipe size	Index by Eq.(3.10)	Pipe size notation (corresponding to Table 3.11)
152	000	0	d_0^0
203	001	1	d_1^0
254	010	2	d_2^0
305	011	3	d_3^0
356	100	4	d_4^0
407	101	5	d_5^0
458	110	6	d_6^0
509	111	7	d_7^0

Table 3-8 Coding and Decoding Scheme of Possible Rehabilitation Actions for the Two Reservoir Network

Actions	Binary string	Index by Eq.(3.12)	Rehabilitation action notation
Leaving a pipe	00	0	e_0^0
Duplicating a pipe	01	1	e_1^0
Cleaning a pipe	10 or 11	2 or 3	e_2^0

Mapping a genotype to a phenotype for this problem follows the mapping scheme described in Section 3.6. An example of mapping one genotype to its corresponding phenotype for the two reservoir network is given in Table 3-8a. A genotype, for example 101110001111011011010011001111, is first divided into five substrings of 3 bits for each of five new pipes added to the network, and three substrings of 5 bits (2 bits for coding the rehabilitation actions and 3 bit for the associated coding diameter) for each of three existing pipes. The diameter and action binary strings are decoded into integer indexes by the Eq.(3.10) and Eq.(3.12). The sizes or diameters of the pipes added and duplicated were found from the Table 3-5 by the mapped indexes. The pipe size was the same as the original size if the pipe was left or cleaned. Similarly, the rehabilitation action for each existing pipe was found from Table 3-8 by the mapped index. The network cost of the solution can be calculated by the sizes for new and duplicated pipes and the rehabilitation actions for the existing pipes. The penalty cost for cases where design criteria are not satisfied is calculated by calling the hydraulic solver EPANET. Thus the genotype has been mapped to a phenotype which represents a particular design solution, and the phenotype (solution) has been evaluated to contribute to the fitness of the genotype.

Table 3-8a An Example of Mapping a Genotype to a Phenotype for the Two Reservoir Network

Pipe Id	[6]	[8]	[11]	[13]	[14]	[1]		[4]		[5]	
Pipe Tag	1	2	3	4	5	6		7		8	
Variable	d_1	d_2	d_3	d_4	d_5	e_1	d_6	e_2	d_7	e_3	d_8
Genotype	101	110	001	111	011	01	101	00	110	10	111
Index	5	6	1	7	3	1	5	0	6	2	7
Phenotype	d_5^0	d_6^0	d_1^0	d_7^0	d_3^0	e_1^0	d_5^0	e_0^0	d_6^0	e_2^0	d_7^0
A solution	407	458	203	509	305	dup.	407	leave	-	clean	254
(diameter or action)											

Results and Comparison

The optimum discrete solution for this problem was found by mGANET with different random seeds and compared with the standard GA results (Simpson et al. 1994; Simpson & Goldberg 1994) and are shown in Table 3-9. A typical convergence rate for the mGANET solution with the seed 0.7 is given in Figure 3-10

As shown in Table 3-9, the messy GA found the lowest cost solution (global optimum) in each of the 10 runs with different random seeds. The number of mGANET evaluations needed for achieving the optimal are less than the standard GA, being only one third to half of the evaluation numbers of the standard GA (Simpson et al. 1994), and also less than the GA with tournament selection (selection pressure $s = 2$) (Simpson & Goldberg 1994). Thus it is shown that the messy GA search for the optimal solution is more efficient and effective than the standard GA.

Most of the evaluations in the messy GA have been taken by initialisation process. The messy GA started the first era with order 1 bit combinations for the initialisation by complete enumeration. A total of 300 one-bit strings were generated and evaluated by using a randomly generated gene string as a competitive template. The initial population size was reduced to 150 (the population size of the justapositional phase) at the end of primordial phase for era 1 and was followed by the juxtapositional phase. In the second era initialisation, 435 two-bit strings were initialised by complete enumeration with order 2 and evaluated by using the best solution as the competitive template from the first era. For the third era, 4060 three-bit strings were initialised by complete enumeration with order 3, as shown in the Figure 3-10. The number of evaluation required for the initial population for all 3 eras is more than 50% of the total number of evaluations. This disadvantage must be overcome to enable the messy GA to be applied to more complicated network design problems. A fast messy GA is introduced in Chapter 6, which overcomes this problem.

Table 3-9 Results of Messy GA Compared with a Standard GA for the Two Reservoir Network

	GA (Simpson et al. 1994) Roulette wheel selection		GA (Simpson & Goldberg 1994) Tournament selection (s = 2)		Messy GA	
Run No.	Cost (dollars) (% difference from optimum)	Achieved at evaluation number	Cost (dollars)	Achieved at evaluation number	Cost (dollars)	Achieved at evaluation number
1	1,791,000 (2.3%)	23,400	1,750,300	9,000	1,750,300	6,148
2	1,750,300*	10,350	1,750,300	9,500	1,750,300	6,148
3	1,841,700 (5.2%)	22,410	1,750,300	8,500	1,750,300	6,148
4	1,839,000 (5.1%)	15,660	1,750,300	9,500	1,750,300	8,958
5	1,750,300	17,190	1,750,300	8,000	1,750,300	2,957
6	1,750,300	11,070	1,750,300	8,000	1,750,300	2,522

Run No.	GA (Simpson et al. 1994) Roulette wheel selection		GA (Simpson & Goldberg 1994) Tournament selection (s = 2)		Messy GA	
	Cost (dollars) (% difference from optimum)	Achieved at evaluation number	Cost (dollars)	Achieved at evaluation number	Cost (dollars)	Achieved at evaluation number
7	1,750,300	10,080	1,750,300	8,000	1,750,300	8,758
8	1,799,900 (2.8%)	4,410	1,750,300	7,500	1,750,300	10,042
9	1,750,300	12,510	1,750,300	10,000	1,750,300	3,977
10	1,750,300	19,890	1,750,300	10,000	1,750,300	6,148
Average		14,697		8,800		6,181

•The global optimum solution

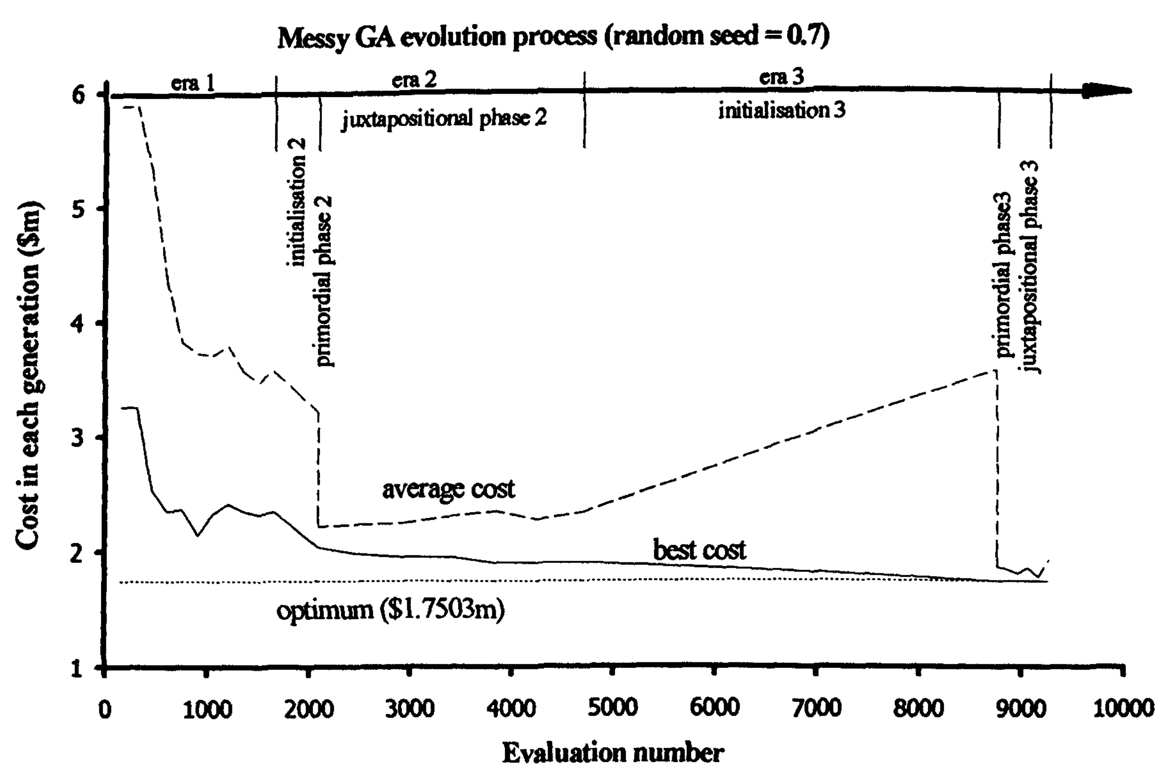


Figure 3-10 Generation Cost Statistics for a mGANET Run for the Two Reservoir Problem

3.10.2 Case study II New York city water supply tunnels

The New York water supply tunnel system consists of one water supply source Hillview reservoir, and two main city tunnels City Tunnel No. 1 and City Tunnel No. 2. The layout is shown in Figure 3-11. The problem was first posed in 1969 by Schaake & Lai to size the optimal pipe diameters by using a fitted cost function $c_f = 1.1d^{1.24}$, where d = diameter of pipe. The pipe sizes are discretised from the function and listed in Table 3-10. The minimum required hydraulic grades are shown in Table 3-11 for each node. This problem has been studied by many researchers (Schaake and Lai 1969; Quindry et al. 1981; Gessler 1982; Bhave 1985; Morgan and Goulter 1985; Dandy et al. 1996). The previous studies provide a sound basis for comparison with the new messy GA based approach.

Table 3-10 Available Pipe Sizes and Costs for New York City Tunnels Expansion

Diameter (in)	Pipe cost ($/ft) (1969 price)*	Diameter (in)	Pipe cost ($/ft) (1969 price)	Diameter (in)	Pipe cost ($/ft) (1969 price)
36	93.5	96	316.0	156	577.0
48	134.0	108	365.0	168	632.0
60	176.0	120	417.0	180	689.0
72	221.0	132	469.0	192	746.0
84	267.0	144	522.0	204	804.0

*based on cost function $c = 1.1d^{1.24}$

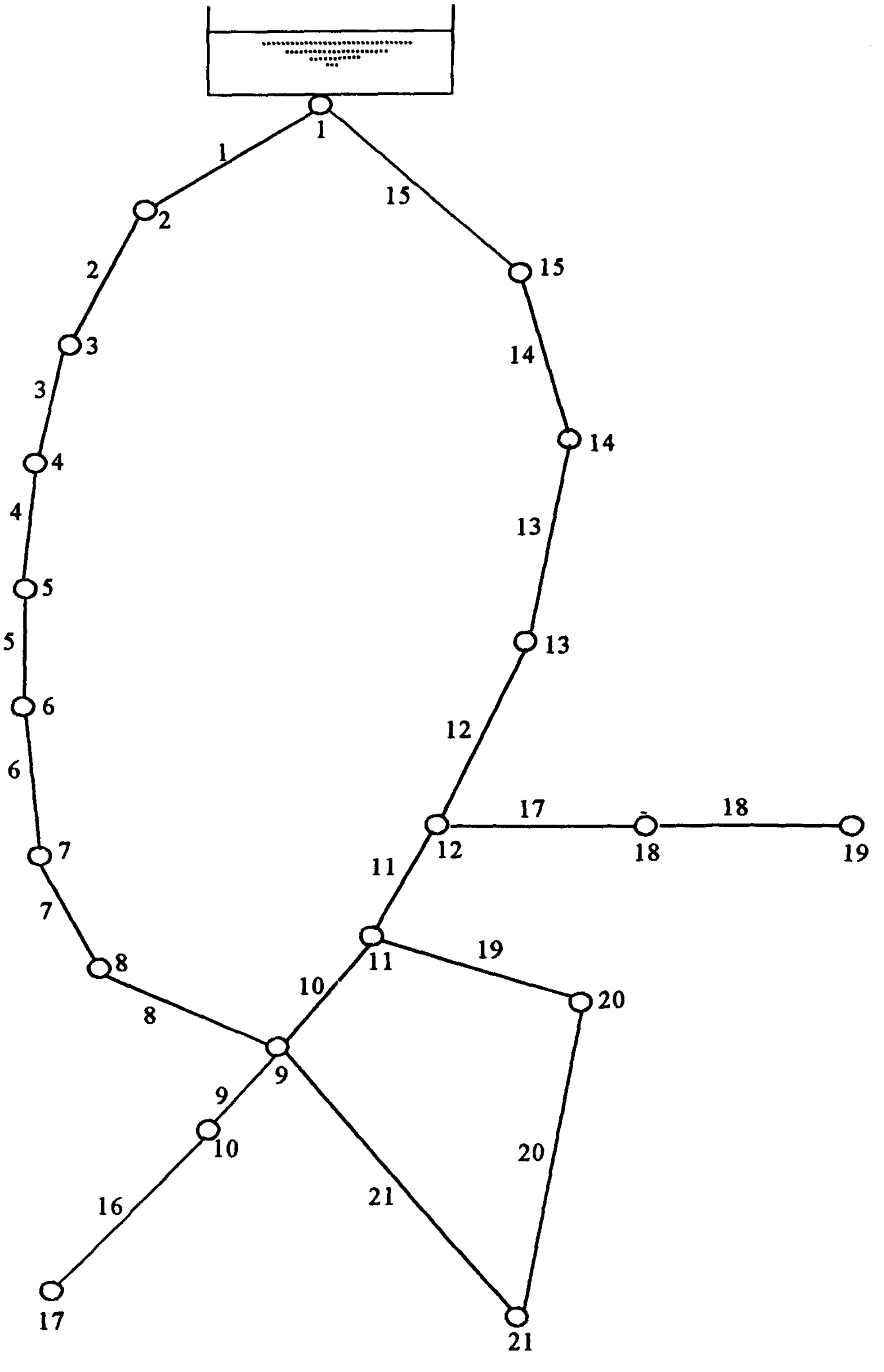


Figure 3-11 New York City Water Supply Tunnels *(from Dandy et al. 1996)*

Table 3-11 Node Data for New York Water Supply Tunnels

Node No.	Demand (cfs)	Minimum Allow-able HGL (ft)	Node No.	Demand (cfs)	Minimum Allow-able HGL (ft)
1	reservoir	300.0	11	170.0	255.0
2	92.4	255.0	12	117.1	255.0
3	92.4	255.0	13	117.1	255.0
4	88.2	255.0	14	92.4	255.0
5	88.2	255.0	15	92.4	255.0
6	88.2	255.0	16	170.0	255.0
7	88.2	255.0	17	57.5	272.8
8	88.2	255.0	18	117.1	255.0
9	170.0	255.0	19	117.1	255.0
10	1.0	255.0	20	170.0	255.0

Table 3-12 Pipe Data for New York City Water Supply Tunnels

Pipes No.	Start node	End node	Length (ft)	Existing Diameter (in)	Pipes No.	Start node	End node	Length (ft)	Existing Diameter (in)
[1]	1	2	11600	180	[12]	13	12	12200	204
[2]	2	3	19800	180	[13]	14	13	24100	204
[3]	3	4	7300	180	[14]	15	14	21100	204
[4]	4	5	8300	180	[15]	1	15	15500	204
[5]	5	6	8600	180	[16]	10	17	26400	72
[6]	6	7	19100	180	[17]	12	18	31200	72
[7]	7	8	9600	132	[18]	18	19	24000	60
[8]	8	9	12500	132	[19]	11	20	14400	60
[9]	9	10	9600	180	[20]	20	16	38400	60
[10]	11	9	11200	204	[21]	9	16	26400	72
[11]	12	11	14500	204					

All pipes, existing and new, are assumed to have a Hazen Williams C = 100 *mGA coding, decoding and parameters*

There are 15 choices in Table 3-10 of pipe sizes for each pipe to be rehabilitated. In order for the mGA to search for the best combination of pipe sizes, four bits (providing 16 choices, 0000 was used to represent *leaving a pipe as it is* for this problem) were used to code the possible sizes for each pipe, so that a total of 84 binary bits were used to represent the problem to be optimised.

Previous Recent Studies of the New York Problem

Dandy et al. (1996) improved the standard GA approach (Simpson et al. 1994) by using a fitness scaling technique, Gray coding scheme and creeping mutation, and applied the improved GA to the New York problem. The optimisation results were checked using the KYPIPE hydraulic network solver. The optimal discrete pipe solution was found to be $38.80 million.

Savic & Walters (1997) found that the optimal solution was quite sensitive to the coefficients of Hazen-Williams pipe head loss formula. They integrated a standard GA with EPANET, and found least cost solutions corresponding to low and high bounds of coefficients in the Hazen-Williams formula used by number of researchers. The results from the improved and standard GA are compared with messy GA design solutions in this Chapter.

mGANET Results and Comparisons

Four bits were used for each pipe in the New York city Tunnels problem, giving a total of 84 bits for 21 pipes for the discrete pipe formulation. The messy GA parameters which were used are as follows.

(i) Primordial phase: random seed = 0.9; maximum number of eras = 2; maximum number of generations for each era = 100; initial population size for era 1 = 420; and initial population size for era 2 = 13,944.

(ii) Juxtpositional phase: juxtapositional population size *NN* = 500; cut bitwise probability P_k = 0.02; splice probability P_s = 0.9; and mutation probability P_m = 0.01.

The least cost solution for the New York city tunnels problem was determined by Savic & Walters (1997) for various values of coefficients of the Hazen-Williams formula. Savic and Walters (1997) identified the low and high bound of the coefficients of Hazen-Williams formula used by Fujiwara & Khang (1992) and Quindry et al. (1981) from a number of combinations of values used by various researchers. Savic and Walters (1997) presented the following:

A low bound of the Hazen-Williams formula:

$$h_f = 4.6847L\left(\frac{Q}{C}\right)^{1.85} D^{-4.87} \qquad (3.16)$$

A high bound of the Hazen-Williams formula:

$$h_f = 4.8306L\left(\frac{Q}{C}\right)^{\frac{1}{0.54}} D^{-4.87} \qquad (3.17)$$

Other forms of the Hazen-Williams formula include that used by EPANET (Rossman 1994) given as:

$$h_f = 4.72L\left(\frac{Q}{C}\right)^{1.85} D^{-4.87} \tag{3.18}$$

In order to test the messy GA, and compare it to the standard GA (Savic and Walters 1997), a set of three mGANET runs have been performed. One set of the runs used the EPANET original Hazen-Williams formula Eq.(3-18); the second set of the runs used the low bound of Hazen-Williams formula Eq.(3-16) and the third one used the high bound of Hazen-Williams formula Eq.(3-17). The results are summarised and compared in the Table 3-13 and Table 3-14. As shown in Table 3-13, the discrete messy GA optimal solution for the low bound Hazen-Williams formula is the same as the standard GA approach. For the high bound of Hazen-Williams formula, however, the messy GA found a lower cost discrete solution than the standard GA.

For comparison the solution found with a particular Hazen-William formula was evaluated by using the other two forms of the Hazen-William formulas. It has been observed that the optimal solutions are governed by the minimum hydraulic pressure heads at certain nodes. As shown in Table 3-14, the greater the hydraulic pressure deficit, then the lower cost of the solution that is found. The hydraulic deficits of different optimal solutions by different Hazen-William formulations Eq. (3-16) to Eq. (3-18) also provide useful information to water engineers to gauge the sensitivity of the optimal solution for selection of the head loss formula.

Table 3-13 Comparing the mGANET Designs with Previous GA Solutions

Duplicated pipe	Savic & Walters (1997)		Messy GA run		
	Hazen-Williams formulation		Hazen-Williams formulation		
	low	high	EPANET	low	high
	Eq. (3-16)	Eq. (3-17)	Eq. (3-18)	Eq. (3-16)	Eq. (3-17)
[7]	108	0	132	108	0
[15]	0	144	0	0	132
[16]	96	84	96	96	84
[17]	96	96	96	96	96
[18]	84	84	84	84	84
[19]	72	72	72	72	72
[21]	72	72	72	72	72
Cost ($million)	37.13	40.42	38.10	37.13	39.60
Evaluations	-	-	48,667	40,382	39,939

Table 3-14 Comparison of Hydraulic Heads at Node 17 Using Different Hazen-Williams Formulations

Hazen-Williams formula	Min. allowable head (ft)	Savic & Walters (1995)		Messy GA run	
		low*	high	EPANET	high
		Eq. (3-16)	Eq. (3-17)	Eq. (3-18)	Eq. (3-17)
Low	272.80	272.86	273.77	273.06	273.47
	Surplus =	+0.06	+0.97	+0.26	+0.67
EPANET	272.80	272.66	273.57	272.86	273.27
	Surplus =	**-0.14**	+0.77	+0.06	+0.47
High	272.80	272.25	273.18	272.46	272.88
	Surplus =	**-0.55**	+0.38	**-0.34**	+0.08
	Cost ($million)	37.13	40.42	38.10	39.60

*The messy GA discrete pipe solution found for low bound Hazen-Williams formula is the same as Savic and Walters (1995) solution for low bound Hazen-Williams formula.

In order to compare the performance of the messy GA with the improved GA (Dandy et al. 1996), the same Hazen William equation as used by Dandy et al. (1996) was adopted in this study. It is given as:

$$h_f = 4.7291L\left(\frac{Q}{C}\right)^{1.852} D^{-4.8704} \quad (3.19)$$

The program mGANET was run several times with different penalty factors for violation of the minimum allowable HGL constraint. A set of least cost solutions that were found by the messy GA are listed in Table 3-15. Three lower cost solutions by the improved GA and the messy GA are also given in Table 3-16. The optimal hydraulic heads at nodes for the optimal solution of \$38.80 million by both the messy GA solution and the improved GA solution are compared in Table 3-17. It shows that the messy GA identified the same critical nodes as the improved GA and that the actual hydraulic heads of the optimal solution by the messy GA are almost the same (to the second decimal place) as that by the improved GA.

As shown in Table 3-15 and 3-16, the solutions found by the messy GA are very similar to the solutions by the improved GA, but the messy GA is more efficient than the improved GA at searching for the optimum design of water distribution systems. The improved GA required 143,790 evaluations on average of five GA runs to reach the optimal or near-optimal solution. In contrast, the messy GA evaluated 48,427 alternatives on average of five mGA runs to achieve the similar optimal or near-optimal solution. The number of evaluations required by the messy GA is about one third of the evaluations by the improved GA for this case study. This has shown that the messy GA is more efficient than the improved GA at searching for the least cost design solution for water distribution systems.

Table 3-15 Results of the Messy GA Runs Compared with the Improved GA

Improved GA (Dandy et al.1996)			Messy GA			
GA runs	Lowest cost ($million)	Achieved at evaluation number	GA runs	penalty factor ($/ft)	Lowest cost ($million)	Achieved at evaluation number
1	38.80	96,750	1	9,000,000	38.80	49,587
2	39.06	137,400	2	13,000,000	39.06	42,787
3	38.80	151,400	3	11,000,000	38.80	48,387
4	39.06	145,700	4	15,000,000	40.17	53,187
5	39.17	187,700	5	7,000,000	38.64*	48,187
	Average =	143,790			Average =	48,427

* The hydraulic pressure constraint at the node 15 is violated by 0.02 (ft).

Table 3-16 Comparing the mGANET Designs with Previous GA Solutions

Duplicated pipe ID	Optimal Diameters (inches)					
	Improved GA by Dandy et al. (1996)			Messy GA		
	GAs 1	GAs 2	GAs 3	mGA 1	mGA 2	mGA 3
[7]	0	144	156	0	144	144
[15]	120	0	0	120	0	0
[16]	84	96	96	84	96	96
[17]	96	108	96	96	108	96
[18]	84	72	84	84	72	84
[19]	72	72	72	72	72	72
[20]	0	0	0	0	0	0
[21]	72	72	72	72	72	72
Cost ($million)	38.80	39.06	39.17	38.80	39.06	38.64*
Evaluations	96,750	137,400	187,700	49,587	42,787	48,187

* The hydraulic pressure constraint at the node 15 is violated by 0.02 (ft).

Table 3-17 Actual Hydraulic Heads of the Optimal Solution of $38.80 Million

Node ID	Minimum allowable head (ft)	Actual hydraulic heads (ft)	
		Improved GA by Dandy et al. (1996)	Messy GA
1	reservoir	300.000	300.000
2	255.0	294.620	294.621
3	255.0	287.204	287.205
4	255.0	285.056	285.057
5	255.0	283.181	283.182
6	255.0	281.754	281.755
7	255.0	279.564	279.565
8	255.0	276.425	276.427
9	255.0	274.223	274.225
10	255.0	274.192	274.193
11	255.0	274.364	274.365
12	255.0	275.820	275.821
13	255.0	279.024	279.026
14	255.0	287.028	287.029
15	255.0	295.301	295.302
16	**260.0**	**260.524**	**260.523**
17	**272.8**	**272.860**	**272.861**
18	255.0	261.842	261.841
19	**255.0**	**255.705**	**255.703**
20	255.0	261.196	261.194

Figure 3-12 illustrates a typical convergence rate of the messy GA solution ($38.80 million) for the New York city tunnels problem. The convergence of the messy GA is compared with the improved GA in Figure 3-13. The messy GA, unlike the standard GA and the improved GA, starts with evaluating all possible building blocks of order 1 namely all the strings with one bit fixed. For this case study, it has been observed that the performance of the messy GA was not affected by reducing the initial population size from all the combinations of one or two-bit strings to half of the required strings in the initial population. Thus, instead of 840 one-bit strings being initiated for era one, an initial population of 420 one-bit strings and 13,944 two-bit strings were created in the messy GA initialisation of era one and two respectively. Each of the one or two-bit strings was evaluated by using a random string or a local optimal string as a competitive template to fill the missing genes. The short strings of high fitness were selected by applying the thresholding selection in the primordial phase. The messy GA enriched the highly fit building blocks of order 1 or 2 at the end of the primordial phase and resulted in a significant improvement in the genotype fitness and the objective function value as shown in Figure 3-12. This provides the messy GA a mechanism identifying good building blocks at early stage of the artificial evolution. In contrast, the improved GA is not purposely designed for identifying the building blocks. The search procedure of the improved GA, as illustrated in Figure 3-13, appears more random and noisy than the messy GA. The messy GA reached the optimal or near optimal solution more efficiently than the improved genetic algorithm.

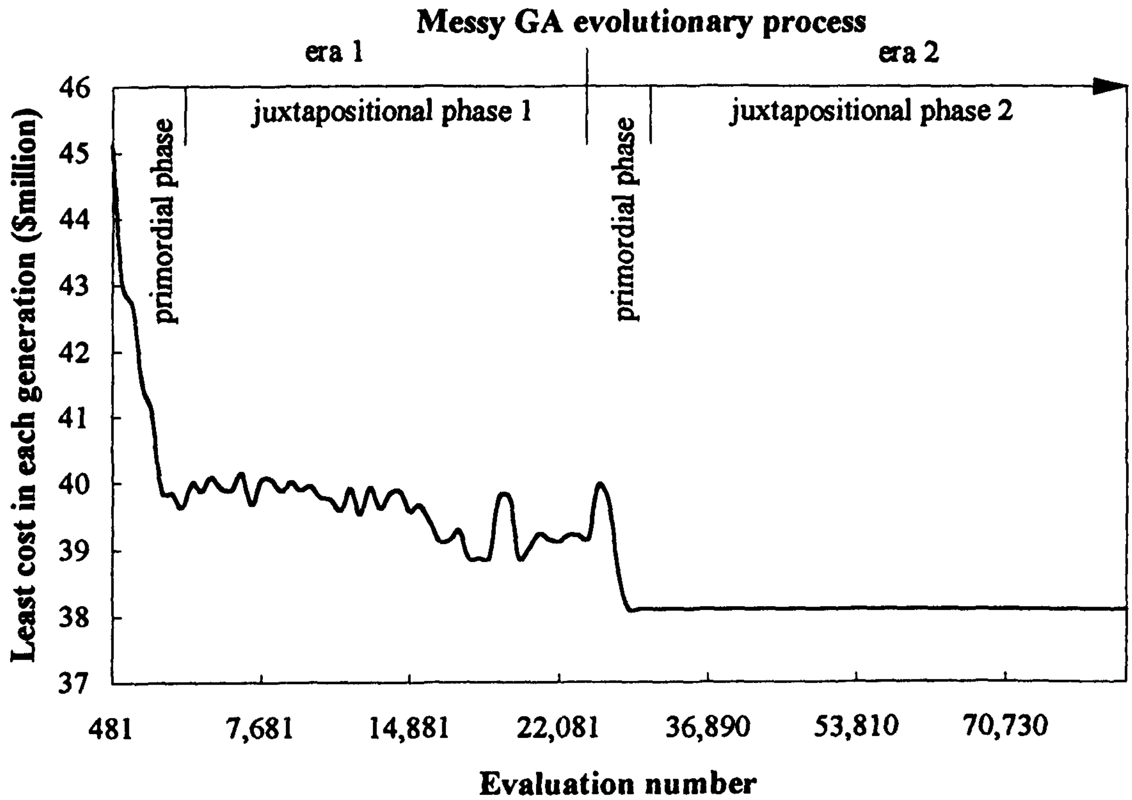


Figure 3-12 Convergence Behaviour of Messy GA for Discrete Pipe Optimisation of New York City Tunnels Problem (Optimal Solution $38.80 Million)

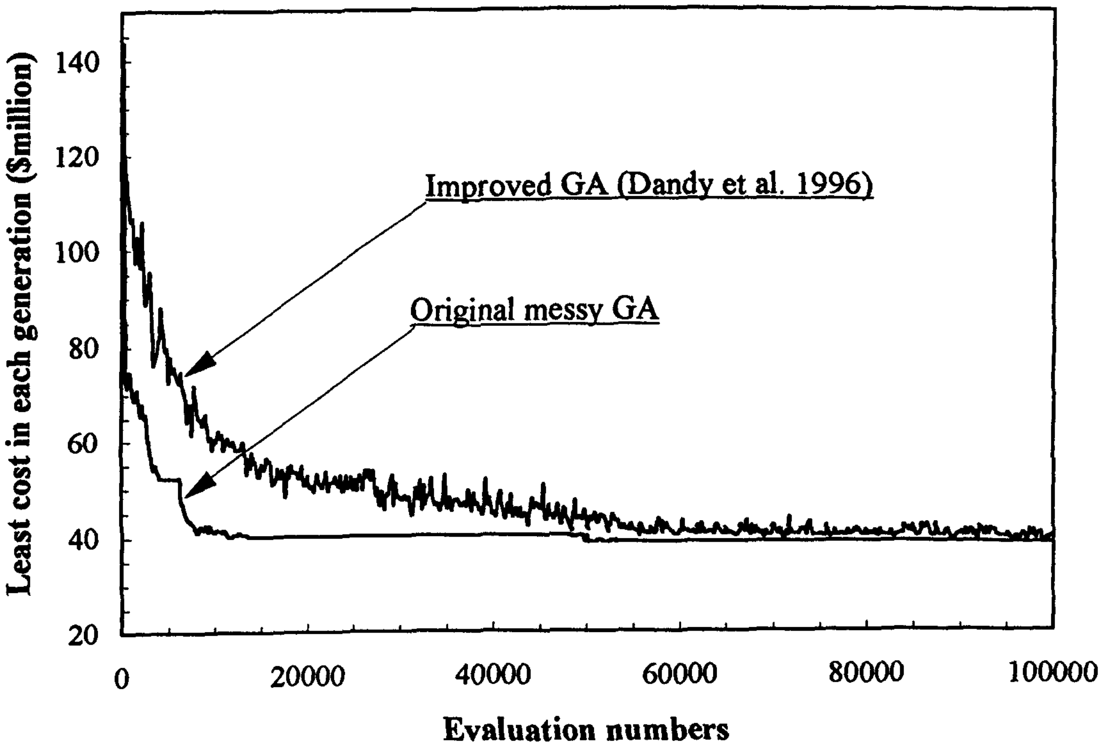


Figure 3-13 Comparison of Convergence Rates of the Improved GA (Dandy et al. 1996) and the Messy GA for the Discrete Pipe Solution $38.80 Million.

3.11 Summary

An optimisation model based on a discrete pipe formulation has been developed in this Chapter. A generalised genotypical representation and mapping scheme for genetic algorithm optimisation of water distribution systems is also presented. The optimisation model was based on a messy GA. It uses variable-length strings, thresholding selection and messy operators of cut and splice. The messy GA starts by enumerating a certain order of building blocks and is followed by a primordial phase and a juxtapositional phase. In the primordial phase, tournament thresholding selection occurs only to allow highly fit strings to be enriched in a population. The strings are evolved in the juxtapositional phase by performing the messy genetic operations of cut and splice. The messy GA has been integrated with the hydraulic network solver EPANET. The integrated program mGANET has been tested for the optimisation of water supply networks on the previously studied examples, the two reservoir network and the New York city tunnels problem. The results have shown that the messy GA approach to optimal sizing and rehabilitation of water distribution networks is more efficient than standard GA (tournament selection pressure s = 2) and/or improved GA approach. The approach will be able to provide the decision-maker a set of lower cost solutions for a comprehensive design of the water distribution systems.

Another type of model for optimisation of water distribution systems is a continuous model, in which the optimisation variables such as pipe diameters are treated as a set of continuous variables. The continuous model, however, finds it difficult to consider rehabilitation actions that are discrete, and also there is no guarantee for the continuous model to reach a real engineering optimal solution. This will be discussed in next Chapter.

4. SPLIT PIPE FORMULATION FOR GENETIC ALGORITHM OPTIMISATION OF WATER NETWORKS

4.1 Introduction

Over the last several decades, optimisation of design of water distribution networks has been studied. Optimisation techniques have been improved. In recent years, standard genetic algorithms (sGAs) have been applied to the optimal design of water distribution networks. It has been found that the sGAs are well suitable for solving this type of problem and outperform traditional optimisation techniques. In Chapter 3, it also has been demonstrated that the messy genetic algorithm (mGA), a new generation of the genetic algorithm, is more efficient and more reliable than sGAs at searching for global optimal design solutions of water distribution networks.

Optimisation models developed to date for optimal design of water distribution systems can be generally classified into two types including a discrete pipe model and a continuous pipe model. Both types of the models may be formulated with the objective of minimising the total cost of water distribution systems by searching for the optimum combination of pipe sizes and rehabilitation actions. The discrete pipe model treats the pipe size as a decision variable taking values directly from a discrete commercially available pipe size list, while the pipe size variables in the continuous model take the values between the minimum and maximum diameter. For a continuous formulation, a unit cost curve (function) is usually fitted by using the discrete pipe sizes and the corresponding cost data. The fitted cost function has been used previously for network cost evaluation in the search process

(Lansey and Mays 1989a; Fujiwara and Khang 1990; Ahn 1993; Lognanathan et al. 1995). A fitted continuous cost function may lead the optimisation process to a lower cost continuous solution than that by the discrete formulation, but the continuous solution is not desirable for practical application. This is because commercially available pipe sizes are discrete. Hence, after the continuous optimisation is carried out, the continuous optimal solution is usually converted to a hydraulically equivalent split discrete solution, namely a solution of two adjacent discrete pipe sizes used for an upstream pipe segment and a downstream pipe segment between a pair of nodes. However, it has been shown by Fujiwara and Dey (1987) that the optimal adjacent split pipe solution exists if and only if pipe costs are a strictly convex function of pipe diameters. In this Chapter, an analysis is presented to compare continuous pipe solution and split pipe solution. The outcome is that there appears to be no guarantee that the continuous model using a fitted convex function will reach an optimal split pipe solution. Thus a split pipe optimisation model is proposed. The model treats the pipe size as a continuous variable in the formulation, but each continuous solution is converted into a split pipe solution during the optimisation process and evaluated by using the cost information of commercially available pipe sizes. It eliminates the need for fitting a continuous cost function and guarantees an optimal split pipe solution to be found. The approach employs the original messy genetic algorithm as a search algorithm and comes up with a discrete optimal split pipe solution for the design and rehabilitation of water distribution networks.

4.2 Analysis of a Continuous Model

For a continuous pipe diameter formulation, the pipe cost function is generally fitted as:

$$c_f = bd^a \tag{4.1}$$

by using a set of commercially available discrete pipe sizes $D^0 = (d_1, d_2, \cdots, d_K)$ and a set of corresponding unit length costs $C = (c_1, c_2, \cdots, c_K)$, where a and b are fitted coefficients, and $a > 1.0$, $b > 0$ (for example a =1.24 and b =1.1 for the New York water supply tunnels problem by Schaake and Lai (1969). It is assumed that the fitted curve of Eq. (4.1) is a convex function (i.e. $a > 1.0$). The cost for a pipe n with a diameter d_n and a length L_n is:

$$C_f^n = L_n c_f(d_n) \tag{4.2}$$

While the Hazen-William equation is valid for the pipe flow, the pipe n with a continuous diameter d_n and length L_n, however, can be split into two segments with discrete diameters d_n^u and d_n^d where superscript u refers to the upstream segment and d to the downstream segment of pipe. The following constraint must hold $d_n^u \geq d_n \geq d_n^d$ such that the pipe of larger diameter is in the upstream segment. The lengths of the two segments between a pair of nodes are L_n^u and L_n^d, where $L_n = L_n^u + L_n^d$. The two lengths are found by an applying a head loss equation to maintain the same total head loss in the continuous diameter system and the split pipe system. One of the most common head loss formulas is Hazen Williams formula, given as:

$$h_{fn} = \frac{4.7291 L_n Q_n^{1.852}}{C_n^{1.852} d_n^{4.87}} \tag{4.3}$$

where Q_n is discharge in the pipe, and C_n is Hazen Williams coefficient for pipe n. The split pipe system should have the same head loss as the continuous one, that is

$$h_{fn} = h_{fn}^{u} + h_{fn}^{d} \qquad (4.4)$$

Solving equations (4.3) and (4.4), the split pipe lengths L_n^u and L_n^d are obtained:

$$L_n^u = \frac{(d_n^d)^{-4.87} - (d_n)^{-4.87}}{(d_n^d)^{-4.87} - (d_n^u)^{-4.87}} L_n \qquad (4.5)$$

$$L_n^d = L_n - L_n^u \qquad (4.6)$$

Thus the cost of the pipe *n*, split by Eq. (4.5) and Eq. (4.6) into two adjacent discrete pipe sizes, is given as:

$$C_s^n = L_n^u c_n(d_n^u) + L_n^d c_n(d_n^d) \qquad (4.7)$$

The optimal continuous pipe solutions (Bhave 1985; Fujiwara & Khang 1990) for the New York water supply tunnels problem have been evaluated by using the fitted continuous cost function of $c = 1.1d^{1.24}$ and its equivalent discrete split pipe cost. As shown in the Table 4-1, the network costs evaluated by the equivalent discrete split pipe cost are greater than that by fitted cost function used in the continuous models (Fujiwara and Khang 1990; Bhave 1985). Figure 4-1 shows a comparison of the cost of a continuous pipe solution (for sizes between 48 and 60 inches) represented by a fitted cost function $c_f = 1.1d^{1.24}$ versus the cost of the equivalent discrete split pipe solution (with an upstream segment diameter of 60 inches and a downstream segment diameter of 48 inches). It is evident that the cost of a continuous pipe is less than that of its equivalent discrete pipe. This has been proven for the general case of a continuous model (assuming a convex cost function) as follows.

Table 4-1 Comparison of Discrete Pipe and Continuous Pipe Costs

Cost evaluation method	Network cost (million $) New York water supply tunnels solutions	
	Bhave (1985)	Fujiwara & Khang (1990)
Fitted cost function	40.18	36.10
Equivalent split pipe cost	40.74	36.62
Percentage of difference	1.39%	1.44%

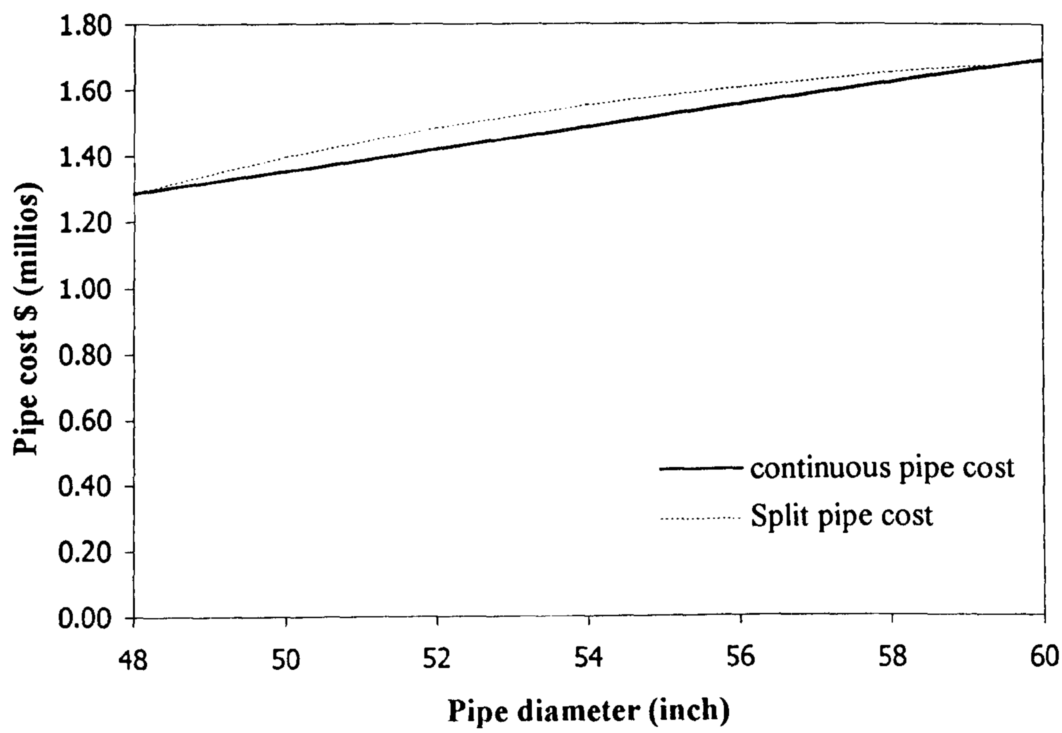


Figure 4-1 Comparing the Cost of a Continuous Pipe Solution to that of its Equivalent Discrete Split Pipe

Assuming $F = C_f^n - C_s^n$ (4.8)

where C_f^n is the cost of pipe n evaluated by the fitted cost curve given in Eq. (4.1) and Eq. (4.2), and C_s^n is the cost of pipe n by the split pipe cost evaluation given in Eq.(4.7). By introducing Eq. (4.5), (4.6) and (4.7) into Eq.(4.8)

$$F = \frac{L_n b}{(d_n^d)^{-4.87} - (d_n^u)^{-4.87}} \{ (d_n)^a[(d_n^d)^{-4.87} - (d_n^u)^{-4.87}] + (d_n^d)^a[(d_n^u)^{-4.87} - (d_n)^{-4.87}]$$
$$+ (d_n^u)^a[(d_n)^{-4.87} - (d_n^d)^{-4.87}] \} \quad (4.9)$$

The term on the right hand side prior to the term in curly brackets is always positive. Thus taking the portion of Eq. (4.9) in the curly brackets and letting

$$\Gamma(d_n^u, d_n, d_n^d) = (d_n)^a[(d_n^d)^{-4.87} - (d_n^u)^{-4.87}] + (d_n^d)^a[(d_n^u)^{-4.87} - (d_n)^{-4.87}]$$
$$+ (d_n^u)^a[(d_n)^{-4.87} - (d_n^d)^{-4.87}] \quad (4.10)$$

Now investigate the properties of the function Γ to find if it is an increasing or decreasing function by taking the first order derivatives and the second order derivatives as follows:

$$\frac{\partial \Gamma}{\partial d_n^u} = 4.87(d_n^u)^{-5.87}[(d_n)^a - (d_n^d)^a] + a(d_n^u)^{a-1}[(d_n)^{-4.87} - (d_n^d)^{-4.87}] \quad (4.11)$$

$$\frac{\partial \Gamma}{\partial d_n^d} = 4.87(d_n^d)^{-5.87}[(d_n^u)^a - (d_n)^a] + a(d_n^d)^{a-1}[(d_n^u)^{-4.87} - (d_n)^{-4.87}] \quad (4.12)$$

$$\frac{\partial^2 \Gamma}{\partial (d_n^u)^2} = 28.59(d_n^u)^{-6.87}[(d_n^d)^a - (d_n)^a] + a(a-1)(d_n^u)^{a-2}[(d_n)^{-4.87} - (d_n^d)^{-4.87}] \quad (4.13)$$

$$\frac{\partial^2 \Gamma}{\partial (d_n^d)^2} = 28.59(d_n^d)^{-6.87}[(d_n)^a - (d_n^u)^a] + a(a-1)(d_n^d)^{a-2}[(d_n^u)^{-4.87} - (d_n)^{-4.87}] \quad (4.14)$$

$$\frac{\partial^2 \Gamma}{\partial(d_n^d)\partial(d_n^u)} = 4.87a(d_n^u d_n^d)^{a-1}[(d_n^d)^{-a-4.87} - (d_n^u)^{-a-4.87}] \quad (4.15)$$

Consider the condition where the diameters of the upstream pipe and the downstream pipe are identical to the continuous pipe diameter, thus the solutions will be the same. In other words $d_n^u = d_n^d$, then $d_n^u = d_n^d = d_n$. This implies that there would be no cost difference between the continuous solution and the equivalent discrete split pipe solution and thus by Eq.(4.8)

$$F(d_n^d, d_n, d_n^u)\Big|_{d_n^d = d_n^u = d_n} = 0 \quad (4.16)$$

If we consider the form of the equation for the function Γ in Eq.(4.10) and the first order derivatives in Eq.(4.11) and Eq.(4.12) it is clear that if $d_n^u = d_n^d = d_n$ then these values must be zero as follows:

$$\Gamma(d_n^d, d_n, d_n^u)\Big|_{d_n^d = d_n^u = d_n} = 0, \ \frac{\partial \Gamma}{\partial d_n^u}\Big|_{d_n^d = d_n^u = d_n} = 0 \text{ and } \frac{\partial \Gamma}{\partial d_n^d}\Big|_{d_n^d = d_n^u = d_n} = 0 \quad (4.17)$$

Now considering the more general condition where the upstream and downstream discrete diameters are different such that $d_n^u > d_n > d_n^d$ and following a term by term analysis of Eq.(4.14) and Eq.(4.15) based on the assumption of $a > 1.0$, we have $\frac{\partial^2 \Gamma}{\partial(d_n^d)^2}\Big|_{d_n^d < d_n < d_n^u} < 0$ which implies $\frac{\partial \Gamma}{\partial d_n^d}$ is a decreasing function of d_n^d. It means that $\frac{\partial \Gamma}{\partial d_n^d}$ increases when d_n^d decreases, thus $\frac{\partial \Gamma}{\partial d_n^d}\Big|_{d_n^d < d_n} > \frac{\partial \Gamma}{\partial d_n^d}\Big|_{d_n^d = d_n}$. Referring to Eq.(4.17), it follows that the derivative:

$$\frac{\partial \Gamma}{\partial d_n^d}\Big|_{d_n^d < d_n} > 0 \quad (4.18)$$

Note that the value of $\frac{\partial^2 \Gamma}{\partial(d_n^d)\partial(d_n^u)}$ in Eq.(4.15) is greater than zero under the general condition $d_n^u > d_n > d_n^d$. In other words $\frac{\partial^2 \Gamma}{\partial(d_n^d)\partial(d_n^u)}\Big|_{d_n^d<d_n<d_n^u} > 0$, which implies that $\frac{\partial \Gamma}{\partial d_n^d}$ is an increasing function of d_n^u. This implies that the derivative $\frac{\partial \Gamma}{\partial d_n^d}$ increases as d_n^u increases, thus $\frac{\partial \Gamma}{\partial d_n^d}\Big|_{d_n^u>d_n} > \frac{\partial \Gamma}{\partial d_n^d}\Big|_{d_n^u=d_n}$. Referring to Eq.(4.17), it follows that the derivative:

$$\frac{\partial \Gamma}{\partial d_n^d}\Big|_{d_n<d_n^u} > 0 \tag{4.19}$$

Considering Eq.(4.18) and Eq.(4.19), the value of the first order derivative given by Eq.(4.12) is greater than zero in the entire range of (d_n^d, d_n^u), an example is shown in Figure 4-2 (based on $c_f = 1.1d^{1.24}$ as shown in Figure 1), and thus

$$\frac{\partial \Gamma}{\partial d_n^d}\Big|_{d_n^d<d_n<d_n^u} > 0 \tag{4.20}$$

This indicates that the function Γ given by Eq.(4.10) is an increasing function of d_n^d. It implies that Γ decreases when d_n^d decreases, thus $\Gamma\Big|_{d_n^d<d_n=d_n^u} < \Gamma\Big|_{d_n^d=d_n=d_n^u}$. Thus by Eq.(4.17), the following inequality holds

$$\Gamma(d_n^d, d_n, d_n^u)\Big|_{d_n^d<d_n=d_n^u} < 0 \tag{4.21}$$

Similarly by considering the general condition $d_n^u > d_n > d_n^d$ and a term by term analysis of Eq.(4.13) and Eq.(4.15) based on the assumption of a > 1.0, we have $\frac{\partial^2 \Gamma}{\partial(d_n^u)^2} < 0$. This implies that $\frac{\partial \Gamma}{\partial d_n^u}$ is a decreasing function of d_n^u. Thus the derivative

$\frac{\partial\Gamma}{\partial d_n^u}$ decreases as d_n^u increases, namely $\left.\frac{\partial\Gamma}{\partial d_n^u}\right|_{d_n=d_n^u} > \left.\frac{\partial\Gamma}{\partial d_n^u}\right|_{d_n<d_n^u}$. Referring to Eq.(4.17), it follows that

$$\left.\frac{\partial\Gamma}{\partial d_n^u}\right|_{d_n<d_n^u} < 0 \qquad (4.22)$$

Because $\left.\frac{\partial^2\Gamma}{\partial(d_n^d)\partial(d_n^u)}\right|_{d_n^d<d_n<d_n^u} > 0$ as given by Eq.(4.18), then $\frac{\partial\Gamma}{\partial d_n^u}$ is an increasing function of d_n^d, and the derivative $\frac{\partial\Gamma}{\partial d_n^u}$ decreases with decreasing d_n^d then it follows that $\left.\frac{\partial\Gamma}{\partial d_n^u}\right|_{d_n^d<d_n} < \left.\frac{\partial\Gamma}{\partial d_n^u}\right|_{d_n^d=d_n}$. Referring to Eq.(4.17), it can be concluded that:

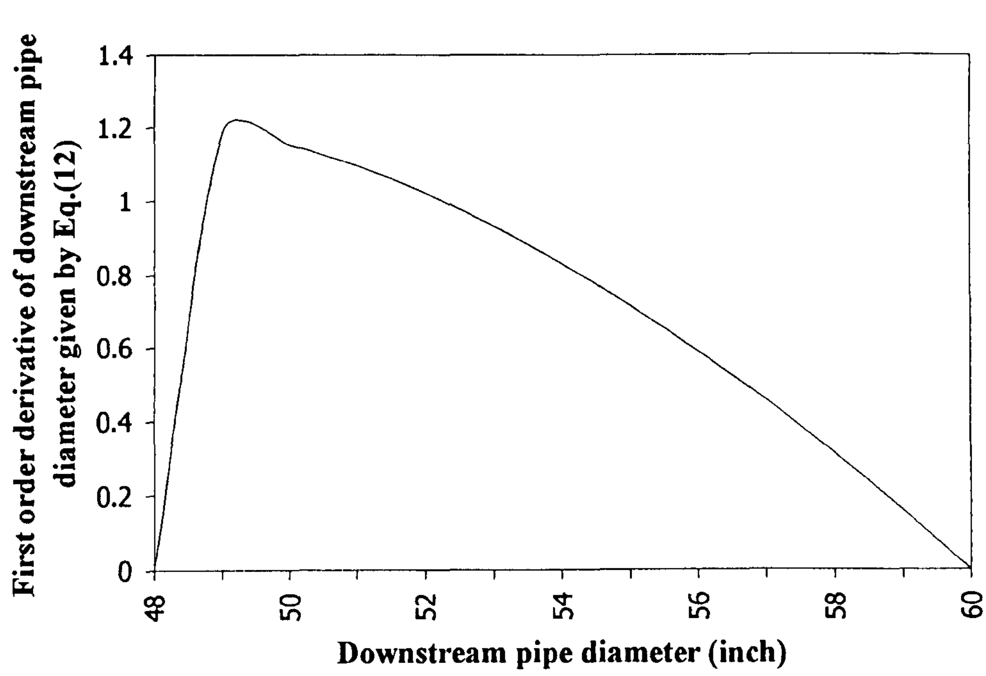


Figure 4-2 Plot of the First Order Derivative Given by Eq. (4.12)

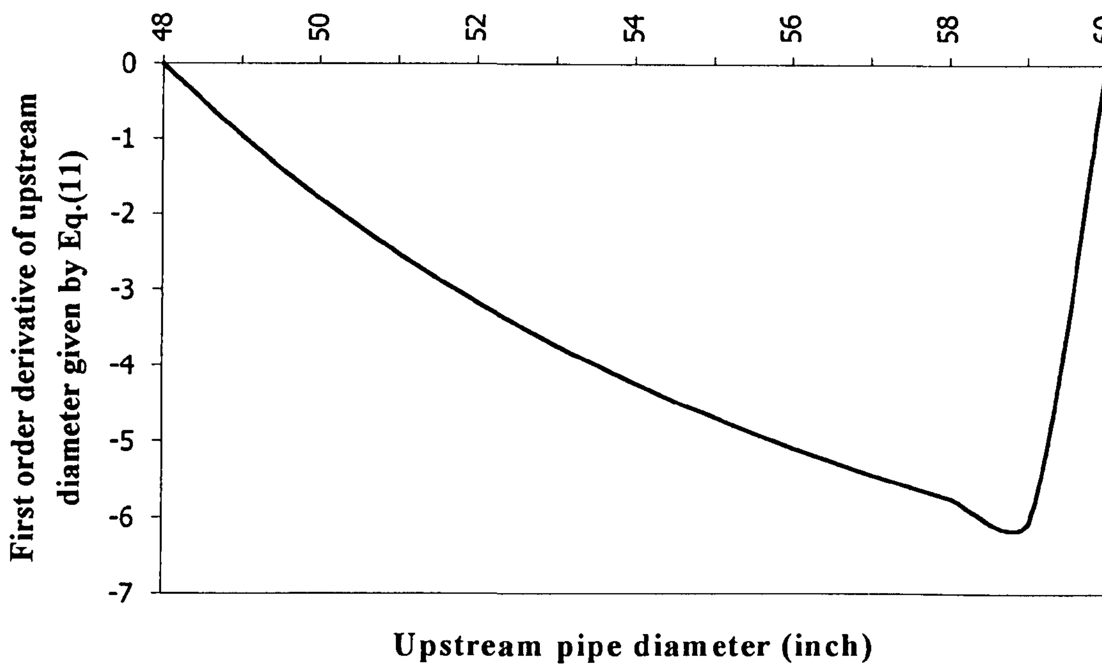


Figure 4-3 Plot of the First Order Derivative Given by Eq. (4.11)

$$\frac{\partial \Gamma}{\partial d_n^u}\bigg|_{d_n^d < d_n} < 0 \tag{4.23}$$

Considering Eq.(4.22) and Eq.(4.23), the value of the first order derivative by Eq.(4.11) is less than zero in the entire range of $(d_n^d,\ d_n^u)$, an example is shown in Figure 4-3 (again based on $c_f = 1.1d^{1.24}$ as shown in Figure 4-1), and thus

$$\frac{\partial \Gamma}{\partial d_n^u}\bigg|_{d_n^d < d_n < d_n^u} < 0 \tag{4.24}$$

This indicates that the function Γ given by Eq.(4.10) is a decreasing function of d_n^u. Then Γ decreases as d_n^u increases, and it follows that $\Gamma\Big|_{d_n^d = d_n < d_n^u} < \Gamma\Big|_{d_n^d = d_n = d_n^u}$. Referring to Eq.(4.17), the following inequality holds:

$$\Gamma(d_n^d, d_n, d_n^u)\Big|_{d_n^d = d_n < d_n^u} < 0 \tag{4.25}$$

Considering Eq.(4.21) and Eq.(4.25), we can now conclude that:

$$\Gamma(d_n^d, d_n, d_n^u)\Big|_{d_n^d < d_n < d_n^u} < 0 \tag{4.26}$$

Consequently, the value of the function F given by Eq.(4.9) is less than zero namely

$$F(d_n^d, d_n, d_n^u)\Big|_{d_n^d < d_n < d_n^u} < 0 \tag{4.27}$$

This indicates that the pipe cost evaluated by the continuous fitted cost function is less than by the discrete split pipe cost namely $C_f^n < C_s^n$. In other words, the pipe cost increases when converting a continuous solution to a hydraulically equivalent split pipe solution. It is, therefore, concluded that there is no guarantee that the continuous model using a fitted convex cost function (i.e. $a > 1.0$ in Eq.(4.1)) will reach an optimal split pipe solution.

4.3 Formulation of a Split Pipe Model

The split pipe model is formulated (1) to select pipe diameter sizes $\vec{D}=\{d_n, n=1,..., N\}$, where N = number of all new pipes added to a water system, and (2) to choose rehabilitation actions (e.g. duplicating or replacing a pipe or cleaning a pipe) and the associated pipe sizes $\vec{E}=\{e_r, d_r, r=1,...R\}$, where R = number of all possible rehabilitated pipes. The pipe diameters d_n and d_r take on the continuous values in an interval between the minimum and maximum pipe sizes namely d_n , $d_r \in [d^{min}, d^{max}]$, where d^{min} = minimum diameter of pipe sizes; and d^{max} = maximum diameter of pipe sizes. The cost of a new pipe or a rehabilitated pipe is evaluated by split pipe sizes given by Eq. (4.3) to (4.6). The objective of the optimisation procedure is to minimise the total cost of the network pipeline materials and also installation and rehabilitation actions subject to minimum allowable pressure heads at nodes. Thus the split pipe model may be expressed as:

search for $(\vec{D}, \vec{E})$

minimise $$C(\vec{D}, \vec{E}) = \sum_{n=1}^{N}[c_n(d_n^d)L_n^d + c_n(d_n^u)L_n^u] +$$

$$\sum_{r=1}^{R}[L_r^d c_r(d_r^d, e_r) + L_r^u c_r(d_r^u, e_r)]$$

subject to $$d^{\min} \le d_n \le d^{\max}$$

$$\forall d_n^d, d_n^u, d_r^d, d_r^u \in D^0 = \{d_k^0, k = 1, \ldots, K\}$$
$$\forall e_r \in E^0 = \{e_m^0, m = 1, \ldots, M\}$$
$$H_{ij} \ge H_j^{min}, j = 1, \ldots, J, \quad i = 1, \ldots, I$$
$$L_n = L_n^d + L_n^u$$
$$L_n(d_n)^{-4.7804} = L_n^d(d_n^d)^{-4.7804} + L_n^u(d_n^u)^{-4.7804}$$

where D^0 = the set of commercially available pipe sizes; d_k^0 = k-th commercially available pipe diameter in set D^0; K = number of the commercially available pipe sizes; E^0 = the set of possible rehabilitation events; e_m^0 = m-th rehabilitation event that may applied to the existing pipe r in set E^0; M = number of the rehabilitation events applicable to pipes; H_{ij} = hydraulic grade at node j under steady state loading case i; J = number of nodes (excluding fixed grade nodes); I = number of steady state loading cases; $H_j^{\min}$ = minimum allowable hydraulic grade at node j. One advantage of the split pipe model is that it eliminates the need for fitting a cost function to discrete values. The fitted cost function may lead the search process to a lower cost continuous solution than that by discrete formulation, but the continuous solution is not desirable for practical application in most of cases.

4.4 Genetic Algorithm Mapping Scheme for Split Pipe Model

The total length of the chromosome is divided into N intervals of the same length dd for a new pipe and R intervals of the same length (ee + dd) for a rehabilitated pipe. Thus each genotype $\vec{g}_{nn}(t)$ has a general form as:

$$\vec{g}_{nn}(t) = (a^d_{1,1},\ldots,a^d_{1,dd},\ldots,a^d_{N,1},\ldots,a^d_{N,dd},a^e_{N+1,1},\ldots,a^e_{N+1,ee},a^d_{N+1,ee+1},\ldots,a^d_{N+1,ee+dd},$$

$$\ldots,a^e_{N+R,1},\ldots,a^e_{N+R,ee},a^d_{N+R,ee+1},\ldots,a^d_{N+R,ee+dd}) \tag{4.28}$$

$a^d_{n,dex}$ is a binary bit, taking the value of either 1 or 0 for a binary coding, and represents pipe diameter, where n = 1, ..., N+R, N = the number of new pipes to be added to a distribution system; R = the number of existing pipes to be rehabilitated; dex = 1, ..., dd, where dd = the number of bits representing the diameter of an expanded or rehabilitated pipe. Similarly, $a^e_{r,edex}$ represents rehabilitation event, where r = N+1, ..., N+R, $edex$ = 1, ..., ee, where ee = the number of bits representing a rehabilitation action for an existing pipe. The number of bits coding a pipe undergoing rehabilitation is the sum of the bits for coding the pipe diameter and the bits for coding the rehabilitation action.

Each chromosome, the genotypical individual $\vec{g}_{nn}$ with the length of $dd(R+N)+R(ee)$ bits, defines a corresponding phenotypical representation or a phenotype namely one design alternative given as:

$$\vec{p}_{nn} = (\vec{D},\vec{E}) = (d_1,\ldots,d_N,e_1,d_{N+1},\ldots,e_R,d_{N+R}) \tag{4.29}$$

a vector of pipe diameters for new pipes and associated rehabilitation actions for existing pipes.

Generally, the genotypical pipe size representation ($a^d_{n,1},a^d_{n,2}\ldots a^d_{n,dd}$) is uniformly mapped into a prescribed interval [d^{min}, d^{max}], given as:

$$d_n = d^{\min} + \frac{d^{\max} - d^{\min}}{2^{dd} - 1}[\sum_{dex=1}^{dd} a^d_{n,dex} 2^{dex-1}] \quad (4.30)$$

Each rehabilitation policy genotypical representation $(a^e_{r,1}, a^e_{r,2} \dots a^e_{r,ee})$, where $r = N+1, \dots, N+R$, for an existing pipe, is mapped into a phenotypical representation e_r by the mapping scheme developed in Chapter 3. The messy genetic algorithm described in the last Chapter is utilised here, however, the results for a comparison of the split pipe and discrete pipe formulation which is the focus of this Chapter are equally valid for a standard genetic algorithm formulation.

4.5 Integrating the Split Pipe Model into Messy Genetic Algorithm

The split pipe optimisation model has been implemented in a genetic algorithm formulation. The cost evaluation of a split pipe design in a genetic algorithm formulation is given in Figure 4-4. For each string from the GA, the cost evaluation begins by extracting the genes for each rehabilitated pipe and decodes the genes as a rehabilitation event and a pipe size for each existing pipe, and then extracts the genes for each new pipe and decodes the genes as the new pipe diameters. The evaluation is followed by updating friction coefficients due to the rehabilitation such as cleaning and lining of an existing pipe, and by assigning a set of diameters to diameter variables for a hydraulic network simulation. The network solver is called for each demand loading case, and the maximum hydraulic pressure (or grade) deficit, found for all the demand cases, is used for the penalty cost computation. The total cost of one solution is the sum of the network cost and the penalty cost and also is converted to the fitness of the genotypical string as given as in Chapter 3.

```
Begin
for each string in a population for the genetic algorithm begin
        for each existing pipe begin
                extract gene for rehabilitation policy;
                decode the policy;
                extract gene for rehabilitation pipe size;
                decode the size;
                        end
        for each new pipe begin
                extract gene for pipe size;
                decode the size;
                        end
        computing split pipe sizes;
        update coefficients and pipe sizes;
        for each demand case begin
                perform network hydraulic simulation;
                find maximum hydraulic pressure deficit;
                                end
        penaltycost = maxpressuredeficit *penaltyfactor;
        networkcost = new split pipe cost + rehabilitation split pipe cost;
        totalcost = penaltycost + networkcost;
        end
end
```

Figure 4-4 Pseudo-code for Cost Evaluation of Split Pipe Model

Two problems, a two reservoir network problem and the New York City water supply tunnels problem, have been chosen for testing the GA based split pipe model for optimal rehabilitation and pipe-sizing of water supply systems. Both problems have been studied by using the discrete model and the messy GA in Chapter 3 and are good examples for comparing the results from the split pipe optimisation model with the discrete pipe optimisation model.

4.6 Case Study I: The Two Reservoir Network

4.6.1 Genetic algorithm coding and parameters

In the application of the split optimisation model to a two reservoir network, the pipe sizes have been coded and mapped into a prescribed interval. Suppose that a *dd*-bit string is used to represent a pipe size, 2^{dd} points, which are uniformly distributed within the interval between the minimum discrete size of 152 mm and the maximum discrete size of 509 mm, are selected to form the search space for GA optimisation. Each *dd*-bit sub-string for one pipe is extracted from a genotype, and decoded into an integer, then the integer is mapped to a real number belonging to the range [152, 509] by Eq.(4.30). Thus a genotype has been mapped to a phenotype in split pipe genetic algorithm optimisation formulation. The other messy genetic algorithm parameters taken as different values compared with the discrete formulation in Chapter 3 are a random seed of 0.9, a juxtapositional size of 300 and the number of maximum generations of 100.

4.6.2 Results and comparison for the two reservoir network

During the application of this model to the two reservoir network, a number of genotype representations with different number of bits for representing the pipe size have been used and the results are compared in Table 4-2. The more bits that are used for representing each pipe, the more points in the interval are picked to form the search space, the more evaluations that are required to search for the lower cost solutions. The best split pipe solution ($1.7145 million) was found with a 9-bit representation. It is 2% cheaper than the discrete optimal solution ($1.7503 million) by Simpson et al. (1994). The solutions, however, found with the other representations are not very different in terms of the cost except for the 4-bit solution. It can be concluded that the split pipe optimisation model gives a lower cost solution than the discrete pipe formulation.

The hydraulic pressure deficits or differences between actual hydraulic pressures and minimum allowable pressures at nodes are given in Table 4-3 for the split pipe solution ($1.7145 million). It is shown that the optimum is controlled by demand case 2. Two nodes, node 4 and node 7, are critical for the optimal design solutions. The characteristics that the optimum solution is governed by certain critical nodes were also observed in a previous GA application (Simpson et al. 1994) to optimisation of water distribution networks.

Figure 4-5 shows the convergence behaviour of messy GA with the 9-bit representation for split pipe optimisation of the two reservoir network. It is observed that the convergence behaviour of the messy GA is similar to that of standard GA applications (Simpson et al. 1994; Wu 1994; Babovic et al. 1994 and Dandy et al. 1996), i.e. the GA quickly detects the optimal solution region but takes a long search process to reach the optimal solution.

Table 4-2 Comparison of the Best Split Pipe Solutions with Different Lengths of Genotype Representation

	Number of bits for representing each pipe size													
	4 bits		5 bits		6 bits		7 bits		8 bits		9 bits		10 bits	
Pipe	Dia. (mm)	Length (m)	Dia. (mm)	Length (m)	Dia. (mm)	Length (m)	Dia. (mm)	Length (m)	Dia. (mm)	Length (m)	Dia. (mm)	Length (m)	Dia. (mm)	Length (m)
[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[4]	356	6437	305	546	305	827	305	820	305	817	305	778	305	796
			356	5891	356	5610	356	5617	356	5621	356	5659	356	5642
[5]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[6]	254	361	305	1609	254	75	254	164	305	1609	254	182	305	1609
	305	1248			305	1534	305	1445			305	1247		
[8]	203	1421	152	66	203	1609	203	1410			203	1609	152	36
	254	188	203	1543	254		254	199	203	1609			203	1574
[11]	203	115	203	173	203	1227	203	1609			254	1609	203	741
	254	1494	254	1436	254	382			203	1609			254	867
[13]	152	1609	152	1609	152	1609	152	1503	152	1609	152	1609	152	1609
							203	106						
[14]	203	1421	203	1516	203	427	203		254	1609	203	1609	203	970
	254	188	254	93	254	1182	254	1609					254	639
Cost ($m)	1.7448		1.7261		1.7164		1.7201		1.7206		1.7145		1.7166	
Evaluation	3,254		10,478		12,453		20,654		23,167		30,803		38,972	

Table 4-3 Hydraulic Pressure Deficit for Lowest Cost Split Pipe Solution ($1.7145 million) for the Two Reservoir Network

Node ID	Pressure Surplus (m)*		
	Load case 1	Load case 2	Load case 3
2	**7.32****	9.64	15.35
3	11.98	3.80	9.31
4	7.99	**0.01**	4.65
6	10.91	3.89	19.93
7	13.92	**0.03**	22.70
8	21.28	22.33	29.56
9	16.98	10.06	23.34
10	15.99	9.15	19.53
11	16.35	15.81	16.39
12	17.64	17.11	**0.45**

*A negative value indicates pressure deficit. **Bold values indicates critical nodes.

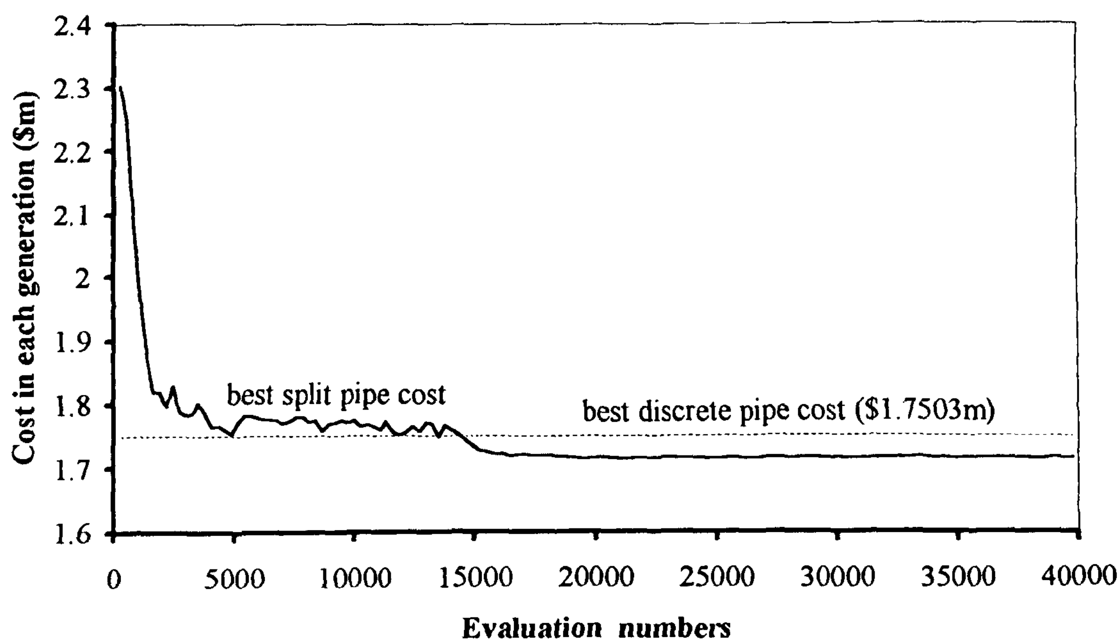


Figure 4-5 Optimal Cost for the Split Pipe Solution $1.7145 Million by mGA —Two Reservoir Network

4.7 Case Study II: New York City Water Supply Tunnels

The New York water supply tunnels problem was posed by Schaake and Lai in 1969 to select the optimal combination of pipe sizes. It has been studied by a number of researchers in literature. More recently, Loganathan et al. (1995) proposed an outer flow search—inner optimisation procedure for choosing a better local optimal. The outer search scheme used ANNEALING and/or MULTISTART to choose alternative flow configurations to find an optimal flow division among pipes. An inner linear program was used for the design of the least cost diameters. The approach yielded a lower cost split pipe solution of $38.04 million for New York water tunnels than results from previous studies up to date. This study together with the improved GA (Dandy et al. 1996) and the messy GA application in the last Chapter provides a sound basis for comparison with the genetic algorithm split pipe formulation proposed in this Chapter.

4.7.1 Results and comparison for the New York city tunnels problem

In order for the GA to search for the best combination of pipe sizes, an 8-bit sub-string was used to code the pipe diameter sizes for each of the 21 pipes. Thus a total of 168 binary bits were required for this problem by the genetic algorithm split pipe formulation. Each sub-string is mapped into an interval between 36 and 204 inches. Thus 256 points (pipe size alternatives) uniformly distributed in the interval [36, 204] have been used to search for the optimal split pipe solution. The pipe diameter choices are separated by 0.6588 inches ((204-36)/255).

The GA split pipe solution and recently published results are summarised and compared in Table 4-4. The split pipe model has found a lower cost solution than the improved GA (Dandy et al. 1996) and the outer flow search—inner optimisation approach (Lognathan et al. 1995). As shown in Table 4-4, the cost of split pipe design solution is $1.07 million lower than the improved GA solution by Dandy et al. (1996) and $0.31 million lower than the split pipe solution by Loganathan et al. (1995). The convergence rate of the GA based split pipe formulation, as shown in Figure 4-6, indicates that to reach a similar cost level to that previously obtained (Dandy et al. 1996 and Lognanathan et al. 1995) the GA split pipe model only requires about 30,000 network evaluations. This number is approximately one third of the evaluations (96,750) required by the improved GA discrete pipe optimisation as shown in Table 4-4. Thus the genetic algorithm split pipe formulation provides an efficient search procedure for reaching the low cost solution. It is also shown in Table 4-5 that the genetic algorithm split pipe solution is closer to the critical constraint boundary than other solutions.

Table 4-4 Comparing the Genetic Algorithm Split Pipe Design with Previous Solutions

Pipe	Discrete pipe solutions		Split pipe solutions			
	Dandy et al. * (1996).		Lognanathan** et al. (1995)		Messy GA**	
	Dia. (in)	Length (ft)	Dia. (in)	Length (ft)	Dia. (in)	Length (ft)
[7]	0	-	120	8902	96	867
[7]	0	-	132	698	108	8733
[15]	120	15500	0	-		
[15]			0	-		
[16]	84	26400	96	20476	84	697
[16]			108	5924	96	25703
[17]	96	31200	96	30986	96	17646
[17]			108	214	108	13544
[18]	84	24000	84	23247	72	4988
[18]			96	753	84	19012
[19]	72	14400	72	13687	48	1635
[19]			84	713	60	12765
[21]	72	26400	72	26291	72	5487
[21]			84	109	84	20913
Cost ($million)	38.80		38.04		37.73	
Evaluation	96,750		-		266,215	

Hazen-Williams formulation:

$* h_f = 4.7291 L \left(\frac{Q}{C}\right)^{1.852} D^{-4.8704}$; $** h_f = 4.73 L \left(\frac{Q}{C}\right)^{1.852} D^{-4.87}$

Table 4-5 Comparison of Hydraulic Heads of Discrete and Split Pipe Optimal Solutions for New York City Tunnels Problem

Node	Minimum required head (ft)	Discrete pipe solution	Split pipe solutions	
		Dandy et al. (1996)	Lognanathan et al. (1995)	Messy GA
16	260.00	260.65	260.06	260.09
	Pressure surplus =	0.65	0.06	0.09
17	272.80	272.98	272.83	272.8
	Pressure surplus =	**0.18***	**0.03**	**0.00**
19	255.00	255.84	255.06	256.12
	Pressure surplus =	0.84	0.06	1.12
Cost ($million)		38.80	38.04	37.73

*critical node in bold

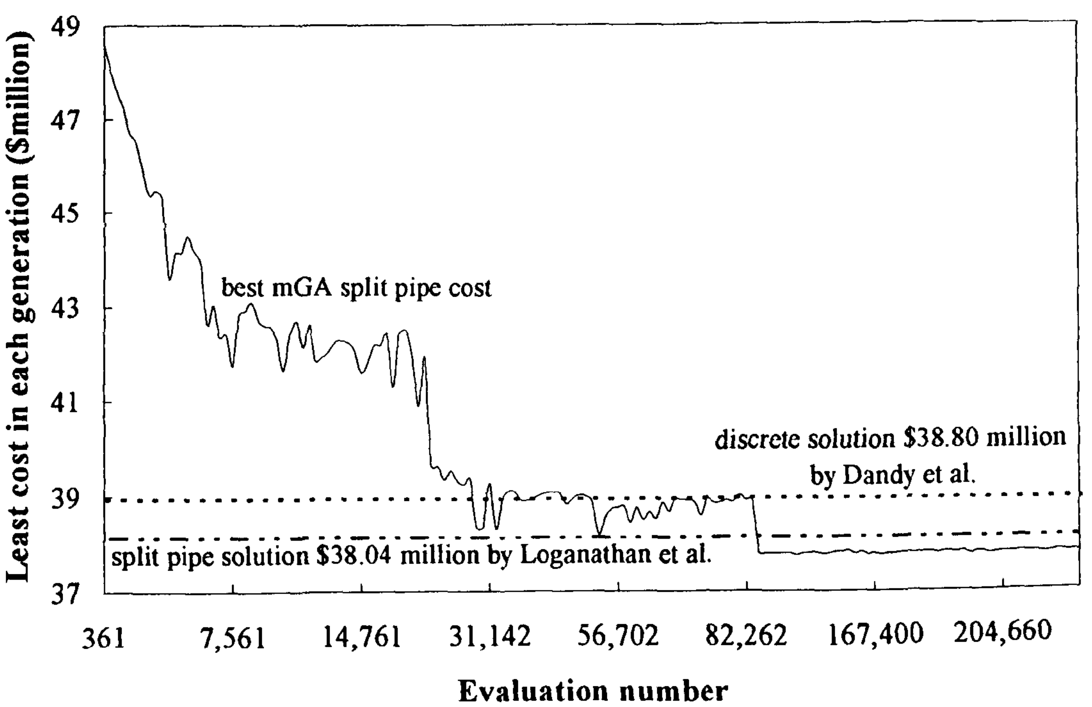


Figure 4-6 Optimal Cost for the Split Pipe Solution $37.73 million by mGA

4.8 Summary

A split pipe formulation for the genetic algorithm optimisation of water distribution systems has been developed. An analysis showing a comparison of the continuous pipe formulation and the split pipe formulation has been presented and indicates that there is no guarantee that the continuous pipe size model using a fitted convex cost function will reach an optimal split pipe solution. The split pipe optimisation model, which allows the split pipe sizes to be used in the optimisation procedure, has been formulated to search for the optimal split pipe solution. This model has been implemented in a genetic algorithm formulation and tested on two previously studied examples of optimisation of water supply networks. The results have shown that the genetic algorithm split pipe model for optimal sizing and rehabilitation of water distribution networks is able to give lower cost solutions than genetic algorithm discrete pipe optimisation. The number of evaluations required for the split pipe optimal solution is greater than that for the discrete pipe approach because more gene bits have been used for representing the solution space of split pipe optimisation.

To develop a comprehensive methodology for optimal rehabilitation of water distribution systems, current optimisation models need to be extended to consider other components such as pump stations. This will be presented in the next Chapter.

5. OPTIMAL REHABILITATION OF WATER DISTRIBUTION SYSTEMS USING A MESSY GENETIC ALGORITHM

5.1 Introduction

Water distribution systems have existed for more than a hundred years in industrialised countries. As these systems age and economies develop, it is essential that the water supply infrastructure is continuously improved (rehabilitated) to meet the current and future demands of both water quantity and water quality. The funds available for rehabilitation of the water supply systems have increased in recent years, but are always limited. Thus developing a technique for achieving an optimal rehabilitation of the water distribution system is of great importance to ensure maximum benefits for every dollar expended.

In this Chapter, a messy genetic algorithm (mGA) is employed as a search algorithm and integrated with a hydraulic solver to search for the optimal solution. The messy genetic algorithm is a search algorithm, as introduced in Chapter 3, uses a population of variable-length chromosomes and mimics the principles of biological reproduction. The approach is applied to a case study network in this Chapter to demonstrate the application of the technique.

5.2 An Overview of Rehabilitation

A water supply system needs rehabilitating when it reaches the stage where the system cannot deliver adequate service. The objective is to satisfy the consumer's demands in terms of quantity and quality of water supplied. Thus rehabilitation of water distribution systems is motivated by a combination of social, economical and political reasons. It is not a single activity but involves numerous decisions and activities, which transform an inadequate water system into one that provides the quality of the service required by the consumers. The decision-making process regarding the best strategy for rehabilitation may take into account all the system components such as water supply dams, water treatment plants, wells, storage tanks, pump stations, distribution pipelines and control valves. However, it is not practical to consider all the facets in one decision-making stage. The development of a computer-based optimisation tool for rehabilitation of water distribution pipelines and pumps on which a large amount of capital investment has been spent (Walski et al. 1987 will constantly enhance the capability and efficiency of the decision-making process.

The rehabilitation decisions relating to the preferred strategy for a water distribution system including pump stations can be undertaken as a search problem. A rehabilitation action for each pipe is selected from a set of possible rehabilitation actions such as cleaning, relining, replacing, duplicating the pipe (laying a pipe parallel to the existing pipe) or just leaving a pipe as it is. A pipe size associated with the action is chosen from a list of commercially available discrete pipe sizes. In addition, the pump capacity is determined for each pump. The optimal rehabilitation strategy is the combination of the decisions for rehabilitation of the pipeline and pumps that results in the minimum overall cost. Although Walski et al. (1987) showed that the methods developed for optimisation of the design of a water distribution system could be used to solve the problem of optimal rehabilitation of water distribution systems, it is difficult to consider all the discrete variables (such as rehabilitation action for each pipe) and the continuous decision variables efficiently by using traditional linear and/or non-linear programming formulations. Kim and Mays (1994) developed a mathematical formulation for the optimal rehabilitation of water distribution systems using a mixed-integer non-linear programming approach. An implicit enumeration procedure called the branch and bound algorithm was used to find the optimal rehabilitation actions for the pipeline while non-linear programming was used to size the pipes and pumps. In Chapter 3 and Chapter 4, the application of a messy genetic algorithm (mGA) (Goldberg et al. 1989) to water distribution system optimisation has shown that the mGA provides an efficient approach within the genetic algorithm paradigm. Thus the mGA is employed to solve the optimal rehabilitation problem in this Chapter. A case study is also presented to demonstrate the application of this technique.

5.3 Problem Formulation

The objective of the optimisation model for rehabilitation is to minimise the sum of the present worth of the costs which includes the pipeline rehabilitation cost, the expected pipe repair cost and the pump energy cost. The total cost is minimised by searching for an optimal rehabilitation strategy for pipelines and an optimal pump operational capacity (assumed to be a constant pump horsepower) subject to constraints such as pressure head requirements for all demand nodes and pipe size limitations.

5.3.1 Pipeline rehabilitation

Pipeline rehabilitation involves choosing a rehabilitation action e_r from a set of possible rehabilitation actions (e.g. duplicating or replacing a pipe or cleaning a pipe) and an associated pipe size d_r for rehabilitated pipe r. The cost of a pipe rehabilitation is a function of the rehabilitation action and the associated diameter (e_r, d_r), where $r = 1, 2, 3, \ldots, R$. It is assumed that the cost for one pipe is a linear function of the cost per unit length of the pipe with rehabilitation action e_r and diameter d_r. Thus the cost function of the pipeline rehabilitation is given as:

$$C_{reh} = \sum_{r=1}^{R} c_r(e_r, d_r)L_r \tag{5.1}$$

where R = number of rehabilitated pipes; L_r = length of rehabilitated pipe r; d_r = diameter of rehabilitated pipe r; e_r = rehabilitation action taking place at pipe r (eg. removing, replacing, duplicating or cleaning) and $c_r(e_r, d_r)$ = cost of unit length of rehabilitated pipe r with diameter d_r and event e_r.

If no rehabilitation action is taken for a pipe, a cost may occur for repairing a pipe break. Breaks are generally repaired by placing a sleeve around the break. The cost for repairing the pipe break is a function of many factors such as the crew, equipment, sleeve details, repaving and the associated overheads. A unit cost (dollars/break) of pipe break repair has been derived as a function of pipe diameter by the U.S. Army Corps of Engineers (1983). Thus the present worth of the expected cost of a pipe repair in its design life can be formulated by introducing a break rate P_b (breaks/mile/year), and is given as:

$$C_{rep} = \sum_{n=R+1}^{N} \sum_{y=1}^{Y} \left[\frac{c_{rep}(d_n)P_b L_n}{(1+r)^y \, 5280} \right] \tag{5.2}$$

where N = the number of the pipes to be expected breaking; Y = planning period in years; P_b = the expected break rate of the unit length pipe n. The break rate P_b can be posed in many ways, but the same break rate formulation as Kim and Mays (1994) is used. It was originally given as a function of the pipe diameter by Goodrich et al. (1989) as follows.

$$P_b = 0.819e^{-0.1353 d_n} \tag{5.3}$$

where d_n is the diameter of pipe n. Thus the pipeline cost is the sum of the rehabilitation cost and the pipe break repair cost namely

$$C_{pipe} = C_{reh} + C_{rep} \tag{5.4}$$

The pipeline cost C_{pipe} is minimised subject to the pressure head requirement that the pressure head h_{ij} being supplied at each demand node j under loading case i must be greater than or equal to the minimum allowable pressure head H_j^{min} and less than or equal to the maximum pressure head H_j^{max}, given as:

$$H_j^{min} \le h_{ij} \le H_j^{max} \qquad (5.5)$$

5.3.2 Pump rehabilitation

Assume that a pump has a typical characteristic curve such as that given in the MAPS manual (U.S. Army Corps of Engineers 1980). The useful horse power HP_k for each pump is used as a decision variable when considering rehabilitation of the pump stations. The present worth of the energy cost for a pump can be obtained by summing the energy cost of each pump in the operation period in its design life (in years), and given as (Kim and Mays 1994):

$$C_{pump} = \sum_{k=1}^{K} \sum_{y+1}^{Y} \left\{ \left[\frac{365*0.746HP_k}{(1+r)^y E_k} \right] \sum_{t=1}^{T_n} \left[M_k \Delta T_{tk} \right] \right\} \qquad (5.6)$$

where M_k = the unit cost (dollars/kwh) of electricity for pump k; and E_k = the efficiency of pump k; ΔT_{tk} = the time of pump k in operation period t during one day. The horse power HP_k cannot be smaller than an existing horse power $\underline{HP_k}$ provided by the existing pumps, that is

$$HP_k \ge \underline{HP_k} \qquad (5.7)$$

which gives the lower bound of the pump decision variable HP_k. Thus the optimal rehabilitation of water distribution systems is formulated as follows:

Search for: (e_r, d_r) and HP_k

Minimise: $(C_{pipe} + C_{pump})$

Subject to: $H_j^{min} \le h_{ij} \le H_j^{max}$

$$HP_k \geq \underline{HP_k}$$

In order to find the least cost rehabilitation strategy for the pipelines and pump stations in the water distribution system, an integrated approach and the computer program (mGANET) developed in Chapter 3, which coupled a hydraulic network solver EPANET with a messy genetic algorithm (mGA), has been employed for optimisation of rehabilitation of water distribution systems.

5.4 A Case Study

A hypothetical network, as shown in Figure 5-1, was studied by Kim and Mays (1994). It provides a good example for demonstrating the application of the messy genetic algorithm approach described in the previous section. The results of this Chapter are compared with the previous results of Kim and Mays (1994).

The network consists of 17 pipes and 12 nodes as shown in Figure 5-1. The Hazen-Williams roughness coefficients in all the pipes are assumed to be 50 (indicating severely encrusted pipe interiors) in order to simulate an existing system that does not adequately meet the water demand and the minimum allowable pressure head requirement. The Hazen-Williams coefficient in a new pipe (replaced pipe) is assumed to be 130, while that of a rehabilitated pipe is 100. We chose to present the case study in SI units although previous work is in US customary units. Table 5-1 gives the characteristics of the pipes in the network while Table 5-2 lists the demands and allowable pressure heads at the nodes. The unit length costs given in Table 5-3 for each possible rehabilitation action and associated diameter were obtained by the fitted cost functions by Kim and Mays (1994). The values of other parameters are given as (Kim and Mays 1994) (1) the planning period of the pump and pipe reparation is 20 years; (2) unit cost of electricity is 0.05 dollars/kWh; (3) pump efficiency 75% and (4) interest rate 6%. . These values were selected to ensure that the results in this study are comparable to the previous results.

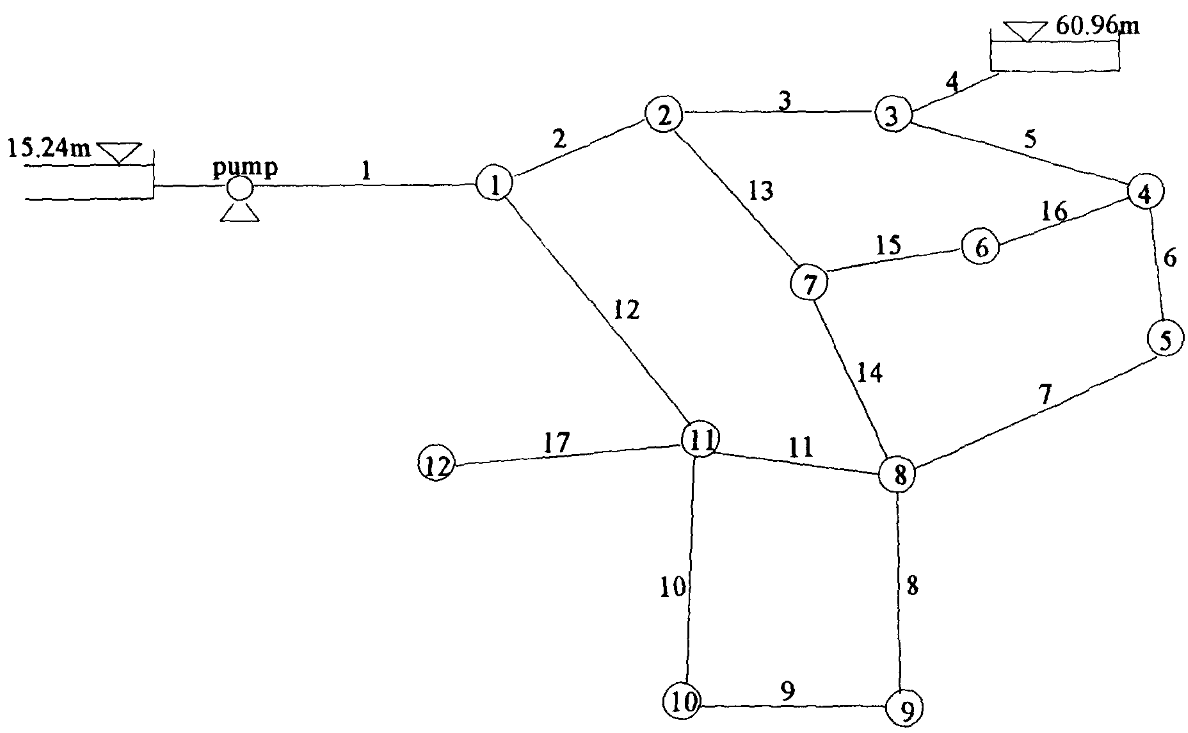


Figure 5-1 Layout of the Water Distribution System *(from Kim and Mays 1994)*

Table 5-1 Characteristics of Pipes in the Case Study Network

Pipe No.	Length (m)	Diameter (mm)	Roughness Coeff. (C)
1	3048	610	50
2	1524	457	50
3	1524	457	50
4	305	152	50
5	1676	381	50
6	1067	381	50
7	1676	381	50
8	1372	305	50
9	762	229	50
10	1067	305	50
11	671	305	50
12	1981	381	50
13	1524	381	50
14	1676	381	50
15	914	381	50
16	1219	381	50
17	1219	229	50

Table 5-2 Characteristics of Nodes Used in the Case Study Network

Node No.	Elevation (m)	Demand (L/s)	Allowable pressure head (kPa)			
			min. (kPa)	min. (m)	max. (kPa)	max. (m)
1	46	0	-	-	-	-
2	49	142	276	28	827	84
3	50	0	-	-	-	-
4	49	121	276	28	827	84
5	46	105	276	28	827	84
6	47	85	276	28	827	84

Node No.	Elevation (m)	Demand (L/s)	Allowable pressure head (kPa)			
			min. (kPa)	min. (m)	max. (kPa)	max. (m)
7	44	142	276	28	827	84
8	43	0	-	-	-	-
9	40	57	276	28	827	84
10	41	57	276	28	827	84
11	44	0	-	-	-	-
12	40	57	276	28	827	84

5.4.1 Messy genetic algorithm coding and decoding

A binary string coding has been used for this study. Each bit of a string takes one value of either 1 or 0. Three binary bits have been used to represent each pipe size variable of the 17 existing pipes to represent 8 possible choices of pipe sizes. Two binary bits have been used for each existing pipe to represent 3 possible choices of rehabilitation actions that include relining, replacing or leaving as it is (one coding is not used). A pipe breakage repair cost is expected to occur for a pipe taking the action of leaving it as it is. Four bits are used for representing the useful pump horsepower. Thus a total of 89 bits are needed for solving the problem by using the formulation described earlier. A binary coding and decoding scheme for the all possible pipe sizes and rehabilitation actions of the network are given in Table 5-4 and Table 5-5. The penalty factor for pressures which do not meet the allowable pressure constraint for this problem was chosen to be $1,524,000/m (assumed originally to be $5,000,000/ft prior to conversion to SI units) of deficit.

The mapping for this problem follows the mapping scheme given in Chapter 3. An example of mapping for the case study network is given in Table 5-6. A string, for example 1011100011011010l1 01 ......01111, is first divided into 17 substrings of 5 bits (2 bits for coding the rehabilitation action and 3 bits for coding of the associated diameter) for each of the existing pipes. The action and diameter binary strings are treated as binary numbers and converted to 10 decimal integers which are used as indexes. The sizes or diameters of the pipes rehabilitated were found from Table 5-4 by the mapped indexes. The size of the pipe for taking the action of 'leaving' was the same as the original size. Similarly, the rehabilitation action for each existing pipe was found from Table 5-5 by the mapped index. The network cost of the solution can be calculated by the pipe diameter sizes and the rehabilitation actions. The penalty cost is calculated by calling the hydraulic solver EPANET to determine any pressure deficits. It is the penalty cost that degrades the fitness of the string.

Table 5-3 Unit Length Cost of Rehabilitation Action and Associated Diameter

Diameter	Possible Rehabilitation Actions for Pipes		
(mm)	Relining ($/m)	Replacing ($/m)	Repairing ($/m)
152	82	136	2775
203	82	152	3317
229	82	161	3568
305	82	191	4265
381	82	226	4898
457	82	266	5484
508	84	294	5854
610	87	356	6555

Table 5-4 Coding and Decoding Scheme of Available Pipe Sizes

Pipe diameter (mm)	Binary substring corresponding to the pipe size	Pipe size index	Pipe size notation
152	000	0	d_0^0
203	001	1	d_1^0
229	010	2	d_2^0
305	011	3	d_3^0
381	100	4	d_4^0
457	101	5	d_5^0
508	110	6	d_6^0
610	111	7	d_7^0

Table 5-5 Coding and Decoding Scheme of Possible Rehabilitation Actions

Actions	Binary string	Index	Rehabilitation action notation
Leaving a pipe	00	0	e_0^0
Replacing a pipe	01	1	e_1^0
Relining a pipe	10 or 11	2 or 3	e_2^0

Table 5-6 An Example of a Mapping for the Network

Pipe Tag	1		2		3		4		......	17	
Variable:	e_1	d_1	e_2	d_2	e_3	d_3	e_4	d_4	.....	e_{17}	d_{17}
Genotype:	10	111	00	011	11	011	01	101	......	10	111
Index:	2	7	0	3	3	3	1	5	......	2	7
Phenotype:	e_2^0	d_7^0	e_0^0	d_3^0	e_3^0	d_3^0	e_1^0	d_5^0	......	e_2^0	d_7^0
A solution:	relining	610	leaving	305	relining	305	replacing	457......		relining	610
(action and diameter in mm)											

5.4.2 Results and a comparison

The optimal solutions found by the messy genetic algorithm optimisation technique for the case study network are compared with the previous results of Kim and Mays (1994) in Table 5-7 and Table 5-8. Two solutions have been found by using mGANET. The solution mGANET 1 has been obtained by using the same pump horse power as Kim and Mays (1994). It shows that the mGANET found a lower cost solution than the method of branch and bound coupled with a non-linear programming (Kim and Mays 1994). The mGANET solutions for this case study are compared to the previous result by Kim and Mays (1994). The solution mGANET 1 reduces the cost from $11.96 million to $11.58 million as shown in Table 5-7. However, it also shows that the cost of the pipeline rehabilitation is only about 10% of the total cost. In other words, the cost of the water distribution system rehabilitation is dominated by the pump energy cost. Thus another mGANET run was performed by relaxing the lower bound of pump horse power from 1454 *kW* to 745.70 *kW*. The solution mGANET 2 found by this run shows that mGANET has chosen a more expensive pipeline rehabilitation strategy but a much lower-cost horse power of the pump than the previous result. The total cost dropped from $11.96 million to $8.76 million a saving of 26.7%. Also the pressure heads at demand nodes, as illustrated in Figure 5-2, are closer to the minimum allowable pressure heads than the mGANET 1 solution.

Table 5-7 Cost Comparison of the Optimal Solutions

Type of cost (US$)	Kim and Mays (1994)	mGANET 1	mGANET 2
Pipeline rehabilitation	1,372,838	996,341	3,280,193
Pump energy	10,587,207	10,587,207	5,480,615
Total cost	11,960,045	11,583,548	8,760,808
Percentage savings	-	3%	26%

Table 5-8 Comparison of the Optimal Rehabilitation Strategy

Components	Existing pipe	Kim and Mays (1994)		mGANET 1		mGANET 2	
No.	Diam. (mm) & pump (kW)	rehabilitation actions	Diam. (mm)	rehabilitation actions	Diam. (mm)	rehabilitation actions	Diam. (mm)
Pipe 1	610	Reline	610	Reline	610	replace	610
Pipe 2	457	Reline	457	Reline	457	replace	610
Pipe 3	457	As is	-	As is	-	Reline	457
Pipe 4	152	As is	-	As is	-	As is	-
Pipe 5	381	As is	-	As is	-	Reline	381
Pipe 6	381	As is	-	As is	-	As is	-
Pipe 7	381	As is	-	As is	-	Reline	381
Pipe 8	305	As is	-	As is	-	As is	-
Pipe 9	229	As is	-	As is	-	As is	-
Pipe 10	305	As is	-	As is	-	As is	-
Pipe 11	305	Reline	305	Reline	305	Replace	508
Pipe 12	381	Replace	437	Reline	381	Replace	610
Pipe 13	381	Replace	269	Reline	381	Reline	381
Pipe 14	381	As is	-	As is	-	As is	-
Pipe 15	381	As is	-	Reline	381	Reline	381
Pipe 16	381	As is	-	As is	-	As is	-
Pipe 17	229	As is	-	Reline	229	Reline	229
Pump (kW)	1454	-	1454	-	1454	-	818

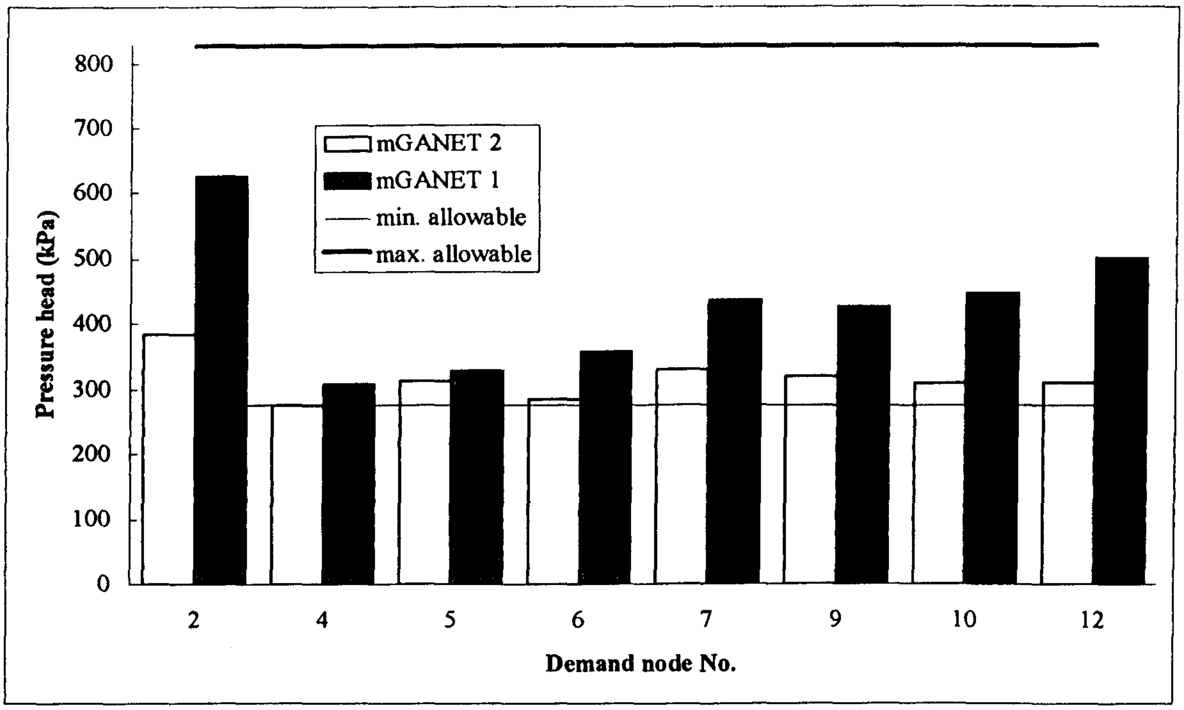


Figure 5-2 Pressure Heads of mGANET Solutions

5.5 Summary

An approach to the optimisation of the rehabilitation for water distribution systems has been developed. An optimal rehabilitation model has been formulated to select a rehabilitation action for each pipe from a set of possible rehabilitation actions such as duplicating, cleaning, relining, replacing or just leaving a pipe as it is. A rehabilitation action and the associated pipe size is chosen from a list of commercially available discrete pipe sizes, and a pumping capacity is employed so that the water demand and minimum allowable hydraulic pressure at all the nodes are satisfied while the total cost of the rehabilitation is minimised. The total cost considered includes both the pipe rehabilitation cost and pumping cost.

To minimise the total cost, an integrated program mGANET, which has coupled the messy genetic algorithm and a hydraulic network solver EPANET, has been employed to search for the optimal rehabilitation strategy. The results of the case study have shown that the messy GA is efficient at searching for the optimal solution. It also shows that the cost of the rehabilitation is dominated by the pump energy cost, which is almost 90% of the total cost of the solution mGANET2 and about 70% of the total cost of the solution mGANET1. This implies that the optimal rehabilitation of water distribution system must consider not only pump useful horse power but also the pump operation (scheduling). The consideration of these aspects will provide more accurate information to evaluate the rehabilitation strategies in an optimal rehabilitation model of the water distribution systems in the future.

6. FAST MESSY GENETIC ALGORITHM FOR OPTIMAL REHABILITATION OF LARGE-SCALE WATER DISTRIBUTION SYSTEMS

6.1 Introduction

Engineering and science disciplines have made extensive use of genetic algorithms. A number of genetic-based search paradigms have been developed and applied to different problems. A growing demand for algorithms to solve new problems and a never-ending process of designing algorithms strongly suggests the need for more efficient and more robust genetic-based optimisation techniques. These algorithms should make few assumptions regarding the objective functions and use as little domain knowledge as possible.

The design of the pipeline within a water distribution system involves selecting pipe sizes from a set of commercially available pipe sizes. It is a discrete optimisation problem. The discrete optimisation problem appears to be easier to solve than the continuous one because fewer possible solutions exist. In general, however, it is more difficult to solve. This is due to the fact that the discrete design space is non-differentiable and nonconvex. The standard gradient-based programming techniques and optimality criteria cannot be applied directly. A global optimal solution of the discrete optimisation problem can be obtained only by an *exhaustive search* (Arora et al. 1994). The genetic algorithm has been proven to be robust for the optimisation of pipeline networks but usually requires a large number of function evaluations (Simpson et al. 1994; Dandy et al. 1996).

Genetic-based search techniques have been developed to exploit the information (fitness and/or objective function value) gathered from samples taken from the search space. The search space is quantified by different regions. The algorithms use the information to decide which region to explore next. In other words, the search space can be seen as being classified into different classes which represent certain relationships among solutions. Thus, the search space can be decomposed into three partitions including the relation space, the class space and the sample space (Kargupta 1995). For example, for a 4-bit problem, # # # *f* is a relation between the samples, where *f* means a fixed bit. The strings # # # 1 and # # # 0 are two classes of the relation, where # is called don't-care symbol which can be either 0 or 1. Finally the strings 0110 and 1110 are two samples of class # # # 0. Searching for an optimum requires a search for right relations and classes.

A standard genetic algorithm (sGA) (Holland 1975) searches for relations and classes implicitly. The sGA population combines the relation space, class space and sample space all together. This results in sGA being a poor search for relations. The messy GA (Goldberg et al. 1989) is designed to search for the relations and classes by using (1) a variable-length genotype; (2) an explicitly enumerative initialisation; and (3) the *cut* and *splice* genetic operators. However, the original messy GA usually requires a large number of members in the initial population. It is difficult to apply the original mGA to a highly dimensional problem. A fast messy GA (Goldberg et al. 1993), using a probabilistically complete initialisation and building block filtering process, was introduced to eliminate this bottleneck. In this Chapter, the messy GA and the fast messy GA are applied to the optimisation of design of pipeline networks. The efficiency of the messy GAs are compared with standard GA paradigms. Finally, the fast messy GA is employed to solve a discrete optimisation problem of large-scale rehabilitation of water distribution systems.

6.2 Lessons From Genetic Algorithm Paradigms

6.2.1 An overview of messy genetic algorithms

A messy genetic algorithm is described as messy because it allows variable-length strings that may be under- or overspecified with respect to the problem being solved as described in Chapter 3. The messy gene, an ordered pair, is defined by its name (a tag) and value. A messy chromosome is a collection of the messy genes. The underspecified strings may be filled in by using *a competitive template,* a string that is locally optimal to the previous level, while the overspecified strings may be dealt with by using a *first-come-first-served* rule scanning from left to right. The messy GA is run in an era-wise search. Each era has three stages namely (1) initialisation; (2) a primordial phase; and (3) a juxtapositional phase. Initialisation in the original messy GA was performed by creating a population of all strings of building blocks of length k, where $k = 1, 2, ..., K$ and $K =$ the maximum number of eras. Good building blocks of the desired length are selected in primordial phase by performing thresholding selection alone. In this phase, it also adjusts the population size to be a prescribed size for the reproductive processing of juxtapositional phase. Having enriched the population with the best building blocks in the primordial phase, thresholding selection and the cut-and-splice operators are used to reproduce offspring over generations with some similarity to the standard GA. The optimal string found in era k is used as competitive template in era $k+1$ of the messy GA run.

A messy genetic algorithm is one among the rare class of adaptive search algorithms that emphasise searching for appropriate relations. The messy GA uses a competitive template and explicit enumeration of good classes—building-blocks—to ensure correct decision making. However, explicit enumeration of building-block essentially means a complete lack of the benefits of implicit parallelism. The development of the fast messy GA has eliminated the major bottleneck of the messy GA—the enumerative initialisation. The probabilistically complete initialisation and the building-block filtering process introduced in the fast messy GA (Goldberg et al. 1993) were used to detect better classes from better relations without sparing the advantage of implicit parallelism.

The fast messy GA has been applied to solving a target tracking problem—a problem a high degree of interest in the field of air defence system (Kargupta 1995). On the hundreds of target-tracking problems (150 clustered missile problem and 300 clustered missile problem), the fast messy GA has found all track correctly after a reasonable amount of evaluations. More recently the fast messy GA has been applied to discrete optimisation of pipeline networks and compared with other genetic algorithm paradigms (Wu and Simpson 1997). It was shown that the fast messy GA is the most efficient algorithm among the genetic algorithm paradigms for discrete nonlinear optimisation.

6.2.2 Why a standard GA sometime fails

A standard genetic algorithm using bit-based representation is usually a robust solver for a wide range of search problems. The fundamental theory describing how the genetic algorithms work is schema theorem (Holland 1975). A schema is a string over the alphabet $\{*, 0, 1\}$ of the length *n*, where *n* is the length of a genotype, which encode a complete solution. A * symbol is a so-called don't-care symbol, which can represent either 0 or 1. For instance, h = 1* * 1 0 is a schema over chromosome of length 10 such as * 1* * 1 0 * * * * . $\delta(h) = 5$ is the physical distance between the outermost defining positions of the schema, and usually called the length of the schema. $o(h) = 3$ is the number of fixed positions contained in the schema, and usually called the order of the schema. If a schema, which is relatively short and of low order, and has above-average fitness, is called a building block. The building blocks are expected to grow in subsequently generations. An important issue, however, is the linkage of the building blocks. The linkage is regarded to be tight if the fixed bits belonging to the building block are close to each other on the chromosome, While loose linkage occurs when the fixed bits are scattered on the chromosome. Loose linkage cause problems to the GA since the standard GA doesn't have any explicitly designed mechanism to handle the loose linkage. For many optimisation problems, the linkage between bits is unknown in advance, so handling the loose linkage is crucially important for an efficient genetic algorithm.

6.2.3 Why messy genetic algorithm works

The messy genetic algorithm has been carefully designed by respecting the schema theorem, particularly by considering the need for *tight linkage* quite seriously. It gets the linkage right prior to subsequent genetic processing. Goldberg et al. (1989) found that messy genetic algorithm was successful at (1) obtaining tight building blocks; (2) increasing the proportion of the best building blocks; (3) making good decisions among the building blocks and eventually and (4) exchanging the building blocks well. Those features lead the messy GA to search efficiently for the optimal solution.

A genetic-based search starts with random initialisation. Initial population of the original messy GA was initiated by complete enumeration as described in Chapter 3. It creates a population with a single copy of all substrings of length or order k. It ensures that all building blocks of the desired length are obtained. After the initialisation, the primordial phase, in which selection alone is performed, enriches the population with fitter genotypes.

This phase, together with thresholding selection in juxtapositional phase, ensures that a high proportion of the best building blocks grows over generations.

The messy code relaxes the fixed-locus representation of the standard GA. It is the messy code that enables the arbitrary building blocks to achieve a tight building blocks by rearranging the tagged genes in close proximity to one other. The tight building blocks correspond to a low probability of good ones being destroyed by crossover and other genetic operators. Thus the messy GA exchanges the building blocks well.

A messy genetic algorithm makes good decisions among the building blocks by decomposing the search space into two different spaces the sample space—template space and the building block space—the population strings (Kargupta 1995). During the primordial phase every string in the population has a length less than the problem length. Each of them defines an equivalence class—a building block. Therefore, the strings in the primordial stage represent the building block space. In the meantime, the template is always a full string and defines the sample space in the messy GA. Since the main decision making in the mGA is made during the primordial stage, the decomposition of the search space makes the decision less noisy when compared to the standard GA. Thus unlike the standard GA, the search for tight building blocks in the messy GA is more accurate and less susceptible to err because of the explicit enumeration and the use of a locally template for class evaluation.

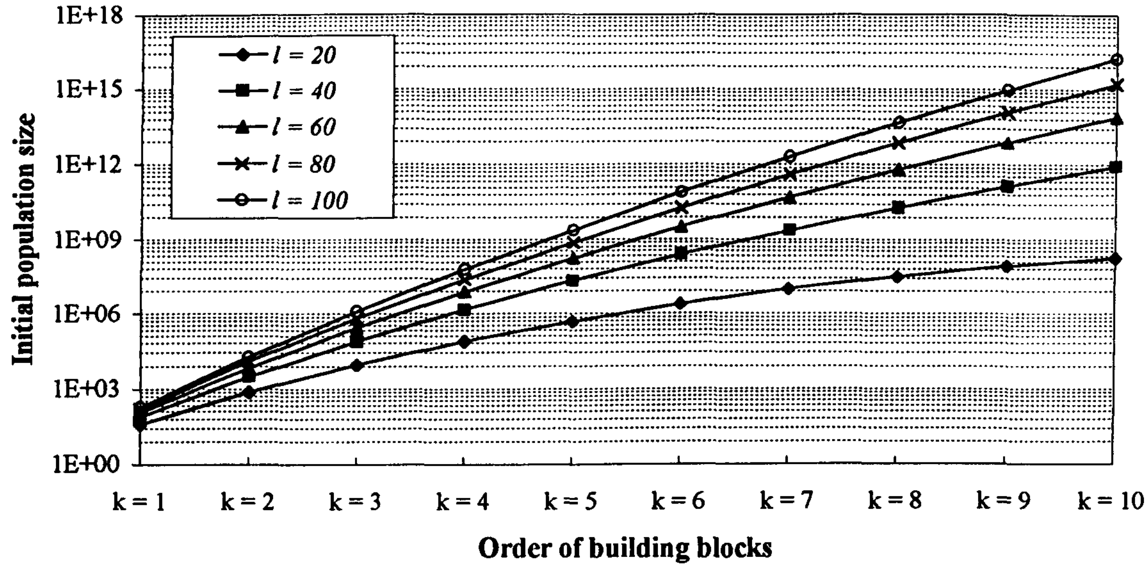


Figure 6-1 Initial Population Sizes Required by Original Messy GA Using Enumeration

6.2.4 Bottleneck of original messy genetic algorithm

The original messy GA, as described in Chapter 3, adopted enumeration of all strings of length k to create the initial population. This requires a population size of $n = 2^k \binom{l}{k}$. Obviously, the number of the evaluations of the initial population increases exponentially as the problem length l and/or the order k of the building blocks increase. The initial population size is in order of $O(l^k)$ while the juxtapositional phase processing of $O(l \log l)$ (Goldberg et al. 1990). The overall computation of the original messy GA was dominated by the initialisation. For example, as shown in Figure 6-1, the initial population size by the complete enumeration of the order 4 building blocks for a problem of 40-bit is more than one million. This bottleneck has been overcome by introducing a probabilistically complete initialisation and a gene filter procedure (Goldberg et al. 1993) into the messy GA. It speeds up the search process, and is called a fast messy GA.

6.3 The Structured Messy Genetic Algorithm of Hahal et al. (1997)

A genetic algorithm called structured messy genetic algorithm (SMGA) was developed by Hahal et al. (1997) and applied to optimal rehabilitation of water networks. The SMGA could be more precisely seen as being constructed (simplified) from the original messy genetic algorithm (mGA) by Goldberg et al. (1989, 1990) rather than Holland (1975). It is believed that incorporating more features of the original messy GA would improve the efficiency and effectiveness of the SMGA as discussed in the following sections.

6.3.1 Structured messy genotypes

The structured messy genetic algorithm (SMGA) starts a complete enumeration of all single element namely one-variable solutions, for instance one pipe rehabilitation solutions. All the single variable (or equivalent one digit genotype) decisions are evaluated and are concatenated to assemble strings of a certain length (the number of digits) to form an initial population. The length of the strings increases with a fixed step size over generations. Thus the SMGA employed fixed-length strings in one generation, that is, the chromosomes in each generation have the same length or a *tidy* structure. The *tidy* genotype representation, similar to a standard GA genotype representation enables standard GA operators such as tournament selection, crossover and mutation to be applied in a SMGA evolution process. In contrast, the messy genetic algorithm of Goldberg et al. (1989) was originally developed by using a tagged gene representation, that is, each gene is represented by its value and gene location in the chromosome. The chromosome can be underspecified with some genes not included in the chromosome. These missing genes are filled in by using a competitive template, which can be a random chromosome initially and replaced by a locally optimal string as the messy GA progresses. Alternatively the chromosome can be overspecified when there is more than one values being generated for the same gene location. The overspecified genes are eliminated by following a rule of the first-come-first-served by scanning from left to right. The length of strings can be varied not only over generations but also within each generation. It provides a flexible coding representation of variable-length to solve search problems of bounded difficulty.

6.3.2 Structured messy evolution

The proposed SMGA a procedure of Hahal et al. (1997) involves three stages including complete enumeration of single-elemental solutions, a concatenation process of the longer strings and standard GA reproduction process. The complete enumeration of single-elemental solutions creates one-bit building blocks, which provides a genetic material pool for the concatenations and the reproductions of the SMGA evolution. The SMGA revisits the pool to pick up good gene materials to assemble a population of longer strings to be evolved. Obviously, the quality of the population depends on both of the quantity and quality of the building blocks in the gene material pool. Since only one-bit building blocks are generated by the enumeration, it hardly guarantees adequate building blocks being provided for the further evolution stages. This may contribute a low efficiency of the SMGA as observed by Hahal et al. (1997) while applying the SMGA to a single-objective optimisation.

The original messy GA procedure by Goldberg et al (1989, 1990) starts with a complete enumeration of order one (one-bit) building blocks, a primordial phase then follows to select fitter building blocks for evolving the population and finally a juxtapositional phase in which thresholding selection, *cut* and *splice* operations take place. The best string found is used as competitive template for evaluating underspecified genotypes. The evolution process continues by enumerating order-two (two-bit) building blocks. The complete enumeration of the building blocks of different orders guarantees adequate gene materials being provided for evolving the population. Although the SMGA can be extended to incorporate a similar process by generating more than one elemental solutions, it will encounter the same bottleneck of the complete enumeration of the building blocks as found for the original messy genetic algorithm. The complete enumeration of building blocks lacks of implicit parallelism of the genetic algorithm and requires a huge number of evaluations for the high order building blocks of a large dimensional problem. Goldberg et al. (1990) overcame this problem by introducing a *probabilistically complete* initialisation and a gene filtering process (Goldberg et al. 1990) to replace the initialisation scheme of the complete enumeration of a certain order building blocks. Thus it is believed that efficiency and effectiveness of the genetic algorithm paradigm can be enhanced by maintaining the original messy GA features including the variable-length genotype which allows the string lengths to vary not only over generations but also within the generation, a two-phase evolution, thresholding selection, and also incorporating the gene filtering techniques for filtering building blocks of any order to provide adequate gene materials for the messy genetic algorithm evolution.

To avoid the possible computer memory difficulties of a genetic algorithm implementation, advanced programming features such as dynamic linked structures, bitwise variables and operations can be used. One gene stored as an integer variable occupies 32 bits or 64 bits while one gene of binary representation stored as a one bit variable just needs one bit. Thus tremendous memory can be saved by using bitwise storage. The dynamic linked structure of genotypes provides a flexible and efficient memory operation, and together with the bitwise operations can improve the efficiency of the genetic algorithms without compromising the effectiveness.

6.4 Fast Messy Genetic Algorithm

Explicit enumeration of the building blocks of a certain length is very computationally time consuming for applying the original messy GA to highly dimensional problems, and also for running the messy GA over a number of eras for a simple problem. The problem is the need to evaluate an initial population of $2^k\binom{l}{k}$ substring structures, from which good building blocks are selected. The fast messy GA preserves the feature of searching for tight building blocks of the original messy GA. The tight building blocks are not initiated by explicit enumeration but obtained by probabilistically complete initialisation and building block filtering. It brings some benefits of implicit parallelism back to the messy GA. The components of fast messy genetic algorithm are described in the following sections.

6.4.1 Components of fast messy genetic algorithm

The fast messy genetic algorithm preserves the main framework of the original messy genetic algorithm. The explicitly initial enumeration of building blocks is replaced with probabilistically complete initialisation and building block filtering procedure. Both techniques are incorporated into the original messy GA to improve the efficiency of the messy evolution process.

(i) *Probabilistcially complete initialisation*

The original messy GA initially enumerates all the order-*k* building blocks. This initialisation is deterministic or can be considered to be a non-GA type operation. The probabilistically complete initialisation has been developed to replace the explicit enumerative initialisation. The idea of the probabilistically complete initialisation is that all the order-*k* building blocks can be defined using a much smaller number of strings, when the string length is much greater than *k*. In other words, multiple combinations of the building blocks can be defined by one long string. The length of the initial strings and the population size are the two crucial factors in the initialisation stage.

Kargupta (1995) has shown that the length of the initial strings is the trade-off between reduction of population size and increase in error probability. As the string length increases, the population size decreases quickly, but the error probability increases. The population size for the probabilistically complete initialisation is suggested $n = O(l')$, l' is the length of initial strings and is suggested to be taken as $l' = l - k$, where l is the problem length or the number of the bits representing one solution, and k is the era of the messy GA run or the length (order) of the building blocks that is expected to be filtered for juxtapositional phase. Thus the probabilistically complete initialisation reduces the initial population size from $O(l^k)$ to $O(l)$, and consequently improves the search efficiency.

(ii) *Building-block filtering*

The Probabilistically complete initialisation technique generates an initial population of strings with a length of $l - k$. Building blocks of order- k are expected to be filtered out by gradual reduction of the string length. The length of the strings is reduced by random deletion of genes. This process of detecting the good classes by thresholding selection and gene deletion is called *building block filtering.*

The building-block filtering process offers a way of gradually detecting certain order-k classes from strings of length l', where $l'>k$. During this stage, the string is selected in presence of thresholding and the genes are occasionally deleted to reduce the string length from l' to k. The gene deletion rate should be chosen so that it is on average less than the rate at which better string get more copies by selection. Good results has been achieved for the numerical experimental testing of fast messy GA by using a deletion rate of 0.5 (Kargupta 1995). A simple building block filtering scheme has been proposed as shown in Figure 6-2. Thresholding selection continues with constant string length to reproduce more copies of the better strings. No fitness evaluations are required at this stage. Gene deletion follows by randomly deleting half of the current genes, which reduces the string length to just half of the previous string length. These shorter strings are then evaluated and the same procedure of thresholding selection and gene deletion are applied until the string length is the same as the order k of the required building blocks.

Repeat

    Repeat

        thresholding selection;

    Until *(population size > juxtapositional size)*

    delete $\frac{l}{2}$ *genes at random;*

    evaluate all new genotypes;

Until *(string length* $\leq$ *building block length* k *)*

Figure 6-2 Pseudo-code of Building Block Filtering

6.4.2 A framework for the fast messy genetic algorithm

The fast messy GA, like original messy GA, is run over a number of eras. The main difference is that the selection-only primordial phase in original messy GA is replaced by a primordial stage that employs the probabilistically initialisation, thresholding selection and gene deletion for filtering the building blocks. A framework describing the fast messy GA is given in Figure 6-3. It starts era $k = 1$ with initial strings of length $l' = l - 1$. The good building blocks are filtered out by following a procedure as shown in Figure 6-2 until the string length equals to order of the era. Then the juxtapositional phase follows to reproduce better individuals by applying messy genetic operators of *cut* and *splice* and also mutation operations. The optimal string of this era is used as competitive template of next era. The fast messy GA is run era by era until the optimal or near-optimal solution is obtained or the algorithm may be terminated when reaching a specified maximum number of eras.

```
{
        era = 0; /*the order of building block*/
        while ( messy GA termination is not true )
        {
                era = era + 1;
                probabilistically complete initialisation;
                evaluation; /*network solver is called */
                t = 0;
                while (building block filtering is true)
                {
                        threshold tournament selection;
                        building block filtering;
                        t = t + 1;
                }
                while (juxtapositional phase is true )
                {
                        threshold tournament selection;
                        cut;
                        splice;
                        mutation;
                        evaluation; /*network solver is called */
                        t = t + 1;
                }
        }
```

Figure 6-3 A Framework of Fast Messy Genetic Algorithm

6.5 An Enhanced mGANET

The components of the fast messy GA described above has been incorporated into the integrated computer system mGANET for optimisation of design and rehabilitation of water distribution systems. The mGANET developed in Chapter 3 was based on the original messy GA. It was efficient at searching for the optimal solution, but constrained to solve small dimensional problems or run a limited number of eras due to the bottle neck problem in the initialisation of eras of the original messy GA. The fast messy GA has been developed and incorporated into the mGANET, as shown in Figure 6-4, to overcome the difficulty of evaluating a huge number of substrings in the original messy GA.

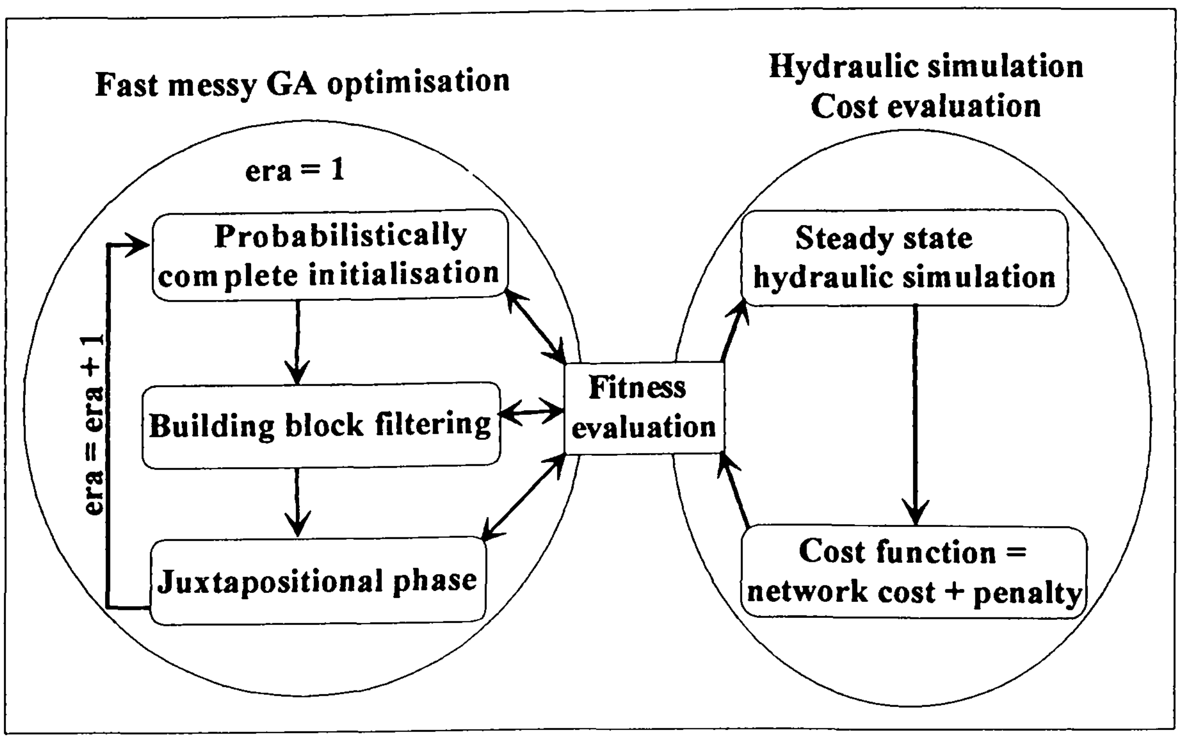


Figure 6-4 Program Structure of Enhanced mGANET—fmGANET

The enhanced program called fmGANET maintained the same program structure as mGANET described in Chapter 3. The fast messy GA is linked with hydraulic network solver EPANET through the fitness evaluation of the genotypes and search for the optimal solution over eras. The optimal string found in current era is used as competitive template in next era. The program is terminated by reaching the maximum number of eras.

6.6 A Comparison Study—The Two Reservoir Network

A network with two water supply sources and fourteen pipes, studied by Simpson et al. (1994) as shown in Figure 3-1, has been chosen for the comparison study. This network has also been solved by the messy GA using complete enumerative initialisation in Chapter 3. The results from the previous studies provide results for comparing the performance of the messy GA approach with the standard GA approach.

Table 6-1 Population Sizes of Messy Genetic Algorithms for Optimisation of the Two Reservoir Network

Messy GAs	Original messy GA			Fast messy GA		
Eras	era 1	era 2	era 3	era 1	era 2	era 3
Initialisation	300	435	4060	60	60	60
Juxtapositional	150	150	150	150	150	150

6.6.1 Fast messy GA coding, decoding and parameters

The genotype representation and the fitness formulation given in Chapter 3 are used in this comparison study. Three binary bits have been used to represent each pipe size variable for the five new pipes and three existing pipes to represent 8 possible choices of pipe sizes.

Two binary bits have been used for each existing pipe to represent 3 possible choices of rehabilitation actions that include cleaning, leaving or duplicating an existing pipe. Thus 30 bits are needed for solving the problem when using the discrete diameter formulation. A decoding and mapping scheme from genotype to phenotype for optimisation of design and rehabilitation of water distribution systems is given in Chapter 3. The penalty factor for pressures which do not meet the minimum allowable pressure constraints for this problem was chosen to be $5000/m of deficit to match the value taken by Simpson et al. (1994). Table 6-1 shows the population sizes used by the original and fast messy GA. The other parameters are splice probability $P_s = 1.0$, cut probability $P_k = 0.017$, Mutation Probability $P_m = 0.01$ and maximum juxtapositional generations $NN=10$.

6.6.2 Results and comparison

The optimum discrete solution for this problem was found by the messy GA with different random seeds and compared with the standard GA results (Simpson et al. 1994; Simpson & Goldberg 1994) as in Table 6-2. Typical convergence rates of the messy GA with the seed 0.7 are respectively given for using explicitly enumerative initialisation and probabilistically complete initialisation in Figure 6-5.

As shown in the Table 6-2, the fast messy GA is the most efficient at searching for the global optimal solution of discrete optimisation of pipeline networks. The messy GAs have found the lowest cost solution (global optimum) in each of the 10 runs with different random seeds. The numbers of original mGA evaluations needed for achieving the global optimal solution are less than for the standard GA. The messy GA using enumerative initialisation required only one third to half of the evaluation numbers of the standard GA (Simpson et al. 1994), and also less than the GA with tournament selection (selection pressure s = 2). Simpson and Goldberg (1994) observed that increasing tournament pressure (s = 5) for the standard GA could reduce the number of evaluations, and thus improve the search efficiency, but too much pressure (s =20) might lead the search to a local optimum. The fast messy GA, using probabilistically complete initialisation and building block filtering, has further reduced the number of the evaluations, being approximately one third of the evaluation numbers of the original messy GA.

Table 6-2 Results of Comparison of GA Paradigms for the Two Reservoir Network

Run No.	Standard Genetic Algorithms					Messy Genetic Algorithms			
	Roulette wheel selection (Simpson et al. 1994) (N = 100; Pc = 0.9; Pm = 0.02)		Tournament selection (Simpson & Goldberg 1994) (N = 1000; Pc = 0.5; Px = 0.5; Pm = 0.0)			Explicitly enumerative initialisation (Wu & Simpson 1996)		Probabilistically complete initialisation	
	Cost (m$) (% difference from optimum)	Achieved at evaluation number	Cost ($m)	Achieved at evaluation number (s = 2)	Achieved at evaluation number (s = 5)	Cost ($m)	Achieved at evaluation number	Cost ($m)	Achieved at evaluation number
1	1.7910 (2.3%)	23,400	1.7503	9,000	4,000	1.7503	6,148	1.7503	1,112
2	1.7503*	10,350	1.7503	9,500	4,000	1.7503	6,148	1.7503	2,243
3	1.8417 (5.2%)	22,410	1.7503	8,500	4,500	1.7503	6,148	1.7503	1,855
4	1.8390 (5.1%)	15,660	1.7503	9,500	5,500	1.7503	8,958	1.7503	3,004
5	1.7503	17,190	1.7503	8,000	4,500	1.7503	2,957	1.7503	4,053
6	1.7503	11,070	1.7503	8,000	5,000	1.7503	2,522	1.7503	2,722
7	1.7503	10,080	1.7503	8,000	4,000	1.7503	8,758	1.7503	3,053
8	1.7999 (2.8%)	4,410	1.7503	7,500	4,500	1.7503	10,042	1.7503	2,622
9	1.7503	12,510	1.7503	10,000	4,000	1.7503	3,977	1.7503	1,622
10	1.7503	19,890	1.7503	10,000	3,000	1.7503	6,148	1.7503	1,722
Average		14,697		8,800	4,300		6,181		2,400

*The global optimum solution

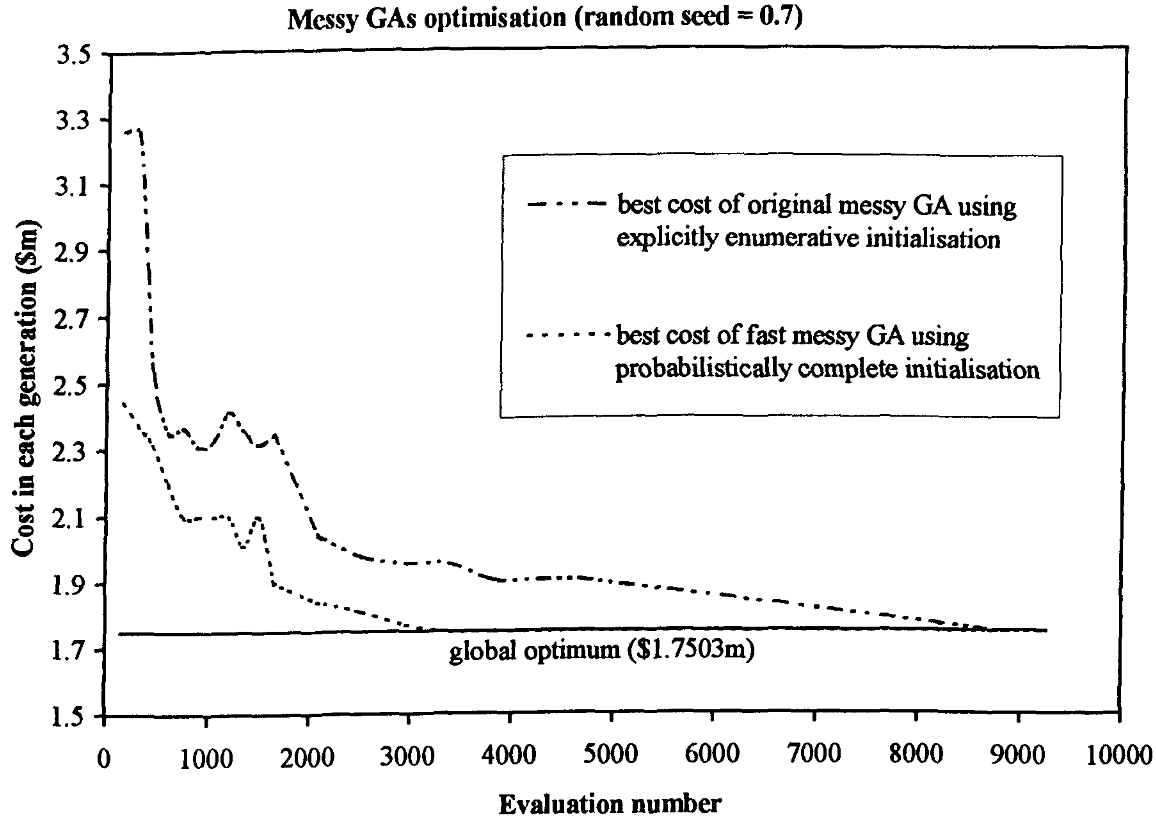


Figure 6-5 Comparison of Generation Best Cost for Original and Fast Messy Genetic Algorithm Optimisation of the Two Reservoir Problem

The fast messy GA employed the probabilistically complete initialisation and the building block filtering process. The initialisation just required a population of $O(l)$ of string length $(l - k)$. The fast mGA started the first era (order 1 with $k = 1$) with 60 strings of 29-bit strings. It was followed by the building block filtering process, in which the strings are cut in half every generation and evaluated by using a random template. Members for the next generation were selected by thresholding selection. The building block filtering process continues until the string length is equal to 1. The population size was increased to 150 at the end of building block filtering process. As for the original messy GA, it was then followed by the juxtapositional phase. For the second and the third eras the same population size of 60 as for the first era was used in the fast messy GA, and also the building block filtering was carried out for order 2 and order 3. In contrast, as shown in Table 6-1, the original messy GA required an initial population size of 435 and 4060 for era 2 and era 3 respectively. Thus, the fast messy GA overcomes the bottleneck of the original messy GA and provides a more efficient search algorithm for the discrete optimisation of the pipeline networks.

6.7 Optimal Rehabilitation of Large-scale Network

Optimisation of rehabilitation of water distribution systems is a non-linear discrete optimisation problem. This type of problem has been studied previously by applying many different optimisation techniques including genetic algorithms. Although it has been found that the GAs are generally efficient at solving the problems of discrete optimisation, the difficulty of searching for optimal or near-optimal solutions increases as the dimension of the problem increases. A real water distribution system, however, always involves hundreds of pipes and dozens of pumps and valves. Previously developed GA techniques for optimisation of rehabilitation of a large-scale water distribution network are not efficient at searching for the optimal solution. The original messy GA has been shown more efficient than the standard GA paradigm at searching for the optimal solutions, but suffers from the bottle neck of complete enumerative initialisation of certain order building blocks. In this Chapter, the fast messy GA has been implemented and compared with the other genetic algorithm paradigms. It shows that the fast messy GA provides the most efficient genetic search algorithm and overcomes the curse of the dimensionalty. As an example of its application the fast messy GA is applied to optimal rehabilitation of a large-scale water distribution network in this section.

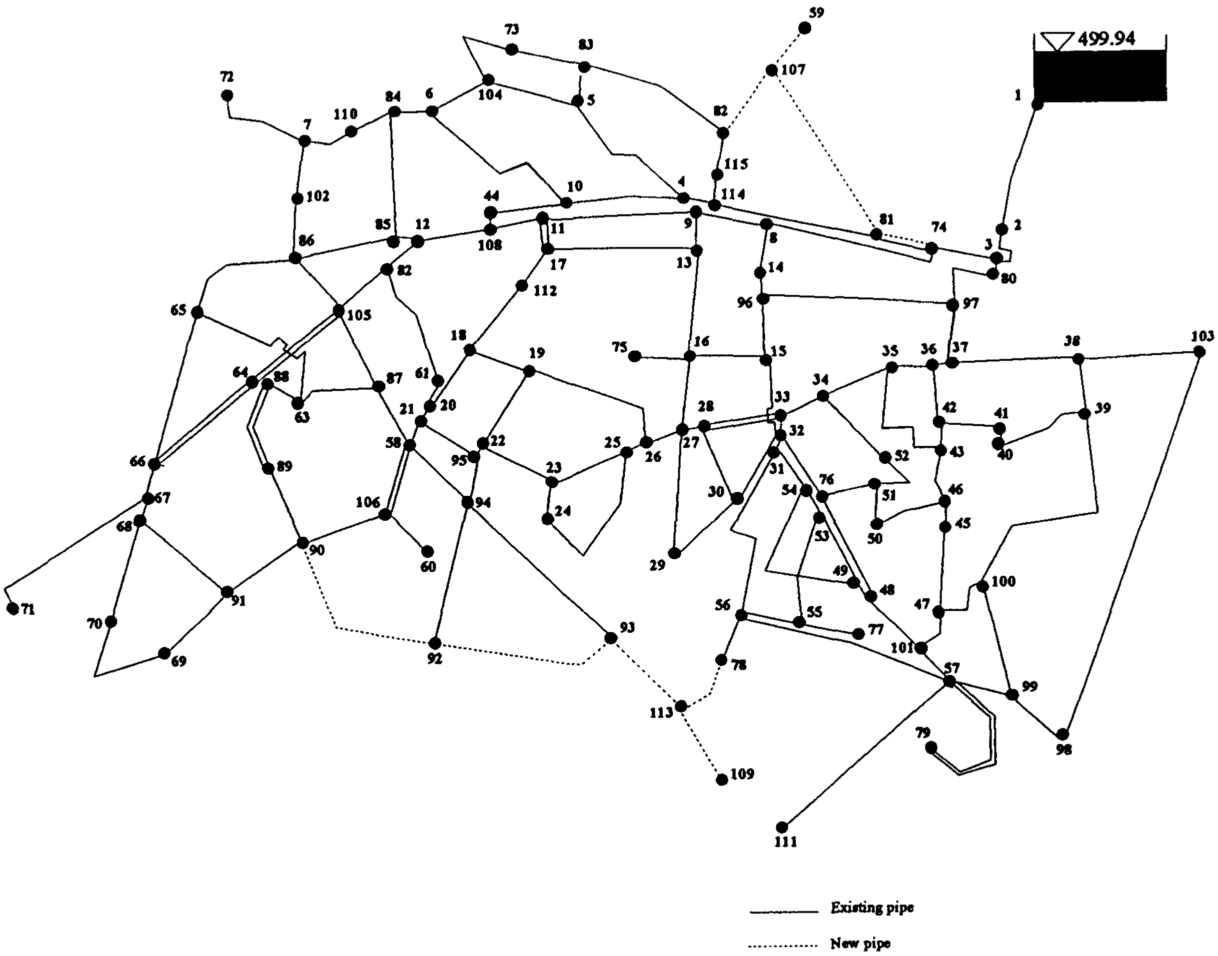


Figure 6-6 Layout of a Moroccan Network

6.7.1 A Moroccan network

A real water distribution system in Morocco, as shown in Figure 6-5, is for a town of 50,000 inhabitants. This network consists of 115 nodes and 167 pipes and is chosen to test the efficiency and effectiveness of the fast messy GA for optimal rehabilitation of the large-scale water distribution system. The problem has been studied by Hahal et al. (1997) using the structured messy genetic algorithm. The fast messy genetic algorithm has been employed to solve this problem.

6.7.2 Optimal rehabilitation criteria

The optimisation of rehabilitation of the Moroccan network has been specified as minimisation of the total cost of the rehabilitation by searching for the optimal pipe diameters for the new pipes and the optimal combination of rehabilitation actions including replacement of pipes, duplication of pipes, cleaning or lining of pipes or leaving pipes as they are. The optimal solution is sought subject to the minimum pressure head of 20.0 metres at each node. Table 6-3 and 6-4 give the data for existing pipes and new pipes to be added to the existing system. Table 6-6 gives the cost information of the rehabilitation action and the pipe size associated with the action. It shows that there are 4 possible rehabilitation actions and 8 discrete pipe sizes available for each pipe.

A repair cost is expected for the pipe if no rehabilitation action is taken with the assumption (Hahal et al. 1997) that new pipes are assumed to have no annual repair cost during their first 10 years, as they are usually under warranty for this period. The repair cost is calculated as follows (Hahal et al. 1997).

$$C_{rep}(j) = \sum_{t=tp}^{t=tr} \frac{J(t)c_{rep}(j)}{(1+r)^{t-tp}} \qquad (6\text{-}1)$$

where $C_{rep}(j)$ = repair cost of a break for pipe j; r = interest rate; tp = present year; tr = year $tp + 10$; and $J(t)$ = break rate in year t, which is given by

$$J(t) = J_o(1+b)^t \qquad (6\text{-}2)$$

where J_o = break rate in year 0 (break/km/yr); b = break rate growth coefficient and t = time in year.

Table 6-3 Data for Existing Pipes of the Moroccan Network

Pipe No.	From Node	To Node	Diameter (mm)	Length (m)	Hazen William C coefficient	Repair cost cost	Break rate (no./Km/yr)
1	1	2	400	330	120	0	0
2	2	3	400	120	120	0	0
3	3	80	200	40	90	0	0
4	74	8	350	422	120	0	0
5	81	114	100	432	90	0	0
6	4	5	80	350	80	3000	1.1
7	83	73	80	190	80	3000	1.1
8	5	104	80	210	90	0	0
9	10	6	100	420	70	4500	1.1
10	84	110	100	220	90	0	0
11	7	72	100	254	90	0	0
12	3	74	300	154	120	0	0
13	8	9	350	170	120	0	0
14	8	14	100	130	90	0	0
15	9	13	100	94	90	0	0
16	96	15	80	150	90	0	0
17	15	16	100	180	80	4500	0.8
18	16	13	100	240	80	4500	0.9
19	16	75	100	130	80	4500	0.95
20	13	17	100	336	70	4500	1.15
21	11	17	100	80	70	4500	1.1
22	9	11	350	360	120	0	0
23	4	10	100	280	90	0	0
24	44	10	100	180	90	0	0
25	108	12	300	180	120	0	0
26	112	18	200	130	120	0	0
27	18	19	100	150	90	0	0
28	26	19	100	366	90	0	0
29	19	22	80	210	100	0	0
30	21	95	100	150	100	0	0
31	20	21	150	36	120	0	0
32	18	20	150	170	120	0	0
33	61	20	80	80	90	0	0
34	62	61	100	300	90	0	0
35	12	62	200	100	120	0	0
36	85	86	100	240	100	0	0
37	62	105	200	144	120	0	0
38	63	65	80	440	80	3000	0.95
39	105	64	100	260	80	4500	0.85
40	65	66	100	380	90	0	0

Pipe No.	From Node	To Node	Diameter (mm)	Length (m)	Hazen William C coefficient	Repair cost cost	Break rate (no./Km/yr)
41	87	63	100	234	70	4500	1.05
42	21	58	150	64	120	0	0
43	58	106	100	170	100	0	0
44	64	66	80	310	100	0	0
45	66	67	100	70	100	0	0
46	67	71	100	490	100	0	0
47	67	68	100	50	100	0	0
48	68	70	80	250	100	0	0
49	68	91	100	260	100	0	0
50	23	22	100	200	100	0	0
51	23	24	100	90	100	0	0
52	25	24	80	398	100	0	0
53	25	23	100	190	100	0	0
54	26	25	100	50	100	0	0
55	27	26	100	100	100	0	0
56	27	29	100	300	100	0	0
57	30	29	80	210	100	0	0
58	28	27	100	40	100	0	0
59	28	30	80	180	100	0	0
60	33	28	100	190	80	4500	1.1
61	33	15	100	140	80	4500	1.1
62	32	76	100	186	100	0	0
63	32	30	80	176	100	0	0
64	33	31	150	80	100	0	0
65	31	54	150	130	70	6000	0.9
66	31	56	150	490	80	6000	0.75
67	56	78	150	100	90	0	0
68	57	56	100	556	80	4500	0.9
69	57	79	80	416	100	0	0
70	56	55	100	148	100	0	0
71	55	77	100	126	100	0	0
72	50	49	100	450	100	0	0
73	54	53	150	70	100	0	0
74	53	55	100	270	90	0	0
75	53	49	150	190	90	0	0
76	76	49	100	210	100	0	0
77	48	101	150	160	100	0	0
78	45	47	100	226	100	0	0
79	46	45	100	40	100	0	0
80	46	50	100	174	100	0	0
81	51	50	80	100	100	0	0
82	76	51	100	138	100	0	0
83	52	51	100	124	100	0	0

Pipe No.	From Node	To Node	Diameter (mm)	Length (m)	Hazen William C coefficient	Repair cost cost	Break rate (no./Km/yr)
84	34	52	80	286	80	3000	1.2
85	34	33	300	114	70	15000	0.65
86	35	34	300	170	90	0	0
87	35	43	100	340	100	0	0
88	36	35	300	100	100	0	0
89	37	36	300	40	100	0	0
90	97	37	400	140	100	0	0
91	37	38	300	300	100	0	0
92	38	39	150	130	100	0	0
93	39	40	150	214	80	6000	0.95
94	40	41	80	20	80	3000	1.2
95	42	41	150	136	80	6000	0.8
96	39	100	150	590	70	6000	1
97	43	46	100	130	100	0	0
98	42	43	100	84	100	0	0
99	36	42	100	134	100	0	0
100	115	82	80	84	120	0	0
101	82	83	100	364	120	0	0
102	97	96	100	450	110	0	0
103	85	84	100	310	110	0	0
104	102	7	60	170	90	0	0
105	86	102	80	110	120	0	0
106	105	87	100	206	120	0	0
107	88	89	80	240	100	0	0
108	64	88	100	18	100	0	0
109	89	90	100	154	120	0	0
110	90	91	150	120	120	0	0
111	91	69	100	220	100	0	0
112	70	69	60	300	120	0	0
113	106	60	100	150	90	0	0
114	106	90	80	210	120	0	0
115	58	87	100	150	90	0	0
116	58	94	60	196	120	0	0
117	94	92	60	340	120	0	0
118	94	93	60	470	120	0	0
119	101	57	150	124	120	0	0
120	47	101	150	100	100	0	0
121	100	47	150	180	100	0	0
122	100	99	60	280	120	0	0
123	99	57	80	150	120	0	0
124	98	99	80	140	120	0	0
125	103	98	100	986	120	0	0
126	38	103	100	30	120	0	0

Pipe No.	From Node	To Node	Diameter (mm)	Length (m)	Hazen William C coefficient	Repair cost cost	Break rate (no./Km/yr)
127	5	83	80	80	100	0	0
128	6	84	100	90	80	4500	0.95
129	12	85	100	64	80	4500	0.8
130	86	65	100	290	80	4500	0.85
131	104	73	60	250	120	0	0
132	6	104	80	160	100	0	0
133	105	86	100	176	120	0	0
134	16	27	60	180	120	0	0
135	22	95	100	20	90	0	0
136	95	94	60	110	120	0	0
137	96	14	80	60	100	0	0
138	80	97	400	190	100	0	0
139	49	48	150	50	100	0	0
140	49	48	100	50	100	0	0
141	11	108	350	130	120	0	0
142	108	44	100	24	100	0	0
143	33	32	100	54	100	0	0
144	74	81	100	140	100	0	0
145	88	63	100	92	100	0	0
146	110	7	100	110	120	0	0
147	57	111	150	1500	120	0	0
148	17	112	200	170	120	0	0
149	3	80	250	40	120	0	0
150	33	28	150	190	120	0	0
151	11	17	200	80	120	0	0
152	57	79	80	416	120	0	0
153	105	64	150	260	120	0	0
154	64	66	100	310	120	0	0
155	88	89	100	240	120	0	0
156	58	106	100	170	120	0	0
157	114	115	80	96	90	0	0
158	114	4	100	38	90	0	0

Table 6-4 Data for New Pipes of a Moroccan Network

Pipe No.	From node	To node	Length (m)	Pipe No.	From node	To node	Length (m)
1	81	107	470	6	93	92	440
2	78	113	230	7	113	93	180
3	107	82	190	8	113	109	300
4	107	59	190	9	74	81	140
5	92	90	460				

Table 6-5 Node Data of a Morrocan Network

Node No.	Demand (L/s)	Minimum head (m)	Node No.	Demand (L/s)	Minimum head (m)
2	5.07	20	59	0.92	20
3	0.73	20	60	3.9	20
4	0.81	20	61	1.24	20
5	1.33	20	62	1.43	20
6	0.86	20	63	3.72	20
7	0.97	20	64	0.95	20
8	0.73	20	65	4.45	20
9	0.11	20	66	3.61	20
10	1.77	20	67	1.52	20
11	0.26	20	68	1.69	20
12	1.26	20	69	6.49	20
13	3.13	20	70	2.33	20
14	1.05	20	71	1.45	20
15	1.02	20	72	6.39	20
16	0.75	20	73	2.9	20
17	3.7	20	74	1.05	20
18	2.93	20	75	1.21	20
19	3.54	20	76	0.59	20
20	0.7	20	77	0.36	20
21	0.23	20	78	0.33	20
22	1.52	20	79	4.43	20
23	0.81	20	80	0.28	20
24	1.71	20	81	2.87	20
25	2.02	20	82	2.49	20
26	1.35	20	83	3.01	20
27	0.39	20	84	3.72	20
28	0.8	20	85	1.97	20

Node No.	Demand (L/s)	Minimum head (m)	Node No.	Demand (L/s)	Minimum head (m)
29	2.25	20	86	0.67	20
30	2.21	20	87	0.58	20
31	1.67	20	88	0.82	20
32	1.06	20	89	0.76	20
33	0.43	20	90	3.6	20
34	2.21	20	91	2.92	20
35	1.46	20	92	3.39	20
36	0.19	20	93	5.22	20
37	0.31	20	94	2.24	20
38	1.73	20	95	1.16	20
39	6.29	20	96	2.18	20
40	1.4	20	97	0.6	20
41	1.51	20	98	1.36	20
42	2.15	20	99	1.35	20
43	0.88	20	100	1.97	20
44	1.08	20	101	0.42	20
45	1.97	20	102	1.74	20
46	1.39	20	103	2.64	20
47	1.91	20	104	0.59	20
48	1.07	20	105	1.09	20
49	1.25	20	106	1.69	20
50	1.22	20	107	1.43	20
51	2.66	20	108	1.65	20
52	0.24	20	109	2.62	20
53	0.66	20	110	0	20
54	1.55	20	111	4.77	20
55	4.74	20	112	0	20
56	2.12	20	113	0	20
57	6.55	20	114	0	20
58	0.91	20	115	0	20

Table 6-6 Unit Cost of Available Pipe Sizes for the Rehabilitation of a Moroccan Network

Pipe Diameters (mm)	Possible rehabilitation actions		
	relining a pipe ($/m)	replacing a pipe ($/m)	duplicating a pipe or a new pipe ($/m)
80	85	110	100
100	100	190	175
150	150	240	220
200	220	350	320
300	300	600	550
400	410	850	780
500	500	1050	980
600	630	1500	1350

6.7.3 Fast messy GA parameters

Binary coding has been used for solving the optimisation of the rehabilitation of Moroccan network. Two bits have been used for coding the 4 rehabilitation actions and 3 bits have been used for coding the 8 pipe sizes for each of 158 existing pipes. Three bits have been used for coding the 8 pipe sizes for each of the 9 new pipes. Thus 817 binary bits are used for one alternative solution of the Moroccacn network. The optimal solution is sought over 4 eras. Each era started with an initial population of 820 and a juxtapositional population of 1500. The other messy GA parameters used for solving this problem are splice probability $P_s = 0.9$, bit-wise cut probability $P_k = 0.0166$, allelic mutation probability $P_m = 0.01$, genic mutation probability = 0.01 and the maximum generation for each era = 200.

6.7.4 Results

The optimisation model as described above has been established for optimal rehabilitation of the Moroccan network by using the fmGANET. A number of fmGANET runs have been carried out to search for the optimal rehabilitation strategy by using the different penalty cost factors. A set of the optimal or near-optimal solutions have been found. The cost of the optimal rehabilitation solution for the Moroccan network for each of the 8 different penalty factors is given in Table 6-7. It shows that there are slight differences among the optimal solutions by using penalty factors from $550,000 to $750,000 per meter of the excess heads. The greater the penalty factor that was used in the optimisation model, the greater the cost of the least solution found. This is because the large penalty factor forced the genetic algorithm search towards the feasible region. The genetic algorithm operations tended to reproduce more genotypes within the feasible region than outside of the region. It helps to ensure the feasibility of the optimal solution, but requires more search effort to reach the optimal solution as shown in Table 6-8.

The optimal rehabilitation actions and associated pipe sizes for the top 4 solutions (the other solutions are given in Appendix A) are given in Table 6-9. The total costs of these 4 solutions are almost same as shown in Table 6-7, but the optimal rehabilitation actions are quite different as observed in Table 6-9. These different optimal configurations provide engineers and/or decision-makers more options to choose the optimal rehabilitation strategy by using other non-quantifiable engineering criteria.

This case study demonstrates that the fast messy GA is highly efficient at searching for the optimal or near-optimal solution for the discrete optimisation of large-scale water network. The total number of possible solutions for the rehabilitation of the Moroccan network equals 2^{817}, which is approximately 8.74×10^{245}. A complete enumeration of this solution space would consume 2.77×10^{232} centuries of CPU time even if assuming that 10,000 objective evaluations can be done every second. The fmGANET have found the optimal or near-optimal solutions by evaluating about 600,000 alternatives as shown in

Table 6-8. Although it is not possible to prove whether or not the fmGANET has found the global optimal solution, the convergence rates of the fmGANET runs, as shown in Figure 6-7 and 6-8, indicate that the fast messy GA has efficiently improved the process of searching for the global optimal solution. The best cost in each generation has been quickly improved from about 10.0 million down to 1.1 million.

Figures 6-7 and 6-8 also show the typical convergence behaviour of the fast messy GA. The best generation cost was initially very large, but improved along the fast messy GA optimisation. The cost increased in next era due to the population being initialised again. The cost improved fairly quickly by using the best string found in last era as competitive template and kept improving by the fast messy GA optimisation until it reached the maximum number of the generations specified for this era. For the next era, the fast messy GA search started again until it reached the maximum number of eras. The optimal or near-optimal solution was found at the end of fast messy GA search.

Table 6-7 Cost of Optimal Rehabilitation Strategies of the Moroccan Network

Fast messy GA solutions	Penalty factor	Repair cost	Relining or cleaning cost	Replace-ment cost	Duplicat-ion cost	New pipe cost	Total cost
fmGA1	550,000	235,352	41,600	29,020	497,680	309,200	1,112,852
fmGA2	750,000	221,087	105,900	0	519,150	301,100	1,147,237
fmGA3	500,000	215,266	82,500	14,740	526,100	309,200	1,147,806
fmGA4	700,000	230,839	79,500	21,600	518,180	309,200	1,159,319
fmGA5	600,000	224,083	54,600	0	606,480	309,200	1,194,363
fmGA6	650,000	195,863	173,300	12,100	515,880	309,200	1,206,343
fmGA7	800,000	158,192	277,200	79,120	543,570	326,000	1,384,082
fmGA8	1,000,000	213,990	67,510	22,400	614,380	477,500	1,395,780

Table 6-8 Pressure Heads and Excess at Critical Nodes of Moroccan Network (EPANET)

Optimal solutions	Node 59		Node 69		Node 87		Node 111		Evaluations achieved
	head (m)	excess (m)	head (m)	excess (m)	head (m)	excess (m)	head (m)	excess (m)	
fmGA1	20.09	0.09	20.01	0.01	20.13	0.13	20.17	0.17	630,290
fmGA2	20.40	0.40	20.24	0.24	20.02	0.02	20.45	0.45	571,290
fmGA3	20.92	0.92	20.18	0.18	20.03	0.03	20.08	0.08	566,290
fmGA4	20.54	0.54	19.99	**-0.01**	20.02	0.02	20.00	0.00	586,290
fmGA5	20.13	0.13	20.017	0.017	20.01	0.01	20.87	0.87	430,290
fmGA6	20.07	0.07	20.00	0.00	20.20	0.20	19.99	**-0.01**	575,290
fmGA7	20.54	0.54	20.12	0.12	20.02	0.02	20.25	0.25	599,290
fmGA8	20.24	0.24	20.04	0.04	20.48	0.48	21.42	1.42	906,970

Table 6-9 Optimal Rehabilitation Actions and Associated Pipe Sizes of the Moroccan Network

Pipe ID	fmGA1		fmGA2		fmGA3		fmGA4	
	Action	Dia. (mm)	Action	Dia. (mm)	Action	Dia. (mm)	Action	Dia. (mm)
1	duplication	500	duplication	500	duplication	400	duplication	400
2	duplication	80	leave	-	duplication	150	leave	-
3	leave	-	leave	-	duplication	100	reline	200
12	leave	-	leave	-	duplication	300	leave	-
21	reline	100	leave	-	reline	100	reline	100
22	leave	-	leave	-	leave	-	duplication	80
26	leave	-	leave	-	duplication	80	leave	-
31	replace	200	reline	150	leave	-	replace	300
32	leave	-	leave	-	leave	-	duplication	200
37	duplication	200	duplication	300	leave	-	leave	-
39	duplication	80	duplication	80	leave	-	leave	-
42	leave	-	leave	-	leave	-	duplication	300
45	leave	-	reline	100	leave	-	reline	100
54	reline	100	leave	-	duplication	80	leave	-
55	reline	100	duplication	100	leave	-	leave	-
58	duplication	80	reline	100	leave	-	duplication	80
60	leave	-	leave	-	reline	100	leave	-
64	duplication	150	reline	150	reline	150	leave	-

Pipe ID	fmGA1		fmGA2		fmGA3		fmGA4	
	Action	Dia. (mm)	Action	Dia. (mm)	Action	Dia. (mm)	Action	Dia. (mm)
66	leave	-	reline	150	duplication	80	leave	-
73	leave	-	duplication	80	leave	-	leave	-
77	leave	-	leave	-	duplication	80	leave	-
79	leave	-	reline	100	leave	-	leave	-
81	leave	-	leave	-	leave	-	reline	80
88	leave	-	duplication	80	leave	-	leave	-
89	duplication	200	leave	-	reline	300	duplication	150
92	duplication	80	leave	-	leave	-	leave	-
94	leave	-	leave	-	reline	80	leave	-
97	duplication	80	leave	-	leave	-	duplication	150
99	leave	-	leave	-	leave	-	duplication	100
106	duplication	80	duplication	100	duplication	80	leave	-
108	replace	150	leave	-	reline	100	reline	100
110	leave	-	duplication	80	leave	-	leave	-
115	leave	-	leave	-	leave	-	duplication	80
124	leave	-	leave	-	duplication	80	leave	-
126	reline	100	leave	-	leave	-	duplication	100
127	leave	-	leave	-	reline	80	leave	-
128	leave	-	leave	-	leave	-	reline	100
129	reline	100	leave	-	reline	100	duplication	150
130	leave	-	leave	-	leave	-	reline	100
135	leave	-	leave	-	leave	-	reline	100
136	replace	80	leave	-	replace	80	leave	-
142	leave	-	leave	-	replace	80	leave	-
143	reline	100	leave	-	leave	-	reline	100
145	duplication	80	leave	-	leave	-	leave	-
146	leave	-	leave	-	reline	100	leave	-
148	leave	-	leave	-	leave	-	duplication	80
149	leave	-	leave	-	duplication	-	leave	-
150	leave	-	leave	-	leave	-	duplication	80
151	leave	-	duplication	80	leave	100	leave	-
153	leave	-	leave	-	duplication	80	leave	-
158	reline	100	leave	-	reline	100	leave	-
159	new	80	new	80	new	80	new	80
160	new	150	new	150	new	150	new	150
161	new	80	new	80	new	80	new	80
162	new	80	new	80	new	80	new	80
163	new	80	new	80	new	80	new	80
164	New	80	new	80	new	80	new	80
165	New	150	new	100	new	150	new	150
166	New	80	new	80	new	80	new	80

Pipe ID	fmGA1		fmGA2		fmGA3		fmGA4	
	Action	Dia. (mm)	Action	Dia. (mm)	Action	Dia. (mm)	Action	Dia. (mm)
167	New	80	new	80	new	80	new	80
total cost	1,112,582		1,147,237		1,147,806		1,159,319	

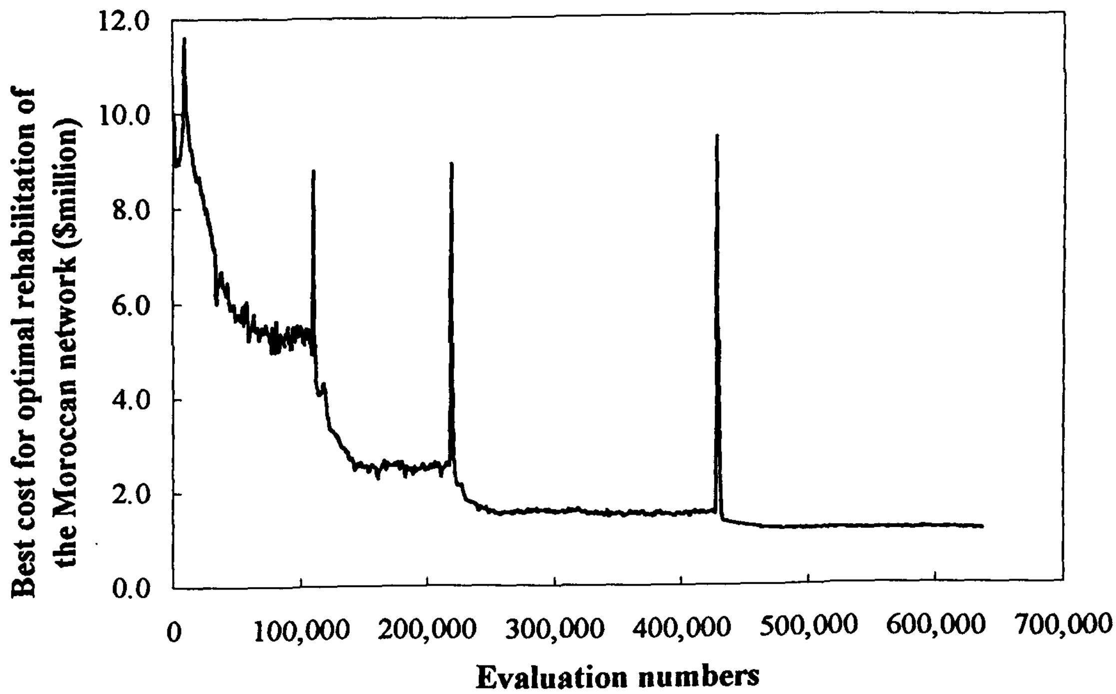


Figure 6-7 Convergence Rate of fmGA1 Solution for Optimal Rehabilitation of the Moroccan Network

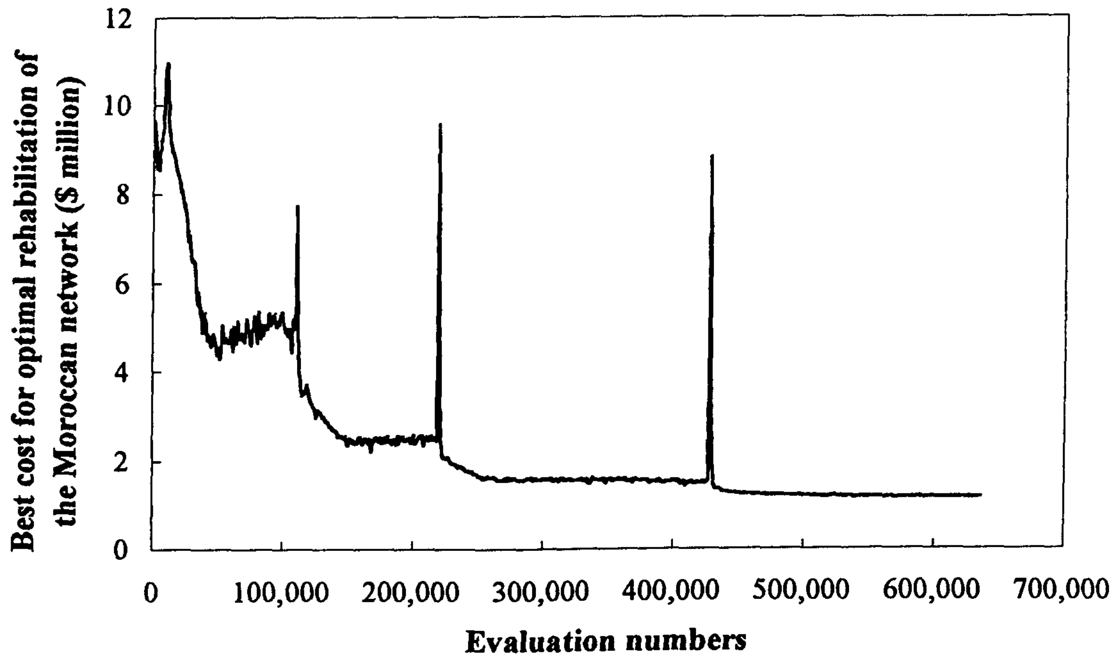


Figure 6-8 Convergence Rate of fmGA2 Solution for Optimal Rehabilitation of the Moroccan Network

6.8 Summary

In this Chapter, the standard genetic algorithm paradigm has been compared with the messy genetic algorithm for the two reservoir network. The standard GA defines the relations and classes implicitly by using a fixed-length representation. It combines the relation space, class space and sample space all together, thus a poor and noisy decision process occurs. Increasing the tournament selection pressure can improve the search efficiency, but too much pressure may lead the search to a local optimal.

Messy GAs emphasise searching for appropriate relations. The original messy GA uses a competitive template and explicit enumeration of good classes or building blocks, to ensure correct decision making. However, using an initialisation procedure where building blocks are explicitly enumerated essentially prevents the messy GA from being applied to highly dimensional problems. The probabilistically complete initialisation and the building block filtering process are introduced into the fast messy GA to detect better classes from better relations. A comparison study of the messy GAs for optimisation of pipeline networks has been carried out and shows that the fast messy GA is the most efficient algorithm among the genetic-based search paradigms. It eliminates the major bottleneck of the original messy GA—the explicitly enumerative initialisation and thus provides a promising optimisation algorithm for solving highly dimensional discrete optimisation problems.

The fast messy GA has been implemented and integrated with hydraulic network solver EPANET. The integrated approach has been applied to the optimal rehabilitation of a large water distribution system in Morocco. This application has demonstrated that the fast messy GA is very efficient at solving large-scale optimisation problems. A set of optimal or near-optimal solutions have been found by employing different penalty factors. The greater the penalty, the greater the optimal cost and also the more evaluations of objective functions, however too small a penalty factor may not be able to ensure the feasibility of the solutions. As observed in the optimisation of the water distribution systems, the optimal solution is often located at the edge of the feasibility boundary since there are always a few of nodes critical for the solution. Thus it may enhance the efficiency and effectiveness of the genetic algorithm search procedure to adapt the population of a GA search towards the edge of the boundary of the feasible and infeasible regions. This will be explored in next Chapter.

7. BOUNDARY SEARCH OF GENETIC ALGORITHMS BY SELF-ADAPTIVE PENALTY

7.1 Introduction

It is generally difficult to solve constrained nonlinear optimisation problems that are often found in engineering design disciplines. A commonly used method for handling the design constraints of a problem in constrained optimisation is by use of a penalty function. Genetic algorithms outperform gradient-based optimisation methods at dealing with the cases where the optimisation problems are characterised by a number of peaks. Much effort is required for developing a penalty function approach for genetic and/or evolutionary algorithms to solve the constrained nonlinear optimisation problems. No method is completely robust and efficient at solving the constrained nonlinear optimisation problem in a general sense. Particularly, not a method is efficient and effective at searching for the boundary area between feasible and infeasible regions of the search space of the constrained nonlinear optimisation problem. This ability of the genetic algorithms searching for the boundary is important in the case of optimisation problems with nonlinear equality constraints while being particularly important for the optimisation of water distribution systems. This is because the optimal or near-optimal solutions are always located at the boundary of the active nonlinear constraints of the minimum hydraulic pressure head requirements at some of the nodes in the water distribution system. These nodes are often called critical nodes for the optimisation of the design and rehabilitation of the networks.

A penalty function method for handling the constraints in the GA distorts the objective function to force the search towards feasibility. Typically, the approach requires one (or more than one) penalty factors to construct a weighted sum of the constraint violations and the objective functions. It is the penalty factor that defines the degree of the distortion of the objective function, and consequently has a major influence on the performance of the GA search. Different penalty factors may lead the search process to a different solution. Tuning the penalty factor is not only time consuming, also requires a trial and error procedure. In this chapter, a strategy for the co-evolution of the value of the penalty factor is proposed by coding the penalty factor into each of the genotype in addition to the normal chromosome coding. The idea is to evolve or to self-adapt the penalty factor for forcing the genotypes into the region of the boundary of the feasible and infeasible regions along the GA optimisation horizon. The genotypes that have encoded penalty factors and produce fitter offspring will survive longer, consequently, the preferred penalty factor value will spread through the population and be evolved over generations. The self-adaptive penalty method is applied in this Chapter to a well-studied example of optimisation of water distribution system to demonstrate the effectiveness of the method. The numerical results have shown the ability of the self-adaptive penalty approach of the genetic algorithm to reach the optimal and near-optimal solutions by searching for the boundary of the feasible and infeasible regions of the search space.

7.2 Constrained Optimisation Problems

A constrained nonlinear programming problem (NLP) is to search for the global optimum solution **x** subject to a set of nonlinear constraints. It can be generally written as

search for $\quad x = (x_1, x_2, \ldots, x_n) \in \mathbf{R}^n$

such that $\quad minimum\ f(x)$

subject to $g_j(x) \geq 0$ $\quad j = 1, 2, \ldots, J;$

$$x_i^{min} \leq x_i \leq x_i^{max} \qquad i = 1, 2, \ldots, n;$$

where x_i^{min} = the lower bound of the i-th variable; x_i^{max} = the upper bound of the i-th variable; $f(x)$ is the objective function defined on the n-dimensional Euclidean space $\mathbf{R}^n$, and the constraint functions $g_j(x)$ define the feasible region of the search space. In general, it is impossible to develop a deterministic method for the NLP problem in the global optimisation category. One of the main difficulties in solving the NLP problem is that of local optima. Local optima satisfy just the mathematical requirements on the derivatives of the functions, and many optimisation techniques based on gradient methods result in obtaining a local optimal solution only.

Evolutionary algorithms are global methods that aim at complex objective functions (eg., those that are nondifferentiable or discontinuous). However, most research on applications of evolutionary computation techniques to NLP problems has been concerned with complex objective functions but no constraints. Many of the numerical optimisation functions used by various researchers during the past 20 years did not include any constraints (apart from specified domains of variables). More recently, a set of constrained numerical optimisation functions have been studied by evolutionary algorithms using different penalty functions and specific genetic operators (Michalewicz and Schoenauer 1996). But it has not been clear what makes nonlinear constrained optimisation problem hard to solve. A number of possible difficulties are summarised as follows (Michalewicz and Schoenauer 1996).

- The ruggedness of the unconstrained fitness landscape is certainly the first important characteristic influencing the overall problem difficulty. For instance, linear or convex objective functions will always result in easier constrained problems than chaotic functions.

- The sparseness of the feasible region is indeed a crucial factor of difficulty. In some problems, finding a single feasible point is difficult. Moreover, a large slope of the constraints on the border of the feasible region is another way in which a constrained problem can make the penalty method difficult to apply.
- A high ratio between the highest global optima of the objective function (on the whole domain where it is defined) and the optima of the constrained function (e.g., the local optima of the objective function in the feasible region), as well as a small distance between these global optima and the feasible region, can also make the constrained optimisation problem almost intractable for penalty methods.
- The number of active constraints at the optimum is of course important: the more constraints that are active at the optimum, the more likely to succeed are algorithms searching close to the boundary.

It is hard to predict (1) the ruggedness of the unconstrained region; (2) the spareness of the feasible region and (3) the number of active constraints. It seems worthwhile to resort to the metaphor of the evolutionary techniques, the self-adaptation of the search procedure for handling nonlinear constraints in constrained GA optimisation.

7.3 Conventional Method for NLP

A favourite method for solving the constrained optimisation problem is to reduce it to a sequence of unconstrained problems that are easy to solve. The idea is that each of the easy problems is solved and that the sequence of solutions of the easy problems will converge to the solution of the original difficult optimisation problem. This is often refereed as to sequentially unconstrained minimisation technique (SUMT) by Fiacco and McCormick

(1968). A penalty function is traditionally introduced to convert a constrained NLP into a sequence of unconstrained nonlinear optimisation as follows.

search for $x = (x_1, x_2, \ldots, x_n) \in \mathbf{R}^n$

minimising $obj(\mathbf{x}) = f(x) + penalty(x)$

subject to $x_i^{min} \le x_i \le x_i^{max}$ $i = 1, 2, \ldots, n;$

where F is the feasible region of the search space.

The penalty functions are designed to penalise the infeasible solutions, namely, to force the search towards the feasible solution region by solving an unconstrained optimisation problem. The penalty function is based on the distance of a solution from the feasible region, it is zero if no violation occurs and is a positive otherwise. The penalty of an infeasible solution is usually calculated based on a function of the distance between the solution and the feasible region. The penalty function is often defined by a weighted sum of the amount of actual constraint violation. The weight on the penalty term is usually called penalty factor. There are number of different ways of constructing the penalty function which gives rise to different performance of the constrained nonlinear optimisation techniques.

7.3.1 Traditional penalty methods

The penalty function $penalty(x)$ is traditionally constructed from the constraint functions in such a way that as the penalty factor approaches zero (or infinity) the unconstrained optimum of the augmented objective function converges to the solution of the original NLP. Two basically different penalty techniques and a hybrid method were suggested by Haug and Arora (1979).

(1) ***Interior penalty method*** This technique defines the penalty function given as

$$penalty(x) = r\left(\sum_{j=1}^{J} - \frac{1}{g_j(x)}\right) \tag{7.1}$$

where r = penalty factor and $g_i(x)$ = the j-th constraint. The penalty factor is selected as a sequence r_i that approaches zero while the solution of NLP is approached. This penalty function is continuous within the feasible region, but constructs an objective surface that is infinitely high at the boundary of the feasible and infeasible. It is the infinitely high objective value that forces the search procedure to be within the feasible region. Thus this technique requires the initial point within the feasible region, but initialising a feasible point is normally "hard" in most cases of NLP.

(2) *Exterior penalty method.* Unlike the interior penalty method, initial points for the exterior penalty method are not required to be within the feasible region of NLP. The idea in the exterior penalty method is to add to the original objective function a penalty for the points outside the feasible region and zero inside the region. The exterior penalty function can be given as

$$\text{penalty}(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \in F \\ t\sum_{j=1}^{J}\left[g_j(\mathbf{x})\right]^2 & \mathbf{x} \notin F \end{cases} \tag{7.2}$$

where F = the feasible region of the search space and t = penalty factor that is selected as a sequence of t_i, $t_i \to \infty$ while the points approach to the optimum of NLP. This penalty function discourages the minimum of the new augmented objective function from being too far from the constraints set. It handles equality as well as inequality constraints without any difficulty.

(3) ***Mixed interior-exterior penalty method.*** A combination of the interior and exterior penalty methods are applied to NLP with inequality constraints and equality constraints. The interior method cannot be used if the interior of the constraint set is empty, such as a single equality constraint. The exterior method cannot used if some constraints are not well defined. Thus the mixed method allows for treatment of problems that may have both these undesirable features and cannot be treated by either pure interior or exterior methods.

These traditional penalty methods are often used for point based optimisation techniques. It establishes a theoretical and empirical foundation for the constrained genetic algorithm optimisation. The penalty methods have been extended in evolutionary algorithms for handling the nonlinear constraints.

7.4 Methods in Evolutionary Algorithms for NLP

A great deal of research work has been done to develop methodologies for the constrained nonlinear optimisation using evolutionary algorithms. They are generally classified as

- penalty method;
- searching for the feasible solutions;
- hybrid method.

These methods were generally reviewed by Michalewicz and Schoenauer (1996) in context of numerical function optimisation. They are discussed in this section in a framework of engineering optimisation.

7.4.1 Penalty methods

The penalty function methods developed in evolutionary algorithms are mainly as follows.

(1) Static penalty — Homaifar, Lai and Qi (1994) suggested this method by using a penalty function as

$$\text{penalty}(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \in F \\ \sum_{j=1}^{J} \gamma_{ij}\left[g_j(\mathbf{x})\right]^2 & \mathbf{x} \notin F \end{cases} \tag{7.3}$$

where γ_{ij} = the penalty coefficient for constraint j at violation level i and i = 1, 2, ..., l. The static penalty method assigns one penalty factor to each of l levels violation for each of J constraints. The drawback of this method is that a large number of penalty coefficients are required even for a small optimisation problem. It is hardly applicable for solving the problem of engineering optimisation due to large number of constraints.

(2) Dynamic penalty — It was proposed by Joins and Houck (1994). The penalty is evaluated by the formula given as

$$penalty(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \in F \\ (C \times t)^{\alpha} \sum_{j=1}^{J}\left[g_j(\mathbf{x})\right]^{\beta} & \mathbf{x} \notin F \end{cases} \tag{7.4}$$

It assumed that the penalty on a infeasible solution increases as the generation (iteration) increases. The penalty is multiplied by $(C * t)^{\alpha}$, where C = 0.5, $\beta = \alpha = 2$, t is the generation or iteration. This method gave very good results, in general, for quadratic objective functions. In most numerical experiments, it gave good results in early generations, but the penalty increased too fast to be useful, as a result the search is hard to escape from a local optima.

(3) Annealing penalty — This approach is based on dynamic penalty method and was introduced by Michalewicz and Attia (1994) and Michalewicz (1996). The penalty cost is given as

$$\text{penalty}(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \in F \\ \frac{1}{2\tau} \sum_{j=1}^{J}\left[g_j(\mathbf{x})\right]^2 & \mathbf{x} \notin F \end{cases} \tag{7.5}$$

where τ is called *temperature*. The penalty increases as the *temperature* is cooling down from a starting temperature such as 1.0 to a freezing temperature such as 0.00001. This method was used in conjunction with a set of special operators in Genocop (for Genetic algorithm for Numerical Optimisation of Constrained Problem) by Michalewicz and Janikow (1991). These special operators maintained the feasibility of linear constraints, and the penalty cost is calculated by decreasing the values of temperature. Obviously, this method requires a starting and a freezing temperature and a cooling scheme to decrease the temperature. The optimal solution was reported to be very sensitive to the cooling scheme.

(4) Adaptive penalty — This method (Bean and Hadj-Alouane 1992) used the same penalty evaluation as the traditional exterior penalty method. The penalty factor was adjusted by a rule similar to "1/5 success rule" in the evolutionary strategy (Back et al. 1991). The penalty decreased for $(t+1)$st generation if all the best individuals in t generations were feasible, and increased if all the best solutions in last k generations are infeasible. The penalty remained the same value if some solutions were infeasible and some of them were feasible.

(5) Death penalty — The death penalty method (Back et al. 1991) just simply rejects infeasible solutions. The method provided good results for some problems, but was not stable as others (Michalewicz 1995). It generally gave a poor performance.

7.4.2 Search for feasible solutions

The methods emphasis the search for feasible solutions only. There are three methods as discussed as following.

(1) Behavioural memory method — This method (Schoenauer and Xanthakis 1993) considers the constraints in a sequence. Switching from one constraint to another is based on arrival of a sufficient number of feasible individuals in the population. It requires a linear order of all the constraints and a flip threshold that is the percentage of the feasible individuals in the population for a specific constraint.

The algorithm starts evolving the population until the flip threshold for constraint j is satisfied, then the next constraint is considered. The population is evolved until the flip thresholds for all the constraints are satisfied. Then the population is optimised by using death penalty approach. Obviously the order of the constraints could have some influence on the optimal solution, and also the solution quality is governed by the death penalty method used at the final stage of the optimisation.

(2) Superiority of feasible points — This method (Powell and Skolnick 1993) was based on the classical penalty method with one exception in which any feasible solution is better than any infeasible solutions. The difficulty with this method sometimes is to locate feasible solution.

(3) Repairing infeasible points — This method was proposed by Michalewicz and Nazhiyath (1995). The initial population P_s is generated to meet linear constraints. The P_s is divided into two sets namely P_r which is the so-called reference population satisfying all constraints and P_f the set of infeasible solutions. Any point s from P_f is repaired by randomly projecting a random point between s and r (r is selected from P_r), that is, $z = as + (1 - a)r$ until a fully feasible point is generated. This method always returns a feasible solution. The feasible point is only evaluated by the objective function without being distorted by the penalty function. For engineering optimisation problems, the repair technique can be essential to guarantee an efficient search procedure, and also the technique needs to be modified to cope with discrete variables that are often found in real world problems such as optimisation of water distribution systems.

7.4.3 Hybrid methods

It has been recognised that hybridising EA and/or GA with traditional optimisation methods improves the efficiency of the search procedure. An interesting approach developed for handling the constraints, which has been successfully applied to some engineering optimisation problems, is Vector Evaluated Genetic Algorithm (VEGA).

The VEGA method was proposed by Schaffer (1985). It treated the penalties of all the constraints as a vector of multiobjective function along with the objective function. Multiobjective techniques are applied to minimise all the components of the vector. This approach was applied by Parmee and Purchase (1994) in the development of a technique for constrained optimisation of engineering design. Surry et al. (1995) modified VEGA into a two-objective optimisation problem by ranking all the population members based on the constraint violation. The rank r and the objective value lead to a two-objective optimisation problem. It performed well for the optimisation of gas supply networks.

Of all the methods discussed above, the penalty methods are generally applied to solving NLP in evolutionary algorithms. Each of the methods, however, involves one (or more than one) penalty factor to be tuned to achieve the best performance of the genetic algorithms. However, tuning the penalty factor produces a dilemma, that is, too small a penalty may lead the search to infeasible region, but too high penalty may restrict the GA search inside the feasible region and forbid any shortcut across the infeasible region. As a result the GA may fail to reach the optimum solution. The idea of the boundary search method as developed in this research is to self-adapt the penalty factor by co-evolution of the factor in such a way that the GA population is adjusted (forced) to search the boundary of the feasible and infeasible regions. Thus it is believed that the performance of GA search can be improved for NLP problems with the optimal solution at the boundary.

7.5 Boundary Search GA by a Self-adaptive Penalty

7.5.1 Boundary optimisation problem

Optimal solutions of many of the nonlinear constrained optimisation problems are located at the boundary of the constraint set. For the optimisation problems given as (Zhu et al. 1984):

search for x

such that $\textit{minimum} \;\; obj(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i x_1^{\beta_{i1}} x_2^{\beta_{i2}} \cdots x_n^{\beta_{in}}$

subject to $g_j(x) \le 0, \; j = 1, \ldots, J.$

where α_i, β_{ij} = constant coefficients and $\beta_{ij} \ge 0$. It has been generally proved (Zhu et al. 1984) that the optimal solution is at the boundary of the constraint set if there is an optimal solution. Thus searching the boundary of the constraint set is essential to reach the optimal solution of this type of the optimisation problems.

7.5.2 Boundary search

Conventional optimisation and heuristic methods recognised the need for searching the boundary of the constraint set. A boundary search method (Zhu et al. 1984), based on the complex method of Box (1965), was proposed for the optimisation problem in the case of the optimal at the boundary. This method was successfully applied to optimisation of design of concrete arch dams. Wu and Whang (1992) employed this method to solve a problem of fuzzy optimisation, which quantified subjective uncertainties of the active constraints in conventional engineering optimisation problems. A heuristic approach for constrained optimisation was developed in conjunction with the strategy of scatter search (Glover, 1977). This approach was more recently applied to a variety of problem settings in combinatorial and nonlinear optimisation by Glover & Kochenberger (1995).

Although it has been realised (Michalewicz and Schoenauer 1996) that evolutionary computation techniques have huge potential for incorporating specialised operators that search the boundary of feasible and infeasible regions in an efficient way, a method has not yet been developed to be generally effective for the constrained engineering optimisation. A common situation in any constrained optimisation problems is that some constraints are active at the global optimum. This optimum thus lies on the boundary of the feasible space. Conversely, it is commonly acknowledged that restricting the size of the search space in evolutionary algorithms (as in most search algorithms) is generally beneficial. Hence, it seems natural in the context of constrained optimisation to restrict the search for the solution to the boundary of the feasible part of the space. Two specific evolutionary algorithms (Schoenauer and Michalewicz 1996) searching for boundary of the feasible regions have been designed and tested on two continuous numerical functions. The algorithms start initialising feasible solutions and generating the next generation of the individuals by using the specific crossover and mutation, which keep the offspring on the surface (boundary) of the constraints. The results obtained show that the algorithms designed specifically for these two numerical functions are very effective and efficient at reaching the global optima. Some discussion was also given on the design of general evolutionary operators searching for the edges, but it is highly unlikely to construct such a type of operators by analytical approach. This is due to the nature of the high nonlinearity of the search problems with implicit constraints in engineering optimisation.

7.5.3 Self-adaptive penalty

The co-evolutionary (or self-adaptation) metaphor was originally introduced by Bäck et al. (1991) into evolutionary strategy to evolve the mutation probability. A similar approach is employed to self-adapt the penalty factor to preserve the population of genotypes close to the boundary. The penalty factor is encoded onto each member of the population and allowed to evolve. The genotypes that have encoded penalty factors and produce fitter offspring will survive longer, consequently, the preferred penalty factor will spread through the population and be evolved over generations.

Co-evolution of the penalty factor is implemented as follows. For any given problem, a genotype (chromosome) contains its normal solution coding, and also the coding of the penalty factor. As shown in Figure 7-1, the sub-string coding the penalty factor is attached to the chromosome of the problem solution. The penalty coding is mapped onto a prespecified range of the penalty factor given as

$$\gamma_n = \gamma^{min} + \frac{\gamma^{max} - \gamma^{min}}{b^{\gamma\gamma} - 1}\left[\sum_{dex=1}^{rr} a^r_{n,dex} b^{dex-1}\right] \tag{7.6}$$

where γ_n = the penalty for the genotype n; γ^{min} = the lower bound of penalty factor; γ^{max} = the upper bound of the penalty factor; $a^r_{n,dex}$ = the *dex*-th bit of the sub-string coding the penalty factor for genotype n; $\gamma\gamma$ = the number of the bits coding the penalty factor; b = 2 for binary coding or 10 for integer coding. The mapped factor will be used in the fitness evaluation of the genotype. Thus the fitness is not only contributed to by the solution coding, but also by the penalty factor coding.

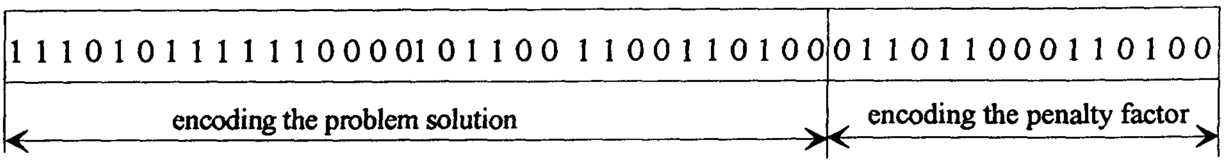


Figure 7-1 Genotype Representation of Co-evolution of Penalty Factor

The scheme of co-evolution of the penalty factor has been implemented within the fast messy GA described in Chapter 6. It has been applied to optimisation of water distribution system to test the effectiveness of the co-evolutionary penalty strategy. The test results of the optimisation indicate that the GA search procedure favoured the lower bound γ^{min} of the penalty factor. After a few number of generations, the lower bound of the penalty dominated the genotype population. Too small a value of γ^{min} led the population far away from the feasible region. A large value of γ^{min}, however, forced the population to converge to a local optimal within the feasible region. Ideally, the lower and upper bounds of the penalty are to be adapted in such a way that the population is maintained searching for the boundary of the feasible and infeasible regions. Thus it will improve the effectiveness of the GA to reach the optimal solution located at the boundary.

A heuristic rule has been developed in this research to adapt the penalty bounds to keep the GA population search for the optimal at the boundary. A ratio of the number of feasible solutions to the sum of the feasible and infeasible solutions is suggested to measure the degree of the genotype allocation in the search space. If the ratio is about 0.5, it simply means that the population is at the boundary of the feasible and infeasible regions. Thus the rule for adapting the bounds of penalty factors is formulated as follows.

Increase the lower and upper penalty bounds by 20% of the upper bound if the ratio of the number of the feasible solutions to the sum of the feasible and infeasible solutions in every certain number (say 20) of generations is less than 30%;

OR

Decrease the lower and upper penalty bounds by 20% of the lower bound if the ratio is greater than 50%.

This rule and the co-evolution of the penalty factor have been implemented and applied to the optimisation of water distribution systems in the following section.

7.6 Boundary fmGA Optimisation of Water Distribution Systems

It has been observed that the optimal solution of design and/or rehabilitation of water distribution systems is found at the boundary of hydraulic pressure constraints of the critical nodes. The optimisation model formulated for the design and rehabilitation of a water distribution system in this study or many others generally falls into the category of a boundary optimisation problem. It is believed that the strategy of searching the boundary of the hydraulic pressure constraints will improve the efficiency and effectiveness of the optimisation procedure.

The self-adaptive penalty described above has been implemented into fmGANET developed in Chapter 6. The penalty function employed for the constrained optimisation of water distribution systems is similar to the exterior penalty method and given as

$$penalty(\vec{D},\vec{E}) = \gamma\left\{\max_{i=1}^{I}\left[\max_{j=1}^{J}\left\{0, H_j^{min} - H_{ij}\right\}\right]\right\} \tag{7.7}$$

where γ = the penalty factor that is to be adapted along the GA optimisation. To demonstrate the application of the boundary search strategy, the New York Tunnels problem has been chosen for this application. Although this problem may not represent all the characteristics of optimisation of water distribution systems, it has been very well studied in literature and provides an excellent example to investigate the behaviour of the boundary GA search strategy.

7.6.1 Results for a case study

The fast messy GA implemented within the fmGANET has been employed for the study of application of boundary GA search strategy. The genotype coding scheme is the same as used in Chapter 3, except additional 4 binary bits are used for coding the penalty factor and attached to the genotypes of the problem solutions. Thus the problem length of the chromosome increases from 84 to 88 binary bits. To investigate the convergence behaviour of the boundary search strategy, the maximum number of eras of 10 has been used in the messy genetic algorithm with a juxtapositional population size of 200. The other messy GA parameters are the same as in Chapter 3.

Table 7-1 Comparison of Conventional GA Optimisation with Boundary GA Optimisation for New York Tunnels Problem

Node ID	Minimum required head (ft)	Fast messt GA with fixed penalty factors per ft				Fast messy GA with adaptive penalty factors per ft		
		$\gamma = 2$ million	$\gamma = 5$ million	$\gamma = 7$ million	$\gamma = 11$ million	initial range [1, 10]	initial range [1, 50]	initial range [0.2 10]
16	260.00	259.14	259.84	261.27	260.40	260.52	260.52	260.52
	Excess =	**-0.86**	**-0.16***	1.27	0.40	0.52	0.52	0.52
17	272.80	271.46	272.63	272.80	272.85	272.86	272.86	272.86
	Excess =	**-1.34**	**-0.17**	0.00	0.05	0.06	0.06	0.06
19	255.00	254.24	254.82	255.47	255.00	255.71	255.71	255.71
	Excess =	**-0.76**	**-0.18**	0.47	0.00	0.71	0.71	0.71
Cost ($million)		32.33	37.62	39.42	39.69	38.80	38.80	38.80
Achieved at evaluation numbers		15,513	44,562	21,245	18,876	22,508	22,508	51,497

*A negative value of the pressure head excess means a constraint violation, and the Hazen-William formula used for this case study is given as $h_f = 4.7291L(Q/C)^{1.852}D^{-4.8704}$.

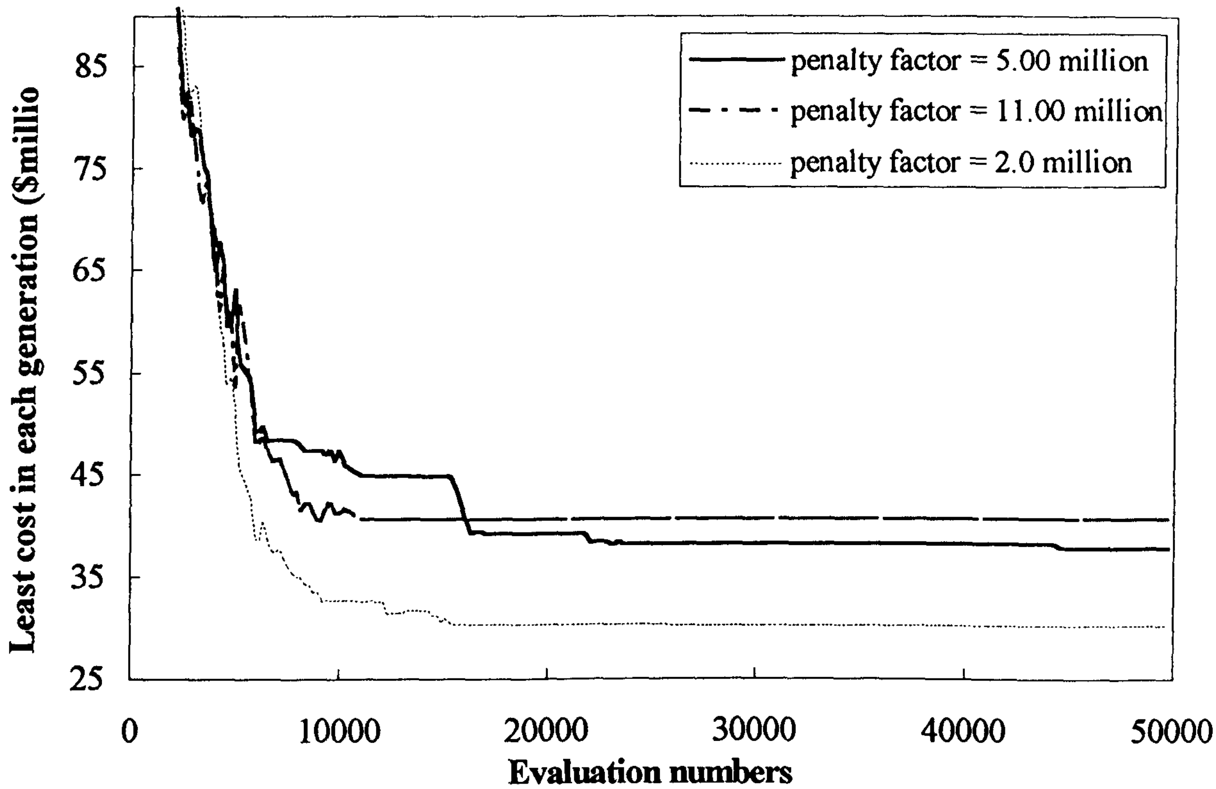
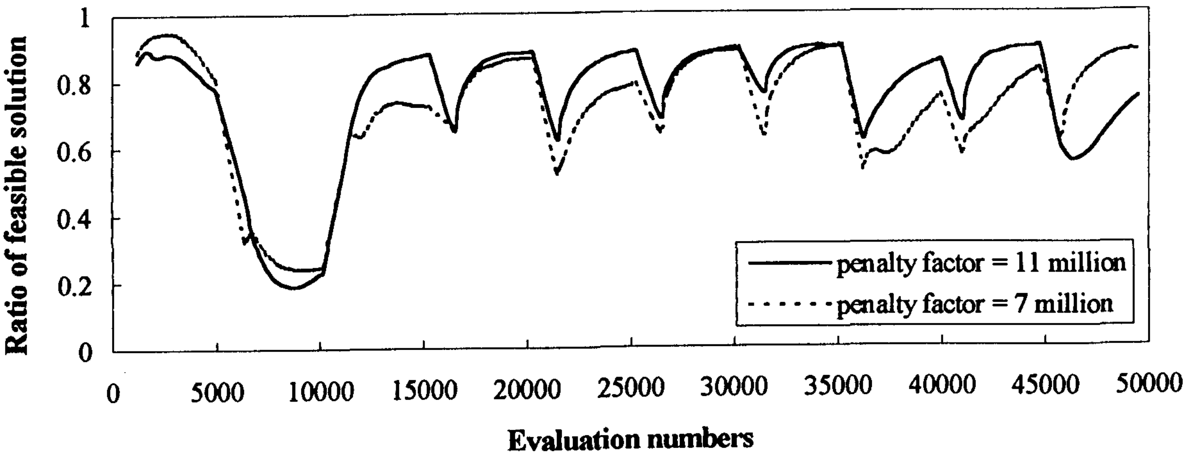


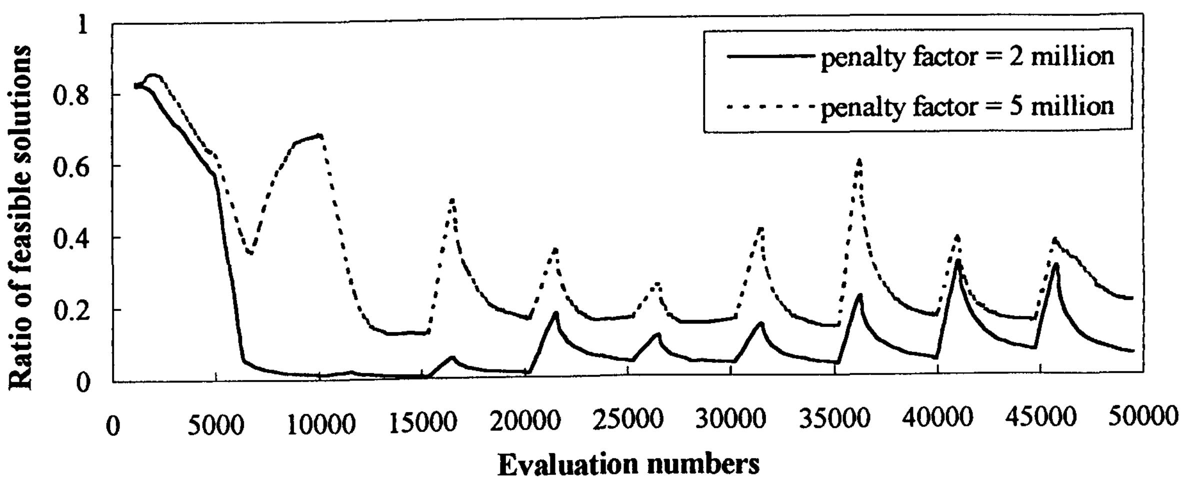
Figure 7-2 Convergence Behaviour of GA Optimisation of New York Tunnels Problem with Conventional Penalty Method

The optimisation of New York tunnels problem has been carried out by using the fmGANET using a conventional penalty approach of fixed penalty factor and the boundary search strategy by adaptive penalty factor. For the conventional penalty optimisation, Four different penalty factors of $11,000,000, $7,000,000, $5,000,000 and $2,000,000/ft have been used for the investigation, while three different ranges of the penalty factor were used for the GA boundary optimisation by adaptive penalty factor. The results obtained by both approaches are compared in Table 7-1. The optimal solutions are given in Appendix B.

The fast messy GA identified the same critical nodes in a similar way as many other studies in literature. The results show that the fixed penalty factor of about $7,000,000/ft is the best parameter for the optimisation of New York tunnels problem. The larger the penalty factor such as $11,000,000/ft, the sooner the search procedure converges to the solution, but it is not the optimal solution of the problem. Conversely, the small penalty factor such as $2,000,000/ft led the GA search to the infeasible local optimal solution after a few generations. To the contrary, the adaptive penalty optimisation has been shown very effective for the different ranges of the penalty factor. The study of a conventional penalty optimisation has indicated that the penalty factor for the New York tunnels problem was around $5,000,000/ft, thus a normal penalty factor range of [1, 10] million was first used to test the boundary GA optimisation strategy. The optimal solution has been achieved after 22,508 evaluations. The other two runs, one with the scaled-up upper bound of the normal penalty range and the other with the scaled-down lower bound of the normal penalty range, have also been carried out. The results show that the optimal or near-optimal solution has been achieved for both the scaled-up penalty range and the scaled-down penalty range. The fast messy GA optimisation using adaptive penalty factor is further demonstrated by investigating the convergence behaviour of the boundary search of the GA optimisation procedure. Comparison of the convergence rate of the adaptive approach with that of conventional penalty approach will provide more understanding about why the self-adaptive approach is more effective than the fixed penalty factor method. This will be discussed in the following sections.



(a) Feasibility of GA population with penalty factor of 7 and 11 million/ft



(b) Feasibility of GA population with penalty factor of 5 and 2 million/ft

Figure 7-3 Population Trace of GA Optimisation of New York Tunnels Problem with Conventional Penalty Methods

7.6.2 Convergence behaviour

(1) ***Without a self-adaptive penalty***

The convergence behaviour of the fast messy GA optimisation with the conventional penalty approach is given in Figure 7-2. The fast messy GA improves the search process very efficiently at early stage of the optimisation and slowly converges to a solution. It has been observed that a big penalty factor may force the GA to select more genotypes within the feasible region of the search space as shown in Figure 7-3(a), but the search converged to an non-optimal solution. The small a penalty factors produce more points within the infeasible region. But too small penalty factor such as $2.0 million/ft yielded a low feasibility of the population as shown in Figure 7-3(b), it consequently converged to a infeasible solution as shown in Figure 7-2 and Table 7-1.

The conventional penalty optimisation approach is not able to adapt the feasibility of the population. As shown in Figure 7-3, it either maintains a high or low feasibility ratio of the population except the variations generated by the random initialisation in each of 10 eras of the fast messy GA. For the boundary optimisation problem such as the optimisation of water distribution systems, adaptation of the population feasibility is essential to guarantee the search to reach the optimal solution. This has been demonstrated by the convergence behaviour of the fast messy GA boundary search through the self-adaptive penalty.

(2) ***With a self-adaptive penalty***

The self-adaptive penalty optimisation of New York tunnels problem has been observed to be effective and efficient at searching for the optimal solution. Two different convergence patterns have been noticed as in Figure 7-4 and described as follows.

As shown in Figure 7-5 and Figure 7-6(a), for the penalty range from 1.0 to 10.0 million/ft, the fast messy GA search started with an average penalty factor about 5.0 million/ft and a very high feasibility ratio of the population. By the rule of penalty bound adaptation, the bounds of penalty factor decreased about 20% of the lower bound of 1.0 million/ft after 20 generations (in about 5,000 evaluations). Thus the penalty factor decreased, as shown in Figure 7-5, the feasibility ratio of the population decreased as shown in Figure 7-6(a), the generation best solution was improved so much that the solution became infeasible as shown in Figure 7-4. The penalty factor range was then adjusted by increasing both the lower and upper bounds by 20% of the current upper bound value of the penalty factor after every 20 generations. Both the average penalty factor and the feasibility ratio of the population increased, and the GA was adapted towards the feasible region of the search space. Thus the generation best solution was found within the feasible region. As the penalty is self-adapted according to the feasibility ratio of the population, the fast messy GA search converged to the optimal solution.

For the scaled-down penalty range from $0.2 to 10 million/ft, the search followed the same pattern as the normal penalty range. Although the lower bound of the penalty was scaled down from 1.0 million to 0.2 million the search still started with quite high feasibility ratio of the population as shown in Figure 7-6(a). This is because the same random seed as the normal penalty range has been used for this case study. The fast messy GA has been self-adapted towards the boundary of the hydraulic pressure requirements. The same optimal solution has been found in the case of scaled-down penalty factor range.

A different convergence pattern has been observed for the scaled-up penalty factor range from 1.0 to 50.0 million, as shown in Figure 7-4. The fast messy GA search started with an average penalty factor of about 30.0 million, as shown in Figure 7-5. Due to the initial high feasibility ratio of the genotype population, the penalty factor was self-adapted by decreasing the factor bounds until the feasibility ratio was low. As shown in Figure 7-6(c), the GA population feasibility increased as the penalty was adapted slightly greater and greater as shown in Figure 7-5. The generation best is improved as the penalty is adjusted. A near-optimal solution has been found for the case of scaled-up penalty range.

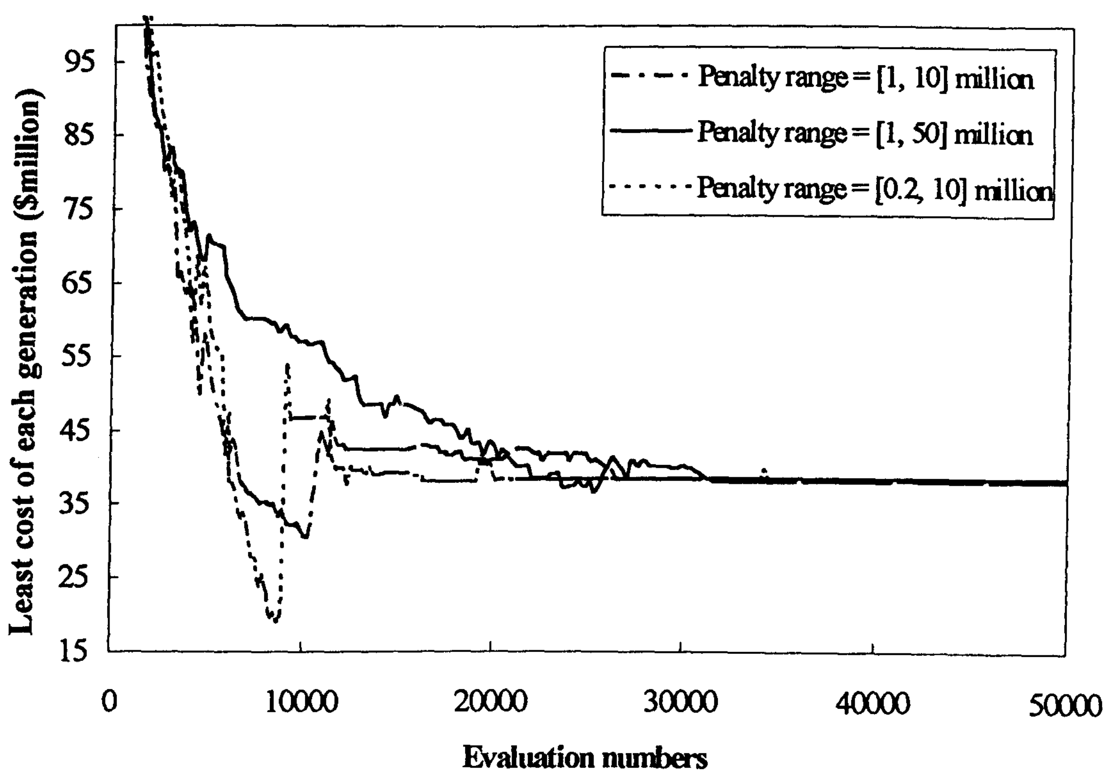


Figure 7-4 Convergence Behaviour of Boundary Search GA Optimisation of New York Tunnels Problem

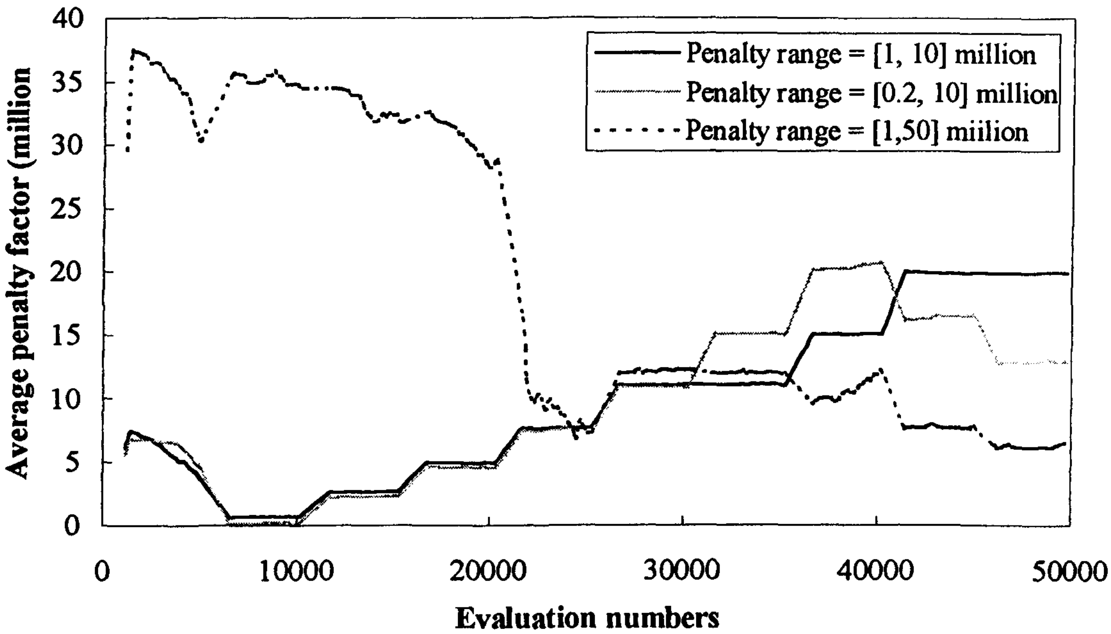
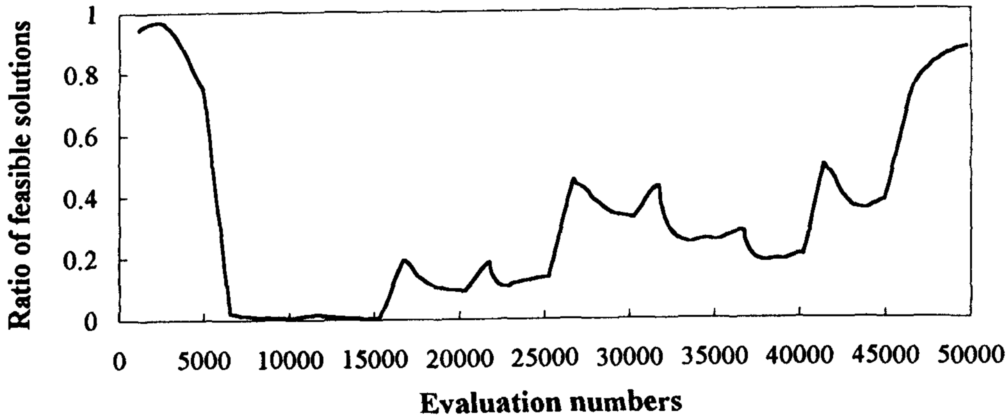
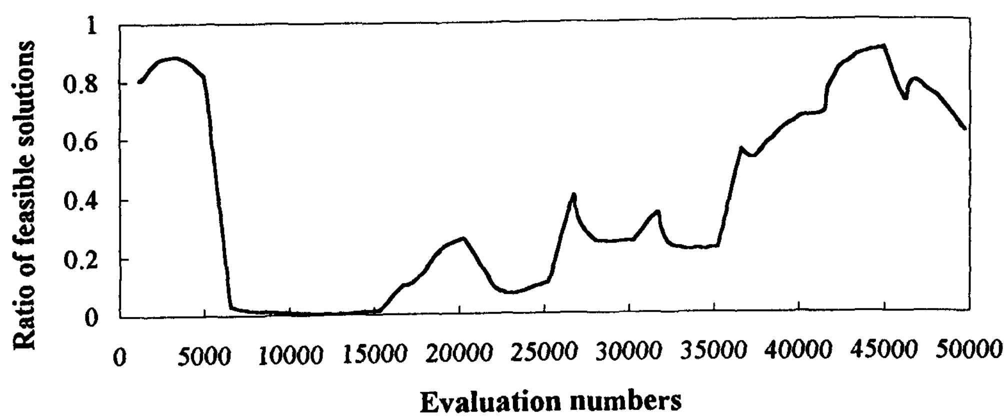


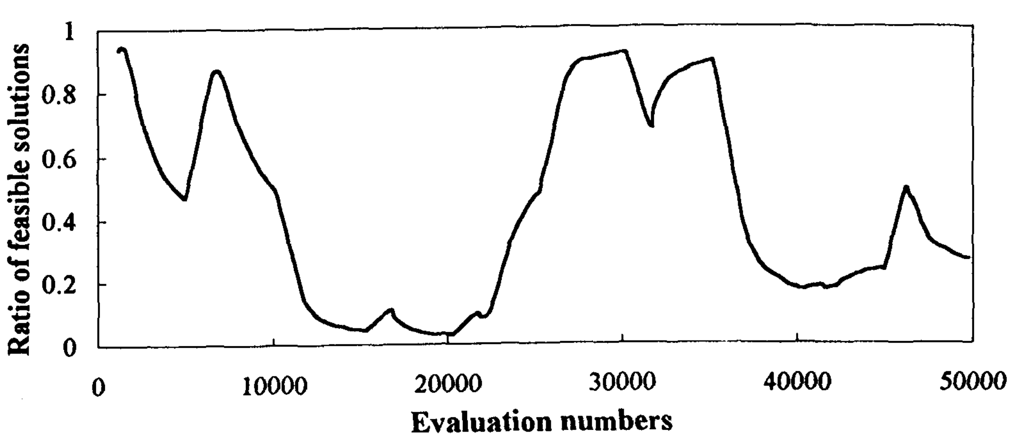
Figure 7-5 Average Penalty of Boundary GA Optimisation of New York Tunnels Problem



(a) penalty factor range = [1, 10] million



(b) penalty factor range = [0.2, 10] million



(c) penalty factor range = [1, 50] million

Figure 7-6 Population Trace of Boundary GA Optimisation of New York Tunnels Problem

7.7 Summary

A method of co-evolution of the penalty factor and a heuristic rule of self-adapting the lower bound and upper bound of the penalty factor has been developed for the optimisation of water distribution systems in this Chapter. The optimal solution of design and rehabilitation of water distribution networks has been observed to be achieved at the boundary of the feasible and infeasible regions of the search space. It is a boundary optimisation problem. A heuristic rule is specifically proposed for adapting the penalty factor range in such a way that the genotype population is forced towards the boundary of the feasible and infeasible regions. The penalty factor of each genotype is coded onto its chromosome. The genotypes which encode the penalty factor and produce better offspring survive longer. The preferred penalty factor spreads through the population and co-evolves over generations.

This approach of a co-evolutionary and self-adaptive penalty factor has been incorporated into the fmGANET for optimisation of design and rehabilitation of water distribution system. A well-studied example of optimisation of New York tunnels problem has been chosen to demonstrate the application of this boundary GA search strategy. The results obtained by using the boundary GA search strategy have been carefully analysed and compared with the conventional penalty GA optimisation approach. It has been found that the conventional approach is quite sensitive to the penalty factor. Too large a penalty value may preserve the feasibility of the genotypes and the search normally converges to an internal point of the feasible region, which is not real optimal solution. Too small a penalty value forces the GA to select the alternative outside the feasible region, which eventually leads the search to converge to an infeasible solution. The boundary search strategy has been shown effective and efficient at adapting the feasibility of the GA population within a large range of the penalty factor. It automatically adjusts the penalty range and co-evolve the penalty factor along the GA optimisation.

The boundary search strategy developed in this Chapter is not only generally applicable to the optimisation of water distribution system, but also to the boundary optimisation problem, which is often found in engineering design. The results are encouraging in that the strategy can be further developed and tested in numerical parameter optimisation using any evolutionary algorithm. It may improve the effectiveness and efficiency for the NLP in the cases where the optimal solution is at the boundary. Finally, this approach will be applied to a more complicated problem of optimisation of water distribution system including water hammer loadings. To be able to consider water hammer loadings in the optimisation model, a computer model for the simulation of the water hammer must be developed. This will be described in next Chapter.

8. MODELLING HYDRAULIC TRANSIENTS IN WATER DISTRIBUTION NETWORKS

8.1 An Introduction

Sudden changes or the stoppage of flows in pipeline systems cause a hydraulic transient or water hammer event. Kinetic energy is destroyed during the change and is converted into pressure energy, which is transmitted as a pressure or water hammer wave in the pipeline system. The water hammer wave often generates the maximum pressure in pipeline systems. The maximum pressure governs the design of pipeline networks including the selection of pipe diameters, pipe wall thicknesses and water hammer control devices. In this Chapter, details of a computer simulation model, based on method of characteristics for modelling water hammer in water distribution networks is presented. The simulation model has been applied to a low head irrigation system. A question arises as to which water hammer event will cause the most severe pressure in the network. The water hammer events of closing valves at different locations in the system are considered. The results obtained show that the various water hammer events create different levels of pressure surge in different pipes within the system. An approach for evaluating the impact of the pressure surge is suggested and is used to measure the severity of the water hammer events. This provides engineers with a method for choosing the most critical water hammer event for the comprehensive design of water distribution systems including water hammer.

8.2 Necessity of Comprehensive Transient Analysis

A water hammer event is important in design, maintenance and operation of water distribution systems. It can cause high pressures, excessive noise and negative pressures. The pipe can be damaged in the short term through over-pressures, or, in the long term, through cavitation in the pipe. Thus the pipeline should be designed either with a suitable pipe size (both diameter and wall thickness) or with an appropriate water hammer control measure to withstand the possible maximum positive pressure and/or the minimum negative pressures. Computer modelling of water hammer in pipeline systems provides a tool for simulation of water hammer events and thus serves a way to provide a better understanding of the behaviour of the transmission of the hydraulic transient pressure waves.

Although the behaviour of hydraulic transient flows have been well understood in simple pipeline systems, little is known about the behaviour of the transients flow in a complex network system. It has been recognised that long pipelines of large diameter may experience severe transient pressures. For network systems, however, the necessity of including water hammer loading into a procedure for design of water distribution system has been controversial. There is a feeling that the network (loop) is more robust than a series pipeline. In other words, it has been postulated (Watters 1984) that the network may behave like a reservoir, which splits the water hammer wave into several subwaves and which in turn diminishes the pressure rise. This assumption has been found little rational basis, and recent research results (Karney and McInnis 1990; 1992; McInnis and Karney 1995) suggested that the opposite might be true.

The traditional wisdom of transient phenomena assumes that the maximum steady-state velocity produces the maximum transient pressure and that surge-protection devices will result in small surge pressures. It has been shown (Karney and McInnis 1990; 1992) that the wisdom can be true only if there is no head loss resulting from friction and no wave reflection. For a complicated water distribution network, fluid transients are not influenced only by the fundamental physical characteristics of the system; their behaviour depends on system configuration, timing of events and initial conditions. In complex water distribution systems, it is unusual to have frictionless flow and an absence of wave reflections. Loops may not diminish the transient response. More protection devices such as pressure relief valves may not guarantee a better transient protection. It is essential, therefore, that computerised tools are developed to enable the investigation of transients flow in complex water distribution networks. A comprehensive analysis of transient conditions must be carried out to identify the worst transient loadings for the water distribution systems. Only then can the diameters and the pipe wall thicknesses (classes) of the pipes be selected and the surge-protection devices be logically sized.

8.3 Governing Equations

The governing equations for unsteady flow in pipeline are derived under the following assumptions including (1) one-dimensional flow i.e. velocity and pressure are assumed constant across a cross-section; (2) the pipe is full and remains full during the transient; (3) no column separation occurs during the transient; (4) the pipe wall and fluid behave linearly elastically and (5) unsteady friction loss is approximated by steady state losses.

The unsteady flow inside the pipeline is described in terms of the simplified unsteady mass balance (continuity) equation and unsteady momentum equation, which define the state variables of Q (discharge) or V (velocity) and H (pressure head), given as (Wylie and Streeter 1993):

$$\frac{\partial H}{\partial t}+\frac{a^2}{g}\frac{\partial V}{\partial x}=0 \tag{8.1}$$

$$g\frac{\partial H}{\partial x}+\frac{\partial V}{\partial t}+\frac{fV|V|}{2D}=0 \tag{8.2}$$

where x = distance along the pipe; t = time; V = velocity; H = hydraulic gradeline; g = acceleration due to gravity; a = wave speed; f = Darcy-Weisbach friction factor and D = pipe diameter.

Eq. (8.1) is the continuity equation and takes into account the compressibility of the water and the flexibility of the pipe material. Eq. (8.2) is the equation of motion. The wave speed a is defined as:

$$a^2=\frac{K/\rho}{1+\left[\left(K/E\right)\left(D/e\right)c_1\right]} \tag{8.3}$$

$$c_1=\begin{cases}1 & \textit{for expansion joints} \quad \textit{when} \;\; D/e>25\\ 1.4(2e/D)+D/(D+e) & \textit{when} \;\; D/e<25\end{cases} \tag{8.4}$$

where K = bulk modulus of elasticity of the flow; e = pipe wall thickness, E = modulus of elasticity and c_l = pipe restraint condition factor.

A method of characteristics transformation is applied to the basic Eq. (8.1) and (8.2). The ordinary differential equation that is obtained is as follows:

$$\frac{g}{a}\frac{dH}{dt}+\frac{dV}{dt}+\frac{fV|V|}{2D}=0 \tag{8.5}$$

This compatibility equation is only valid along the C^+ characteristic equation of :

$$\frac{dx}{dt}=+a \tag{8.6}$$

The other ordinary differential equation obtained is given as:

$$-\frac{g}{a}\frac{dH}{dt}+\frac{dV}{dt}+\frac{fV|V|}{2D}=0 \tag{8.7}$$

This compatibility equation is only valid along the C^- characteristic equation of:

$$\frac{dx}{dt}=-a \tag{8.8}$$

The differential equations (8.5) and (8.7) are solved at the intersection of C^+ and C^- characteristic lines given by equations (8.6) and (8.8).

8.4 Characterisation of Pipeline Systems

The governing equations Eq. (8.1) and (8.2) are seldom solved analytically, a numerical approach is used to approximate the solution. A pipeline system is usually discretised into a number of sections. The compatibility equations Eq. (8.5) and (8.7), derived by applying a method of characteristics, are valid only along the so-called C^+ and C^- characteristic curves, and thus solved at the intersection of these two curves for the interior sections of the pipe. The end of the pipe is regarded as a boundary condition. A set of characteristic equations are derived for various boundary conditions.

8.4.1 Interior section

The x - t grid as shown in Figure 8-1 is used to ensure the characteristic lines given by $\Delta x = \pm a\Delta t$ for solving the Eq. (8.5) and (8.7) for the interior sections of the pipeline. Eq. (8.5) is integrated along the C^+ while Eq. (8.7) is integrated along the C^-. Both integrated equations can be written for the unknowns of H_P and Q_P at P. They are given as:

$$H_p(i,t) = \frac{C_p B_m + C_m B_p}{B_p + B_m} \tag{8.9}$$

$$Q_p(i,t) = \frac{C_p - C_m}{B_p - B_m} \tag{8.10}$$

in which

$$C_p = H_A(i-1) + BQ_A(i-1,t-1) \tag{8.11}$$

$$B_p = B + R\left|Q_A(i-1,t-1)\right| \tag{8.12}$$

$$C_m = H_B(i+1,t-1) - BQ_B(i+1,t-1) \tag{8.13}$$

$$B_m = B + R\left|Q_b(i+1,t-1)\right| \tag{8.14}$$

where B and R are pipe constants given as:

$$B = \frac{a}{gA} \tag{8.15}$$

$$R = \frac{f\Delta x}{2gDA^2} \tag{8.16}$$

where A = cross-section area of the pipe. The information required for the solution at P at time step t is the head and discharge at previous time step t - 1 and the pipe constants. A steady state solution at the first time step is needed to commence the transient simulation.

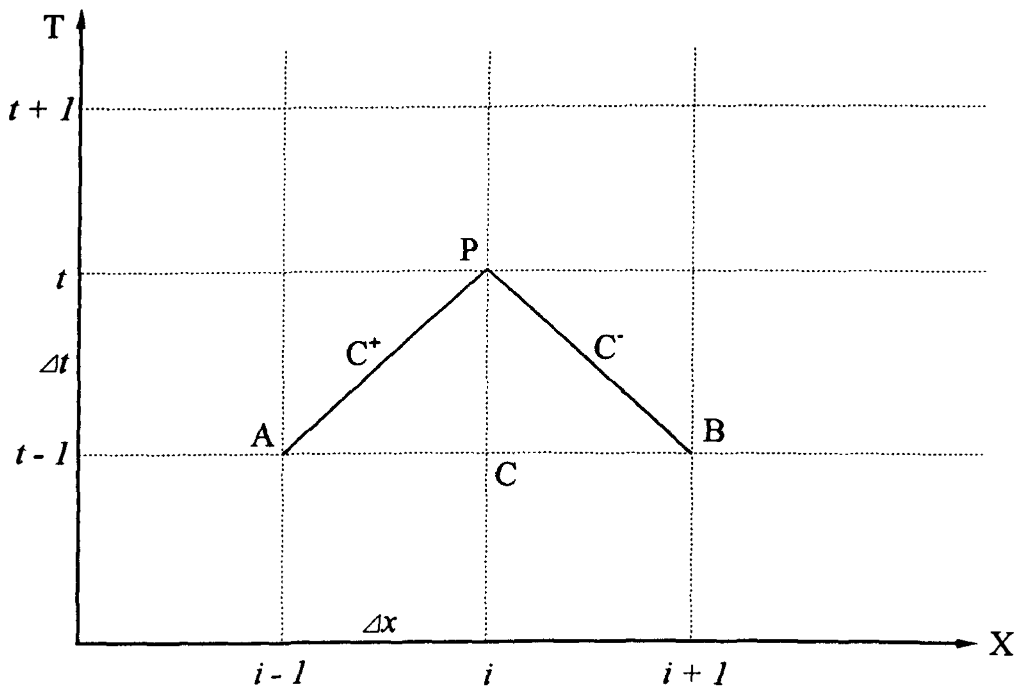


Figure 8-1 Calculation of Q and/or H at Interior Section *i* from Section *i* + 1 at Time Step *t*

8.4.2 Boundary conditions

The method of characteristics provides a systematic formulation for calculating the transient flow within a pipeline. However, if this approach is to applied to the whole range of hydraulic devices at the end of a pipe, an extra relation between head and discharge must be specified. The relationship of the head and discharge is the so-called boundary condition.

The boundary conditions considered in this research are one-node conditions including reservoir, generalised junction, a valve discharging to the atmosphere and an open surge tank, and also the boundary condition between two nodes namely an inline valve. Comprehensive relations of head and discharge have been first derived for one-node boundary conditions by Karney and McInnis (1990) and then extended by McInnis (1992). This approach simplified the control logic of modelling implementation and improved the computation efficiency. Thus it has been adopted in this research. The boundary condition of an inline valve has 3 unknown variables that include head on the upstream side of the valve ; head on the downstream side of the valve and flow through the valve. They can be explicitly solved by a set of 3 equations including the orifice flow equation, one characteristic equation on the upstream side of the valve and the characteristic equation on the downstream side of the valve. The boundary conditions for solving hydraulic devices such as inline valves, outlet valves and surge tanks, and the characteristic equations for solving interior points have been implemented as described in the following section.

8.5 Modelling System Implementation

A transient model (Simpson et al. 1992), originally developed for water hammer simulation of hydro-electric power plant systems, was based on the method of characteristics. This method was used to solve the unsteady continuity equation and unsteady equation of motion governing flow and pressure head variation in the networks. For flow elements where non-linear equations are added to the method of characteristics equations—a set of nonlinear equations results. A Newton-Raphson iterative solution procedure is used to solve for the unknown variables for each flow element and/or boundary condition.

The transient solution, based on the method of characteristics, proceeds point by point. Each pipe in the network is divided into an even number of reaches. Head and discharge conditions are solved at the end of each reach (called sections). These sections constitute interior points. Each interior point in a pipe is solved at every second time step for the head and flow independently of other points in the system. A non-pipe element such as a junction, reservoir, valve or surge tank is referred to as a particular boundary condition and is solved simultaneously for all the variables associated with that boundary condition.

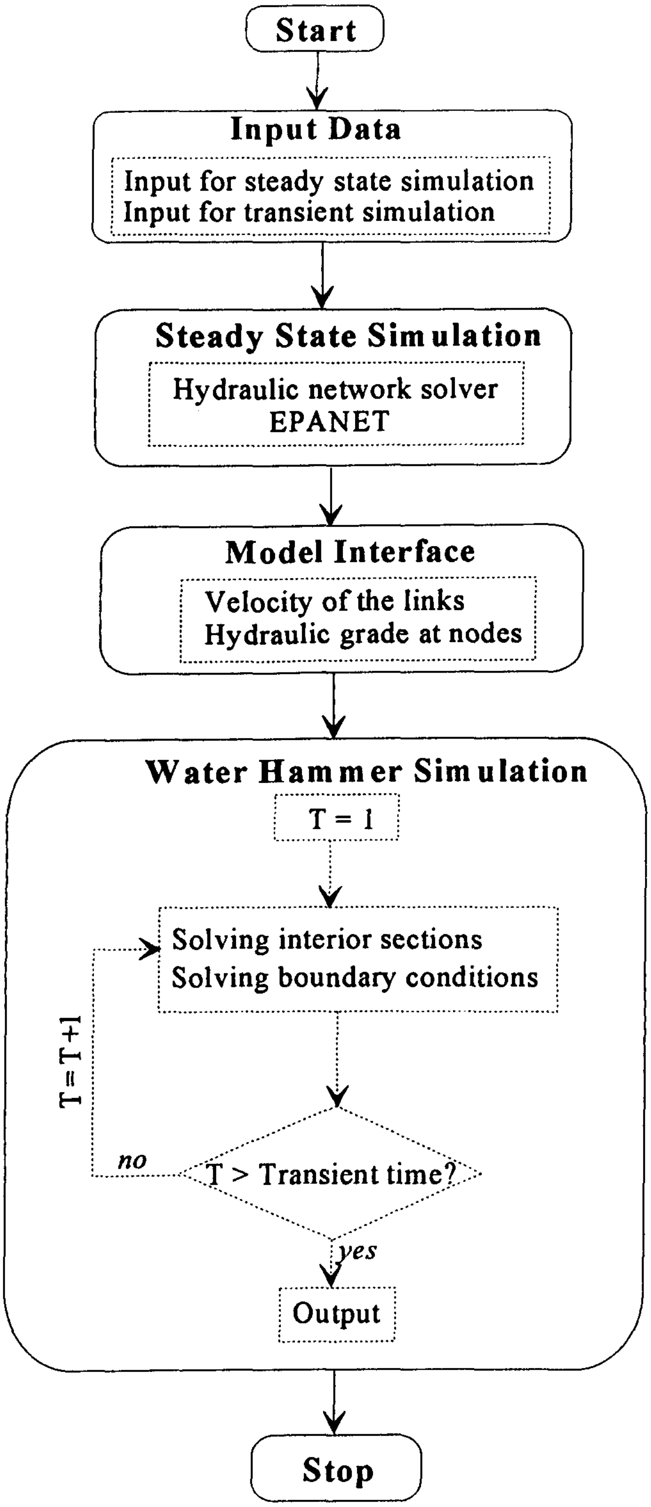


Figure 8-2 Model Structure of Integrated Hydraulic Transient Simulation Program EPANET/HAMMER

A transient model, developed for hydro-electric power systems by Simpson et al. (1992), has been improved for the transient simulation of water distribution systems in this research. The model has been enhanced to be more robust at handling surge tanks, different pipe materials, wave speeds, outlet valves and valve operations. The enhanced transient model is integrated with a hydraulic network solver EPANET (Rossman 1994), which is employed for simulating steady state flow in the network to establish the initial flow conditions for the transient simulation. The integrated computer model is then able to simulate the hydraulic transient of the water distribution networks with reservoirs, nodes and/or junctions, in-line valves, outlet valves and surge tanks.

The integrated transient modelling system, named HAMMER, follows a program structure as shown in Figure 8-2. The program starts with reading input data for steady state simulation and transient simulation. The input data for steady state model has been kept in the same format as original input file of EPANET computer simulation model. The input for the transient simulation includes the time period of transient simulation, pipe material type, surge device data, pipe thickness and valve operation for generating water hammer events. The hydraulic network solver EPANET is called for the simulation of the steady state flow conditions. The simulation results of velocity of each link and hydraulic grade of each node are passed through a model interface program to the transient model. This establishes an initial flow condition for the water hammer simulation. The transient model follows an iterative procedure until the transient simulation time is reached. Interior sections are alternatively solved at every time step while the boundary conditions are solved at every other time step. The integrated EPANET/HAMMER model provides a hydraulic model base for optimisation of water distribution system including water hammer. The modelling system has been used in this research to simulate water hammer event in a low head irrigation system. The results have been verified by using a widely-distributed commercial program LIQT in the following sections.

8.6 A Low Head Irrigation System

Irrigation areas are usually supplied by reservoirs (or storage tanks) at the upstream end of an irrigation distribution network. In a low pressure system, the minimum allowable pressure head is usually in the order of 2 to 5 meters. Velocities depend on the particular location of a pipe in the network and the demand distribution in the network. A sudden change of the water demand by an irrigator, for instance by closing the valve, may cause a severe water hammer pressure. Thus the system must be designed in such a way that the pipeline can sustain the maximum water hammer pressure. The computer model described in Section 8.5 has been applied to simulating the transient behaviour of a low head irrigation system as depicted in Figure 8-3. A comprehensive investigation of different water hammer events has been carried out to identify the critical water hammer loading for the system design.

8.6.1 System description

An existing irrigation system, called Loveday irrigation network as shown in Figure 8-3, has been chosen as a case study in this Chapter. The Loveday irrigation area is adjacent to the River Murray in the Riverland region of South Australia. It was managed by the South Australia Water Corporation until 1 July 1997 when the Central Irrigation Trust was formed. Most of the system was constructed in the 1920's. An extensive rehabilitation of the existing aged pressurised system is presently being undertaken. The network data and wave speeds used for the transient analysis are given in Appendix B.

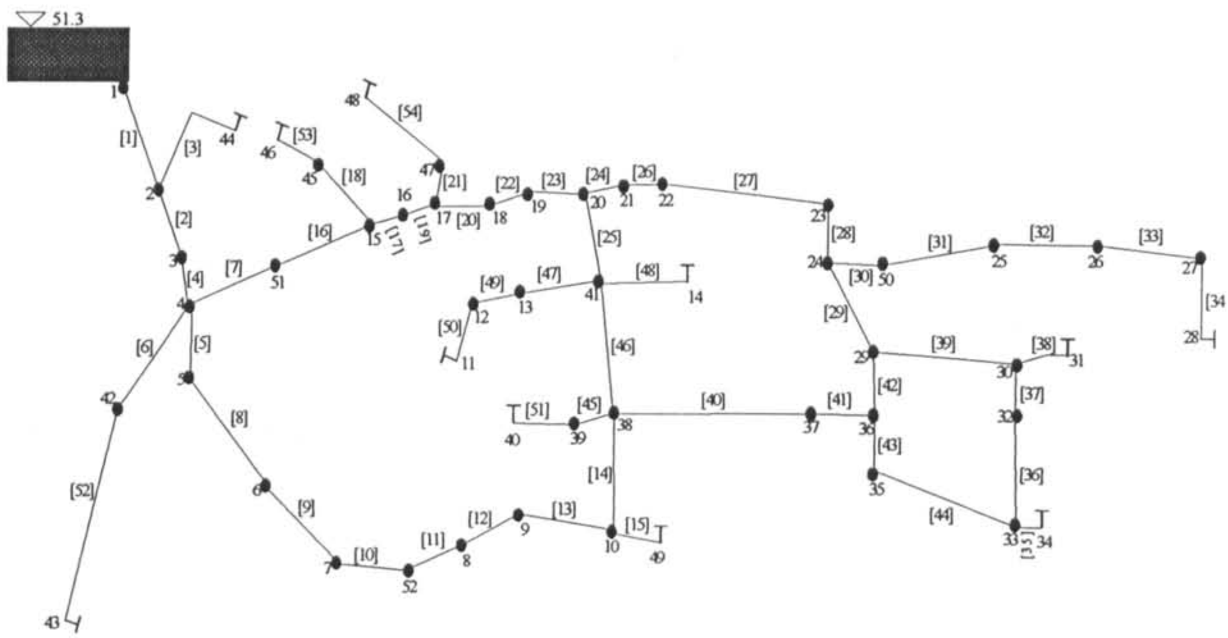


Figure 8-3 Layout of Loveday Irrigation Network

8.6.2 Transient events and boundary conditions

The possible water hammer events for this case study are sudden change of water demand i.e. closing the valve at the outlet from the pipeline. The boundary conditions associated with the simulation of the water hammer events in the Loveday irrigation system are the reservoir and the outlet valves in the network.

There are 11 operating valves considered in this study. The valves are at nodes 11, 14, 28, 31, 34, 40, 43, 44, 46, 49, and 48 as shown in Figure 8-3. The transient that results by operating each of the valves is independently simulated. In other words, for the simulation of the transient behaviour generated by operating each of the 11 valves separately, with only one valve being assumed to be operating (with all other outlets closed) and water hammer is created by instantaneous closure of that valve. A steady state simulation is performed for each fully opened valve to establish the initial flow condition for the water hammer simulation, and then the transient simulation is carried out for the instantaneous closure of the valve.

8.6.3 Transient evaluation

In proceeding with the analysis of the different water hammer events, a question arises as to how to measure the severity of a specific transient event. The severity of the transient event can be evaluated not only in terms of the magnitude of the maximum water hammer pressure generated by the event, but also the region of the network affected by the event. Thus a measure for evaluating the severity of the transient event is introduced as the total length of the pipes where the maximum associated pressure is greater than a certain pressure threshold. The affected length is defined as:

$$L_{affected} = \sum_{i=1}^{N} L_i \qquad when \quad H_i^{\max} > H_{threshold} \tag{8.17}$$

where $L_{affected}$ = total length of pipes affected by water hammer events; L_i = length of pipe i; $H_i^{\max}$ = the maximum transient pressure of pipe i; $H_{threshold}$ = transient pressure threshold and N = total number of pipes.

8.6.4 Simulation results

A comprehensive investigation of the transient events caused by operating one valve in every transient run has been carried out for Loveday irrigation network. The transient events that have been simulated include (1) instantaneous closure of valve 28 in 2 transient runs for concrete and Hobas pipes; (2) instantaneous closure of 10 valves in 10 transient runs for concrete pipes and (3) closure of 11 valves in 10 seconds and 5 seconds in 22 transient runs for concrete pipes. The transient simulations of 34 transient runs provide a comparison of the transient behaviour for different pipe materials and different events, and an evaluation of the transient events and also the envelope of transient pressure heads for Loveday irrigation system.

(a) Different pipe materials

Two different pipe materials have been considered including reinforced concrete pipes and Hobas pipes. The modulus of elasticity (E) of these two materials is quite different, and thus contributes to different wave speeds for water hammer events. For the reinforced concrete, Young's modulus is $E = 45\text{x}10^9$ Pa while for Hobas $E = 10\text{x}10^9$ Pa. Figure 8-4 shows that reinforced concrete and Hobas pipes have a similar transient behaviour for the same water hammer event of instantaneously closing valve 28, but a different magnitude of transient pressure occurs due to the different modulus of elasticity of pipe materials. The transient pressures in a concrete pipe system are higher than in a Hobas pipe system. This implies that a severe hydraulic transient may occur not only in a simple pipeline, but also in complicated networks of low elasticity pipes namely high Young's modulus pipe material such as reinforced concrete.

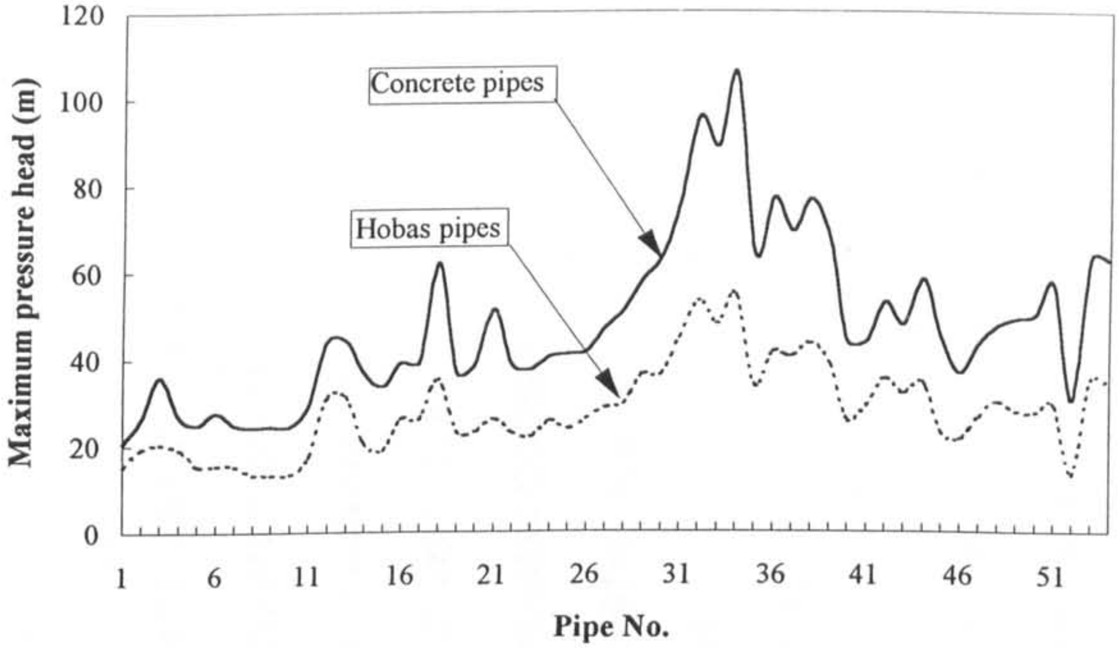


Figure 8-4 Comparison of Maximum Pressure by Instantaneous Closure of Valve 28 for Both Concrete and Hobas Pipes

(b) Evaluation of transient events

The transient events are evaluated by the measure introduced above with pressure head threshold of 40 meters. As shown in Figure 8-5, the operation of valve 28 generated the most severe transient pressure for both reinforced concrete and Hobas pipes. The transient in the network of reinforced concrete pipes is much more severe than the Hobas pipes for all the water hammer events. Although the total length of the pipes affected by the water hammer event caused by an instantaneous closure of valve 28, 31, 34, 11, 14, 40 is very much same, the locations of pipes affected by the valve closure are different. As shown in Figure 8-6, 8-7 and 8-8 (for reinforced concrete pipes), the maximum transient pressure normally occurs just upstream of the valve which has been closed. Thus not a single water hammer event (loading) can represent a critical transient loading for the design of the network.

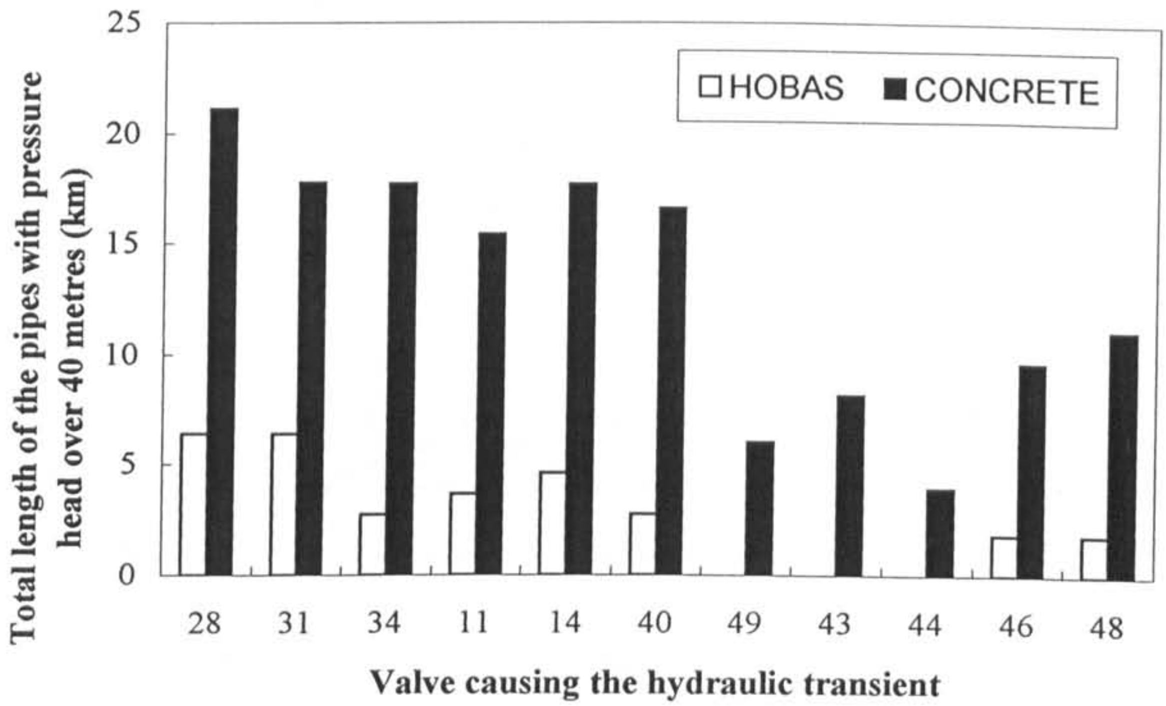


Figure 8-5 Evaluation of Different Transient Events

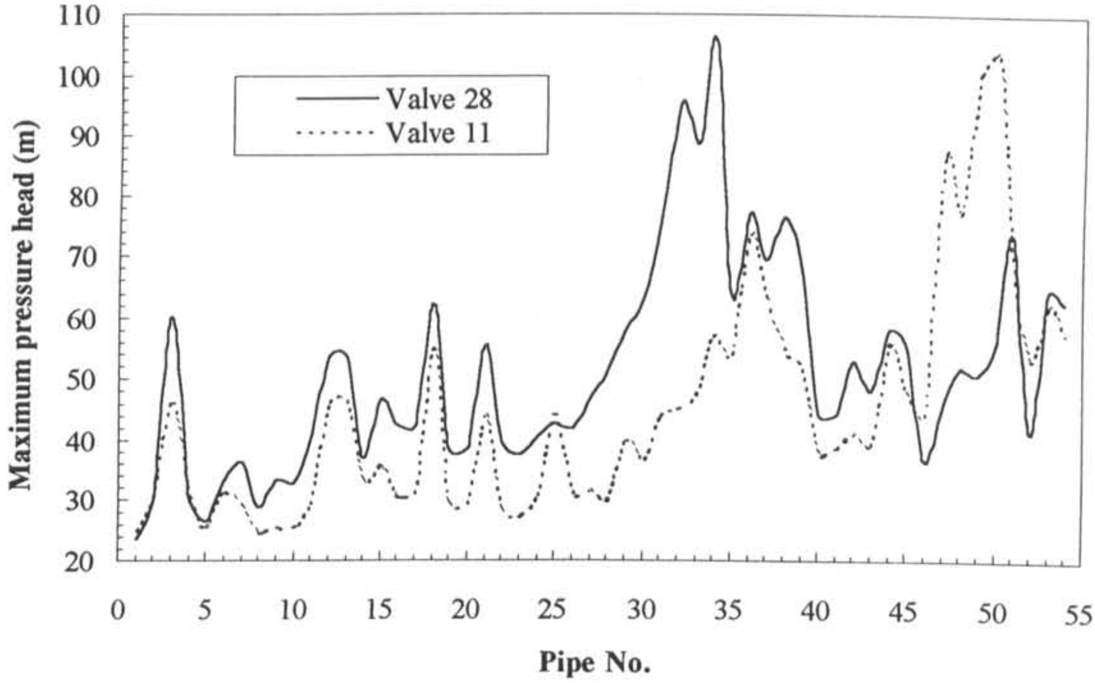


Figure 8-6 Maximum Pressure Heads of Loveday Network by Operating Valve 11 and Valve 28 for Concrete Pipes

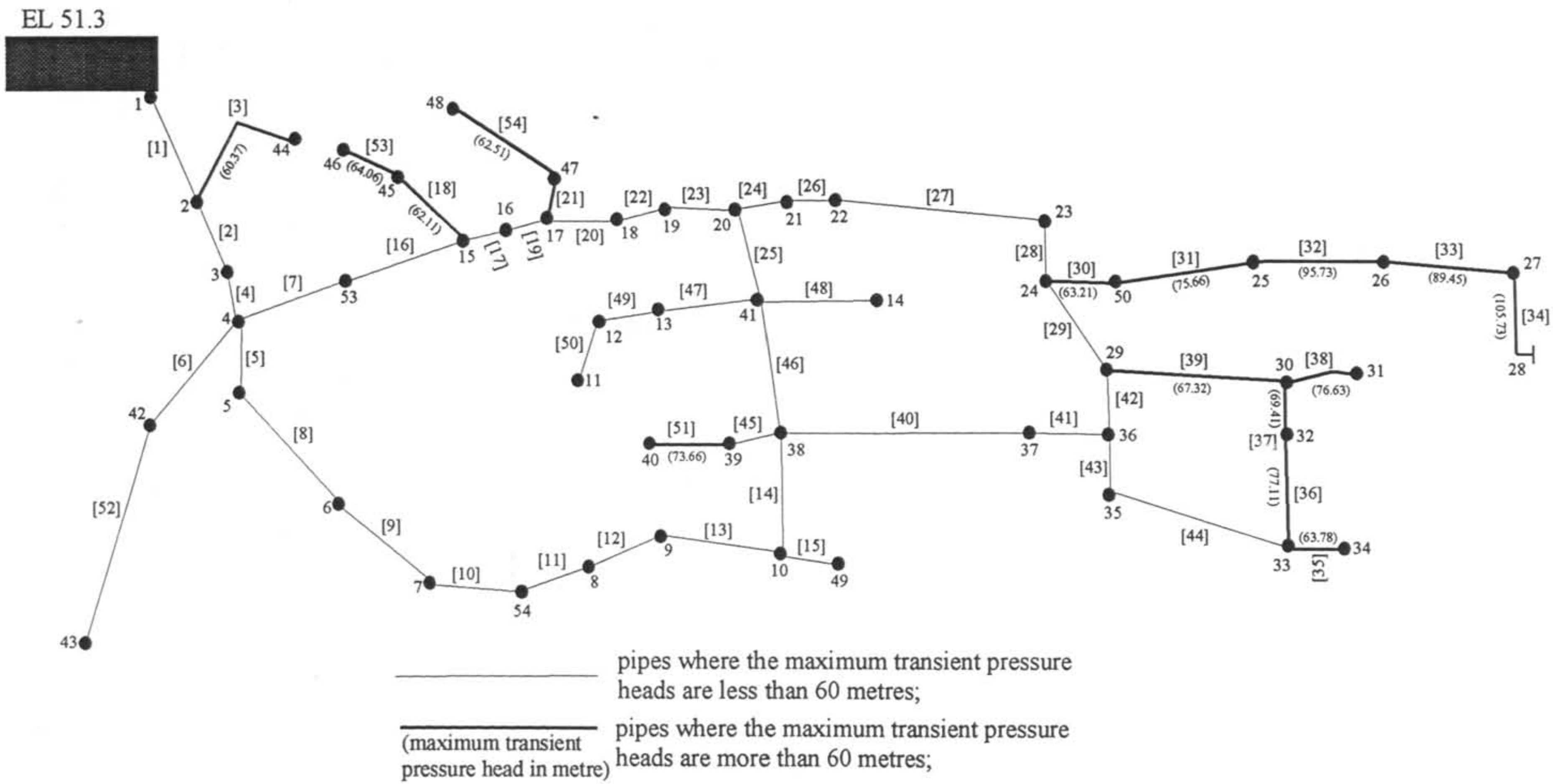


Figure 8-7 Locations of the Maximum Transient Pressure Heads of Loveday Network by Operating Valve 28 for Concrete Pipes

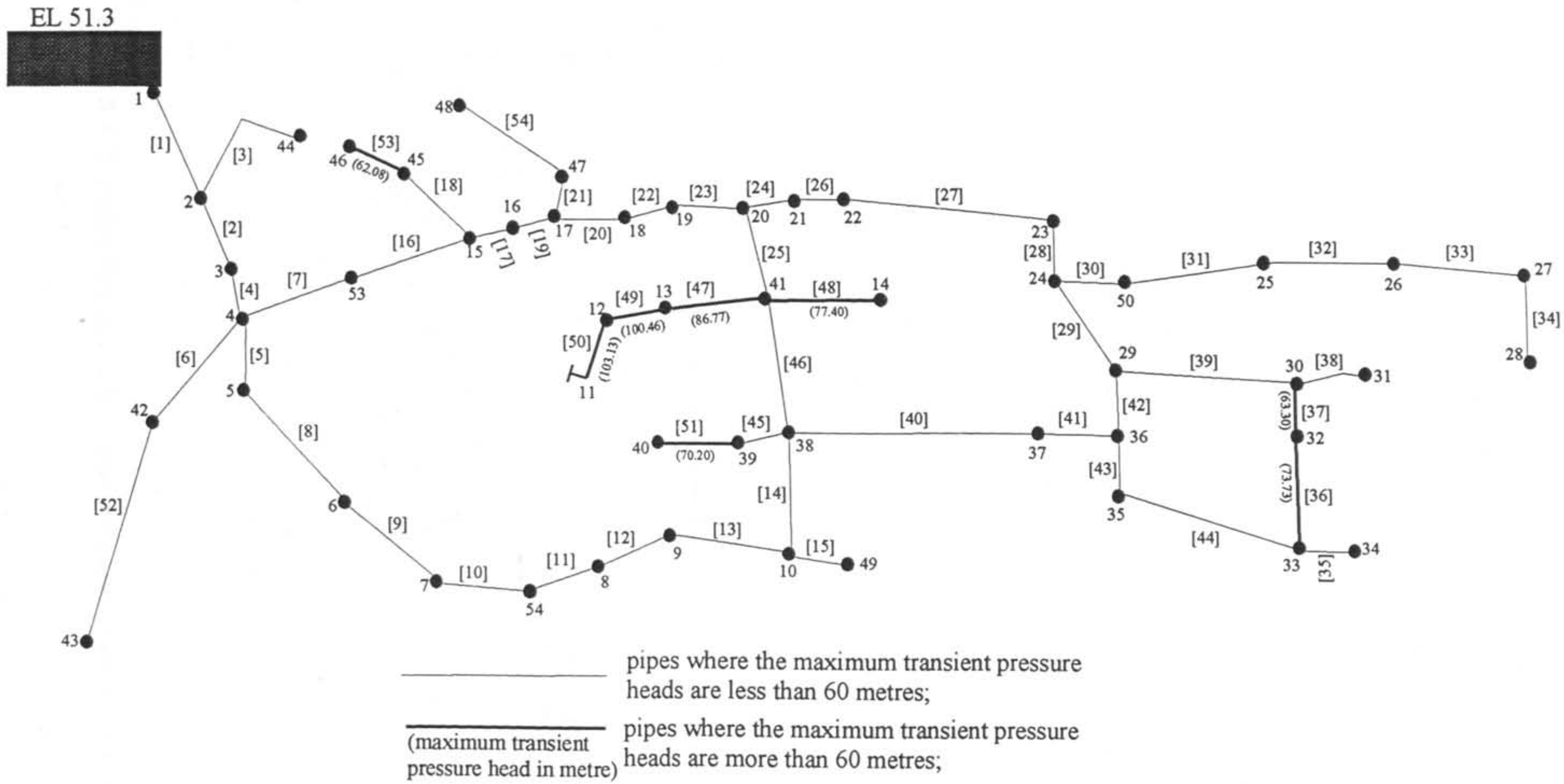


Figure 8-8 Location of the Maximum Transient Pressure Heads of Loveday Network by Operating Valve 11 for Concrete Pipes

(c) Transient pressure envelope

The maximum and minimum transient pressure heads usually govern the transient design of water distribution networks. Different pipes in the network have different responses to the different water hammer events. The pipe maximum and minimum pressure heads of Loveday irrigation network (concrete pipes) are considered the highest and lowest pressure heads that occur among all the computation points of the pipe over the time of all the possible water hammer events. This provides an envelope of the maximum and minimum transient pressures for the Loveday system, as shown in Figure 8-9. It has been observed that the gap between the maximum and minimum transient pressures is quite large for most of the pipes. Thus a comprehensive analysis of hydraulic transients must be carried out to identify the critical water hammer loadings for the cost-effective selection of the water hammer control measures for the water distribution networks.

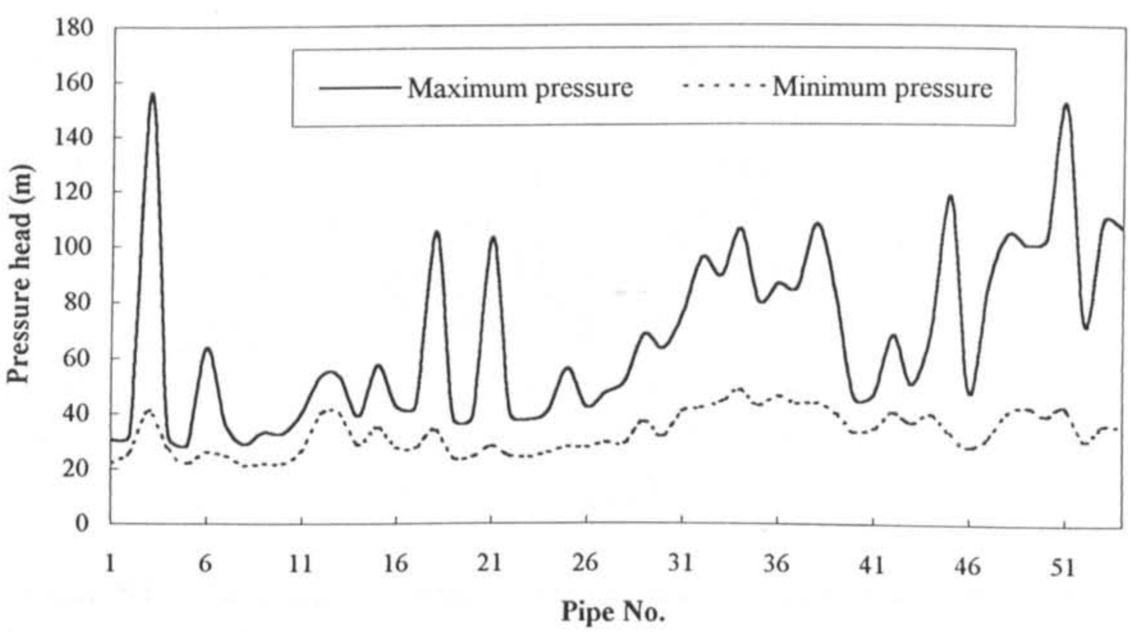


Figure 8-9 Pressure Envelope of Loveday Irrigation System by Operating 11 Valves

(d) Different valve operating times

The transient events of closing the outlet valves in different time periods generate significantly different transient pressures as shown in Figure 8-10. The transient simulations of 22 runs for closing 11 valves in 10 seconds and 5 seconds have been carried out for Loveday irrigation system of concrete material. The maximum transient pressure head of each pipe over the transient events of closing 11 valves in 5 seconds and 10 seconds is compared with the maximum transient pressure heads generated by instantaneous closure of the valves. Figure 8-10 shows that the instantaneous valve closure generates the most severe transient pressure in all the pipes and that the transient events of valve closure in 5 seconds and 10 seconds significantly reduced the maximum transient pressure heads. Thus a pipeline network must be designed to sustain the most severe transient event.

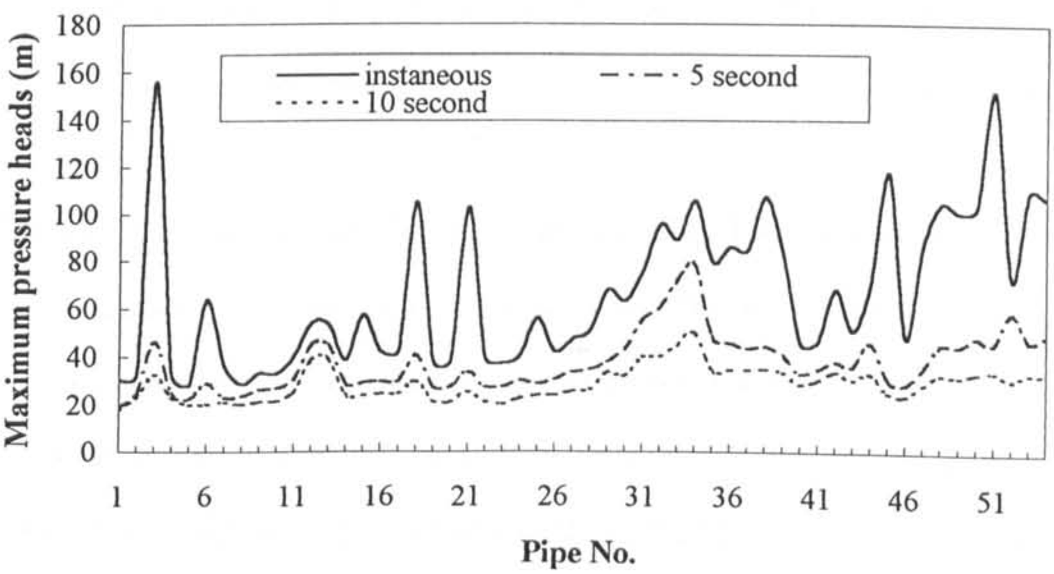


Figure 8-10 Comparison of Maximum Pressure Heads of Loveday Network by Operating 11 Valves in Different Periods of Time

8.7 Verification of the Transient Model

The hydraulic transient model established above for Loveday network has been verified by using LIQT, a commercial computer program developed by Stoner Associates for transient analysis in pipeline systems. The verification has been carried out for Loveday irrigation system of concrete pipes with and without surge tank at node 23. The comparison of the simulation results by HAMMER and LIQT is described in this section.

8.7.1 Verification without a surge tank

The transient event used for the verification of the water hammer model of Loveday network without a surge tank is initiated by the instantaneous closure of valve 28. The simulation results by HAMMER are compared with those obtained by LIQT. The comparisons include a transient pressure head envelope, namely the maximum and minimum transient pressure heads of the pipes, and the time series of transient hydraulic grade lines at nodes 11 and 28.

The verification of the water hammer simulation of the Loveday network has been carried out by increasing the reservoir level from 51.3 m to 100.0 m. It is observed that the simulation results by HAMMER can not match with that by LIQT over the whole period of transient simulation time but just the first 15 seconds. This is because the function of simulating column separation is not incorporated into HAMMER properly. It is found that raising the reservoir level to 100 m is able to get rid of the column separation in Loveday network. Thus it helps to compare the consistent HAMMER simulation with LIQT.

Figure 8-11 shows the comparison of transient pressure head envelope for the pipes in Loveday network. It has been observed that the maximum and minimum transient pressure heads by both HAMMER and LIQT match very well. The differences between the transient pressure head envelops by HAMMER and LIQT are shown in Figure 8-12. The discrepancy between the two computer program simulations is within a range of ±0.4 metres. The time series of the transient grade at node 11 and 28 are compared in Figure 8-13(a) and Figure 8-14(a). The differences of the transient grades by LIQT and HAMMER are illustrated in Figure 8-13(b) and 8-14(b). They are not more than ±0.6 metres at node 11 and ±0.8 metres at node 28. Thus it is validated that HAMMER is able to produce quite accurate results for transient analysis of water distribution networks without a surge tank.

8.7.2 Verification with a surge tank

The verification of Loveday network with a surge tank at node 23 has been carried out in the same way as the verification without the surge tank. The simulation results compared include the transient pressure head envelope for the pipes, the time series of transient hydraulic grade lines at nodes 20 and 28, and also the water level of the surge tank. As described in Section 8.7.1, the verification has been carried out by raising the reservoir level from 51.3 m to 200.0 m to get rid of the column separation when simulating the water hammer events with the surge tank at node 23.

It has been found that the transient model by HAMMER predicts accurate results for the Loveday network with the surge tank by comparing with the results by LIQT. Figure 8-15 shows that the maximum and minimum transient pressures occur in a small number of pipes due to the surge tank protection of the transient flow. The discrepancy of the maximum and the minimum pressure by LIQT and HAMMER is given in Figure 8-16. It shows that the differences are within a range of ±0.4 metres. The comparison of time series of the transient grade at node 20 and 28 is shown in Figure 8-17 and 8-18. The difference of the time series by HAMMER and LIQT for Loveday network with the surge tank is in the same scale as the verification of the network without the surge tank. The water level of the surge tank by HAMMER is also compared with the level predicted by LIQT in Figure 8-19. Because of the rounding error of LIQT program for preparation of plots, the curve shape of the time series of the tank level by LIQT is different from that by HAMMER. But it can be observed that the amplitude and the phase of the surge levels are very similar and that the difference between the tank levels by both computer programs is not more than ±0.5 metres. These results presented in this Chapter prove that the transient model HAMMER program is able to accurately simulate the behaviour of water hammer events of water distribution systems with and without the surge tanks. Thus the model implemented in this Chapter provides a computational tool for the comprehensive transient analysis and the cost-effective transient design of the water distribution systems.

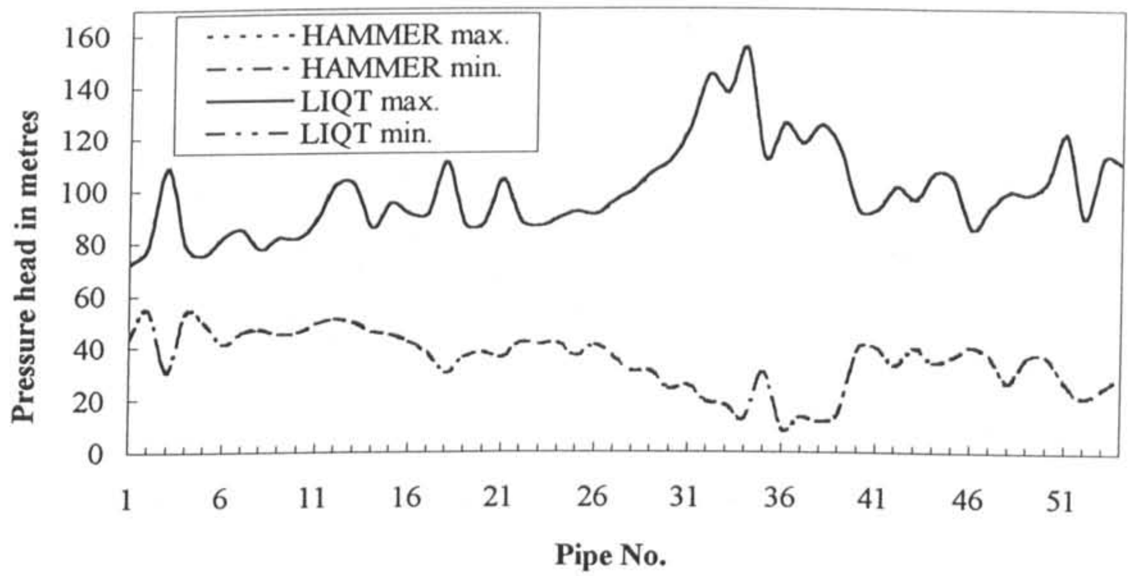


Figure 8-11 Comparison of the Maximum and Minimum Pressure Heads of Loveday Network Without Surge Tank

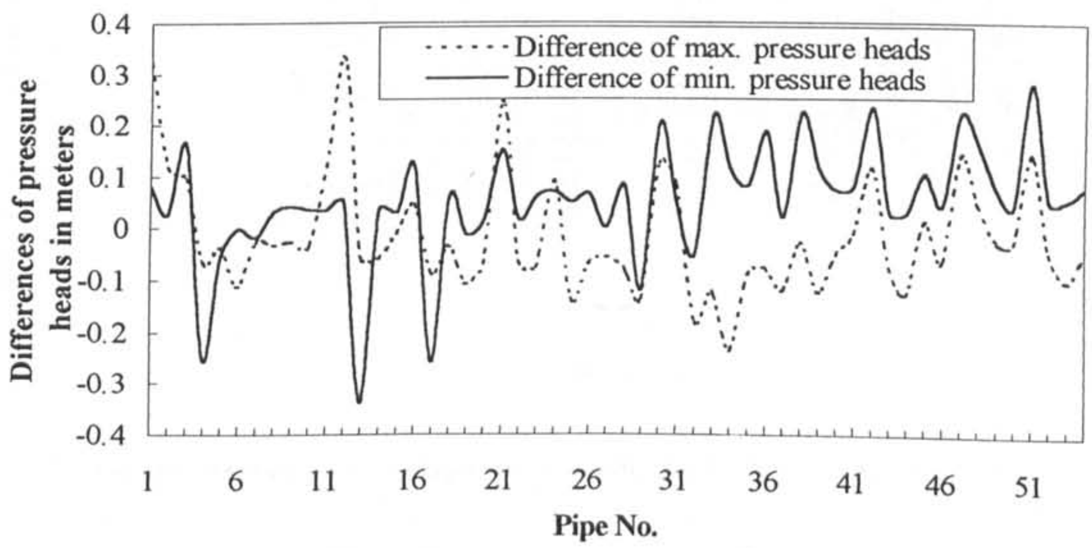
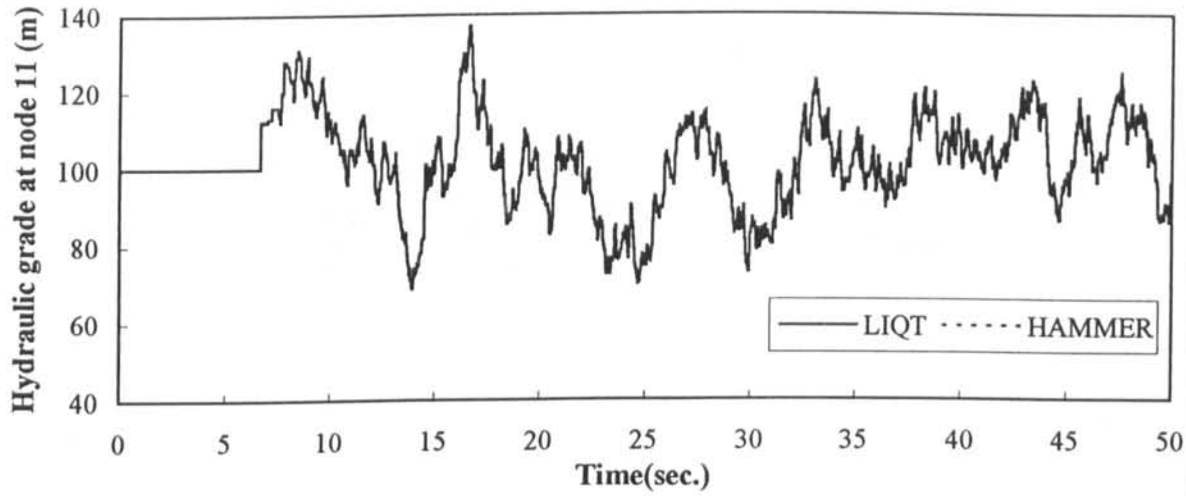
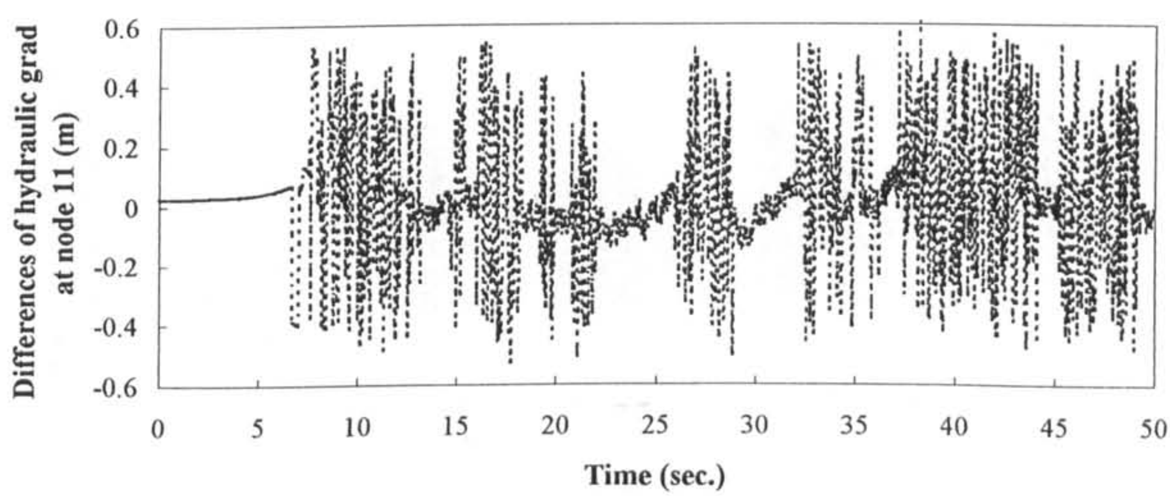


Figure 8-12 Difference of the Maximum and Minimum Pressure Heads Between HAMMER and LIQT for Transient Simulation Of Loveday Network Without Surge Tank

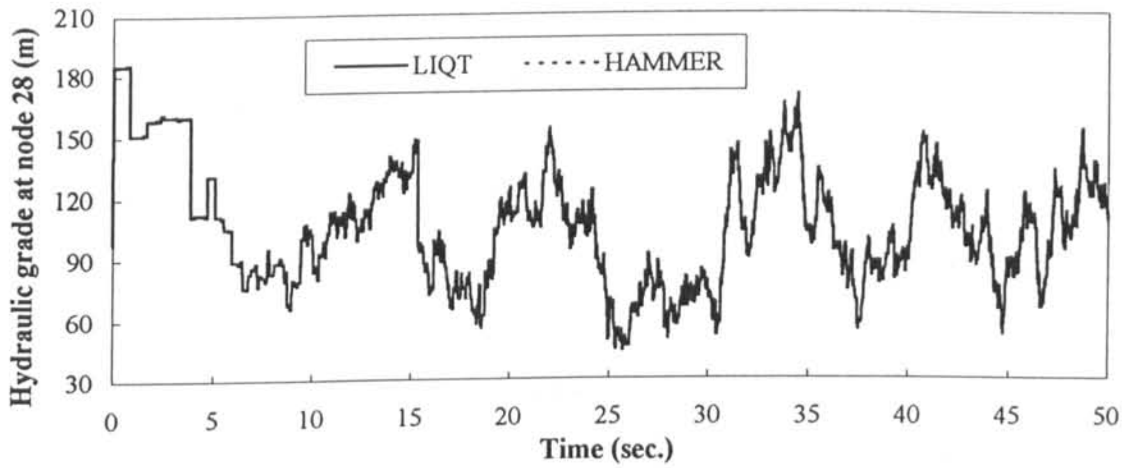


(a) Comparison of time series of hydraulic grade by LIQT and HAMMER

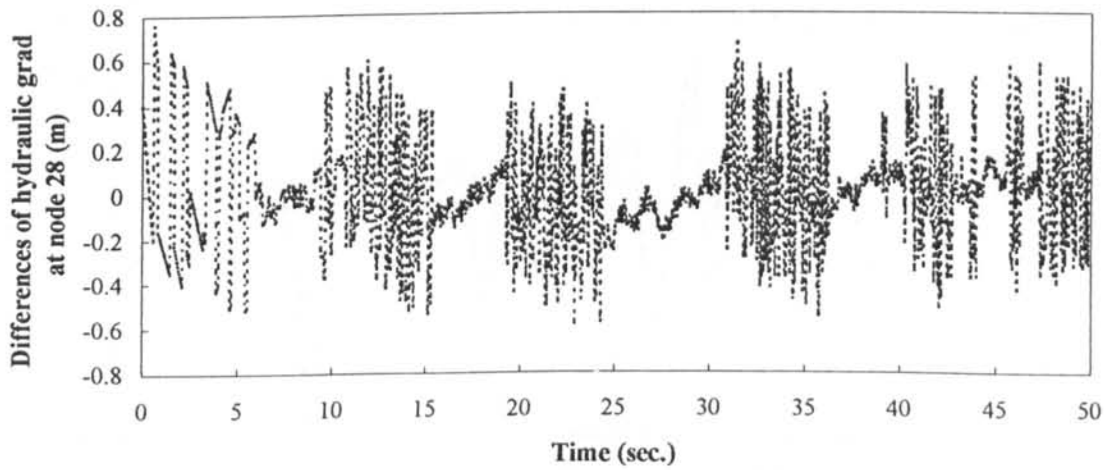


(b) Differences between hydraulic grades by LIQT and HAMMER

Figure 8-13 Comparison of Time Series of Hydraulic Pressure at Node 11 of Loveday Network Without a Surge Tank



(a) Comparison of time series of hydraulic grade by LIQT and HAMMER



(b) Differences of hydraulic grades by LIQT and HAMMER

Figure 8-14 Comparison of Time Series of Hydraulic Pressure at Node 28 of Loveday Network Without a Surge Tank

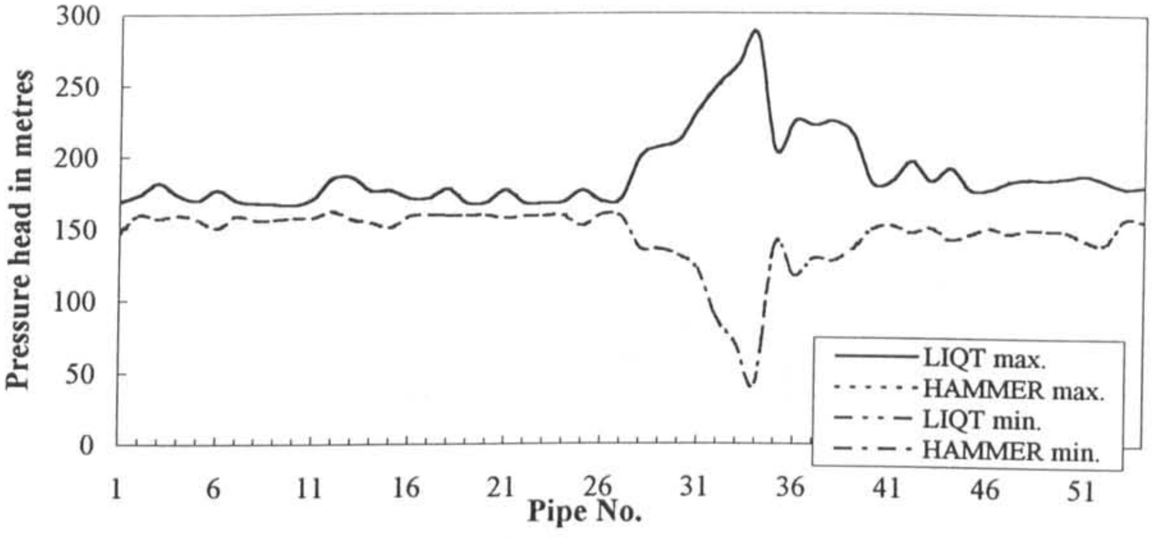


Figure 8-15 Comparison of the Maximum and Minimum Pressure Heads of Loveday Network With a Surge Tank

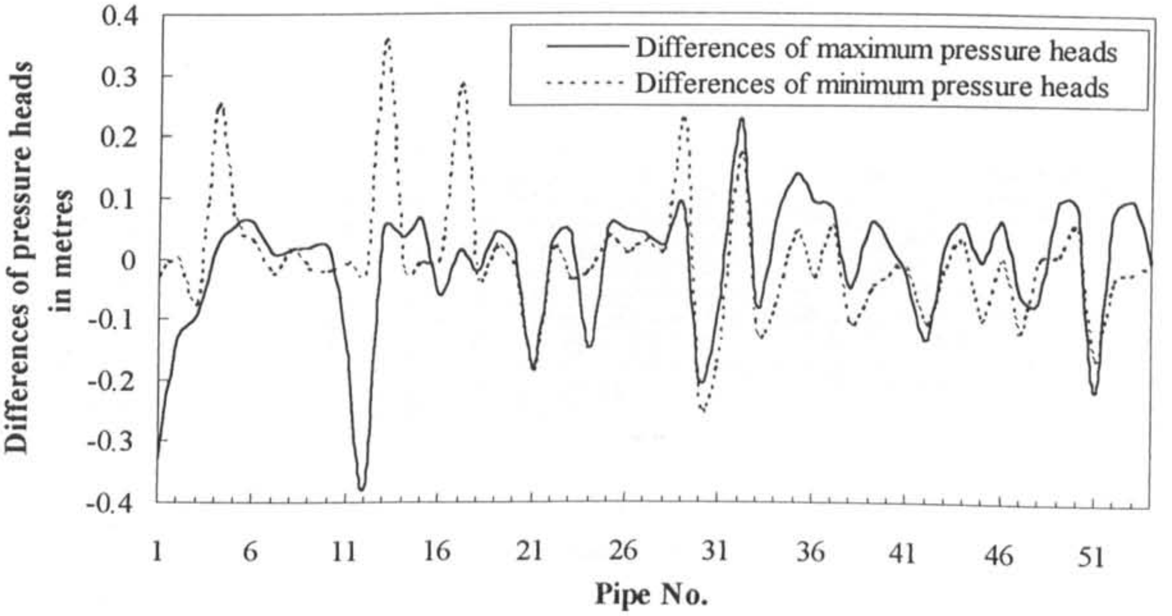


Figure 8-16 Differences of the Maximum and Minimum Pressure Heads Between HAMMER and LIQT for Transient Simulations of Loveday Network with a Surge Tank

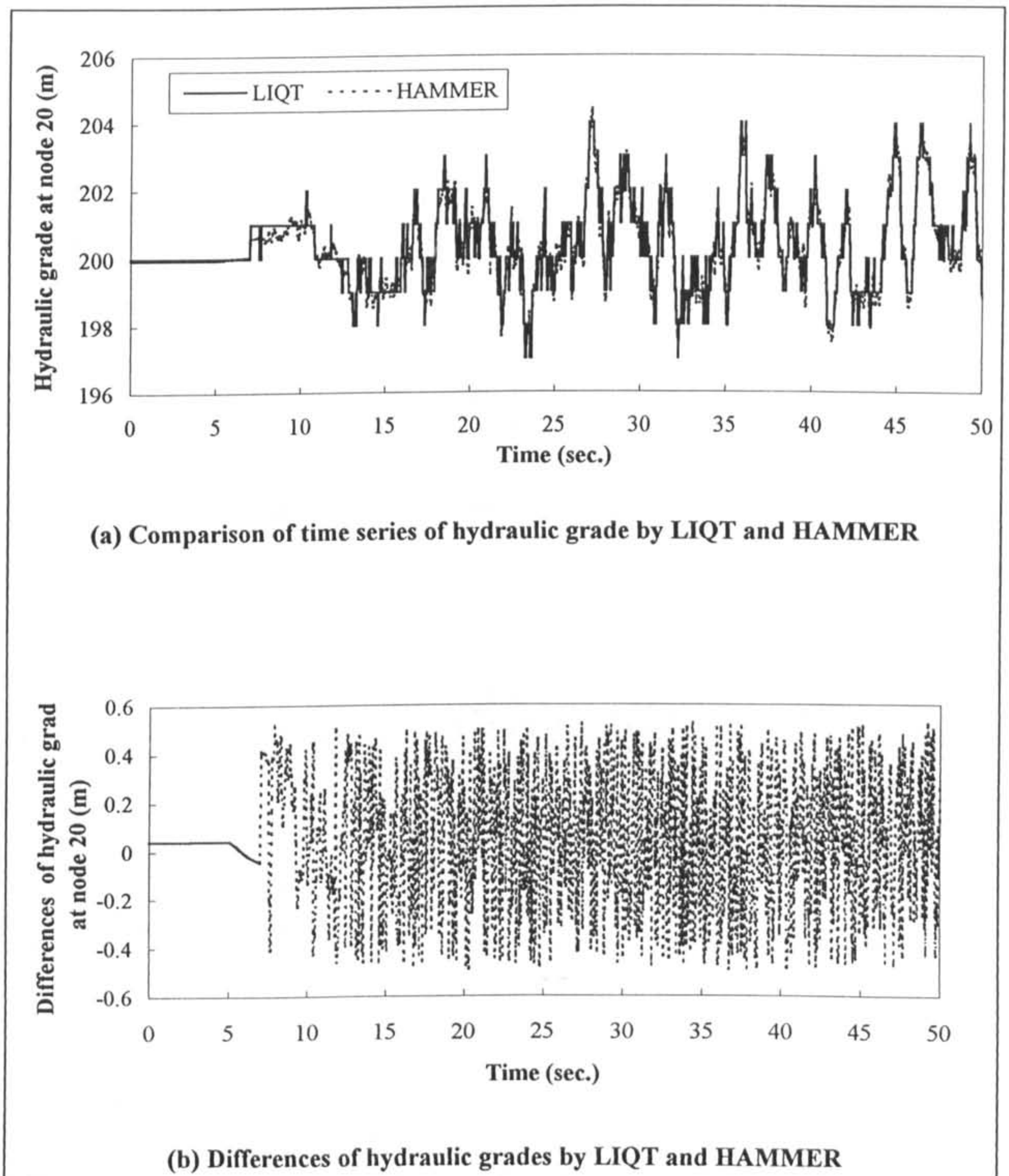
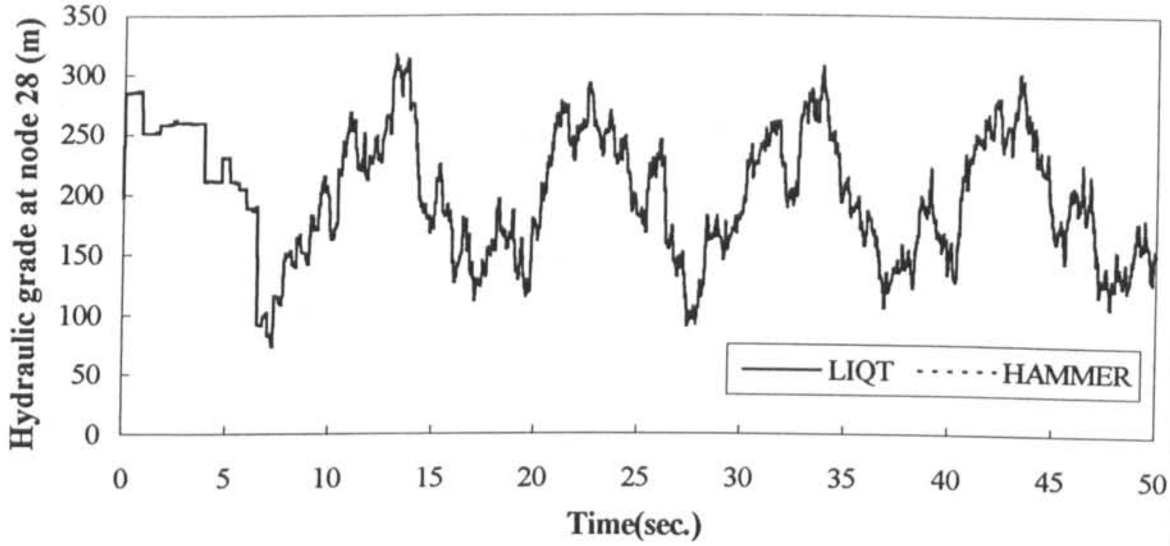


Figure 8-17 Comparison of Time Series of Hydraulic Pressure at Node 20 of Loveday Network with a Surge Tank



(a) Comparison of time series of hydraulic grade by LIQT and HAMMER

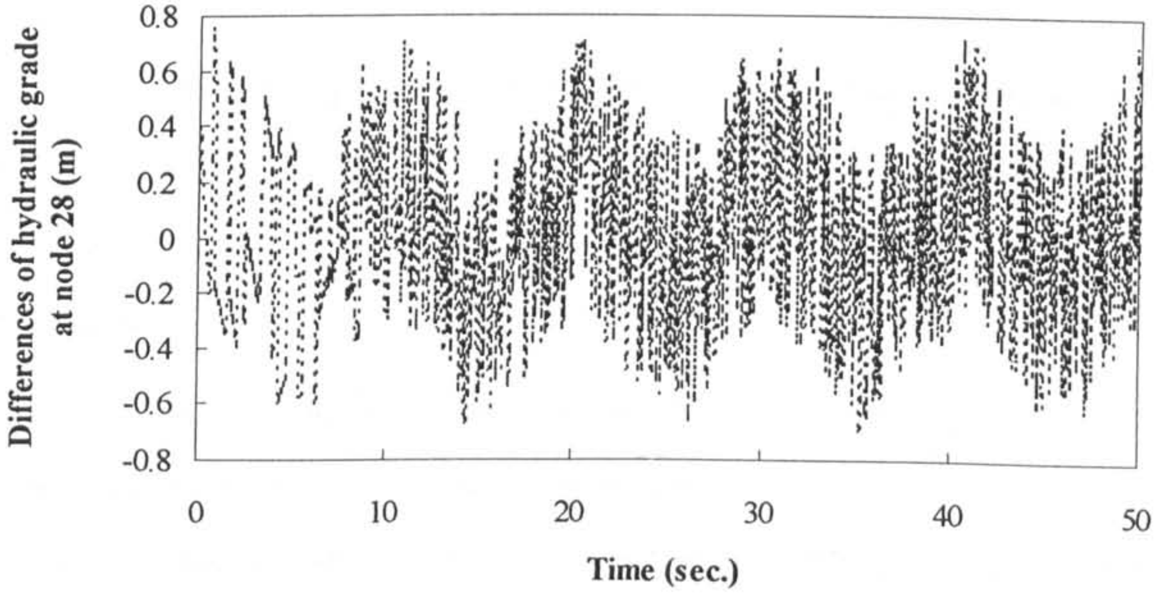


Figure 8-18 Comparison of Time Series of Hydraulic Pressure at Node 28 of Loveday Network with a Surge Tank

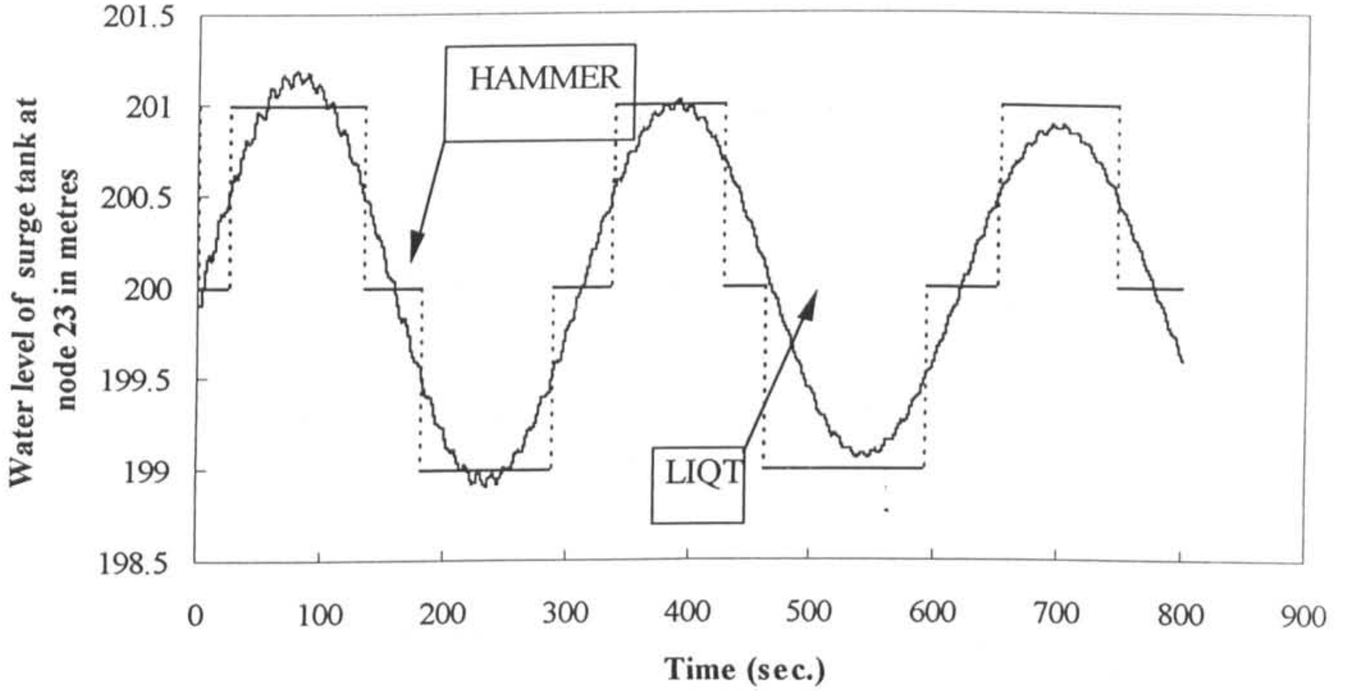


Figure 8-19 Comparison of Water Levels of the Surge Tank at Node 23

8.8 Summary

A computer model for water hammer simulation of water distribution systems has been developed and applied to the investigation of transient behaviour of a low head irrigation system. The transient model, based on the method of characteristics, has been integrated with a hydraulic solver EPANET and is able to perform the simulation of the steady and transient flow in the pipeline network with reservoirs, nodes, junctions, in-line valves, outlet valves and surge tanks. It provides a computational tool for transient analysis of water distribution systems.

A comprehensive analysis of the transient events has been carried out for Loveday irrigation system in South Australia. This example has contributed to a better understanding of the system behaviour under possible water hammer loadings, and thus enables engineers to select the cost effective water hammer control devices. The simulation results obtained for Loveday irrigation system show that different water hammer events create different levels of pressure surge in different pipes within the system. It indicates that water hammer events generate a severe transient pressure not only in a simple series pipeline, but also in networks.

A criteria for the evaluation of the transient events has been developed by considering the total length of pipes where the transient pressure surge by a water hammer event is greater than a threshold of transient pressure heads. It provides engineers with a method for identifying the critical water hammer events for the cost effective and transient protective design of water distribution systems. The identification of the critical water hammer events is particularly useful for the optimisation of design of water distribution systems including water hammer loadings, which will be discussed in Chapter 9.

9. OPTIMAL TRANSIENT DESIGN OF WATER DISTRIBUTION NETWORKS

9.1 Introduction

The research to this point in the thesis has been undertaken to optimise the selection of pipe diameters such that the pipeline cost is minimised. A complete design, however, includes not only selecting pipe diameter sizes but also selecting pipe wall thicknesses (referred to as pipe classes) and sizing water hammer control devices such as surge tanks. Thus a comprehensive optimisation of the water distribution system should be carried out not only under steady state flow conditions but also under unsteady state conditions where water hammer may occur. The maximum transient pressure governs the selection of pipe classes.

Different genetic algorithm paradigms including the standard GAs and the messy GAs have been compared for the optimisation of water distribution networks in earlier Chapters. It has been shown that the fast messy GA is the most efficient GA paradigm for the discrete optimisation of the water distribution networks. In this Chapter, an optimisation model has been developed for optimal design of the networks including consideration of both steady state and water hammer loadings. This model is integrated with the transient model HAMMER developed in Chapter 8. Both models are further incorporated into the computer program fmGANET described in Chapter 6. The integrated approach provides engineers a comprehensive methodology for optimisation of water distribution networks including consideration of water hammer pressures. Two case studies are presented to investigate the efficiency and effectiveness of the integrated models. The first example is a hypothetical branched system. The search space was completely enumerated to check the optimality of the solution found by the fast messy GA. The second example is a real water irrigation system in South Australia. The network is fairly complicated to demonstrate the application of the technique.

9.2 Design for Transient or Water Hammer Events

Water distribution systems are costly to build, operate and maintain. One source of the system loading that may cause component failure including pipe breakage and that is sometimes neglected in water distribution system analysis is water hammer or transient flow conditions. As shown in Chapter 6, transient events such as closing outlet valves may generate quite high transient pressures even in a complex looped water distribution system. However, the necessity of taking the transient loadings into account in the design procedure for design of water distribution network is controversial.

Transient analysis is a complicated area, and thus is often referred to experts. The complexity of the transient analysis has led practitioners to adopt either a design with simplified transient loading or a high-safety design without considering water hammer loading. The former facilitates that the complex components and other complications in the physical system itself may be ignored and that the range of loading and operation conditions is reduced. This may lead to the underdesign of these systems. The latter usually leads an overdesign of the water distribution systems. To break the cycle of the current standing-in design procedure for water distribution networks, the comprehensive transient analysis must be included in the design procedure.

Over the last decade, a great effort has been made at improving the accuracy and efficiency of transient analysis. Particularly, advancement of computer technology in terms of speed and memory size make it feasible to model transient behaviour of large complex networks. This provides an opportunity for developing a methodology for optimisation of design of the water distribution networks to include consideration of transient conditions. The fast messy GA, the most efficient search algorithm, has been integrated with the hydraulic transient model developed in this research. Employing state-of-the-art optimisation techniques and computer technology as a implementation vehicle will tremendously reduce the cost associated with the transient analysis and design.

9.3 A Model for Optimal Transient Design

A complete design of water distribution system often requires selecting pipe diameter and pipe class (i.e. wall thickness) for each pipe, and also requires sizing the transient pressure relief devices such as surge tanks, which are often used as pressure surge protection devices in water distribution systems. Optimisation of such a complete design is to search for the best combination of the pipe sizes, pipe wall thicknesses and surge tank sizes such that the total cost of the system components is minimised subject to satisfying the constraints including minimum allowable pressure heads, available pipe sizes and maximum allowable transient pressure heads.

The objective is to search for a set of pipe diameters $D=(d_1, d_2, d_3, ..., d_N)$ and pipe classes $CLS=(cls_1, cls_2, cls_3, ..., cls_N)$ for each pipe in the network, where d_n is the diameter of the n-th pipe and cls_n is the class of the n-th pipe. Pipe sizes (pipe diameter and pipe class) of each pipe are selected from a list of commercially available pipe sizes, namely $\forall d_n \in D^0 = (d_0^0, d_1^0, ..., d_K^0)$ and $\forall cls_n \in CLS^0 = (cls_0^0, cls_1^0, ... cls_{CI}^0)$. The unit cost of every pipe is the function of the pipe diameter and pipe class, that is noted as $c_n(d_n, cls_n)$. Thus the cost of the pipeline networks is given as

$$C_{pipe}(D, CLS) = \sum_{n=1}^{N} c_n(d_n, cls_n)L_n \tag{9.1}$$

where L_n is the length of pipe n and N is the total number of the pipes. The cost of the surge tank is the function of tank diameter D_{tank} and tank height H_{tank}. The unit cost of the tank is given as the cost per unit perimeter area of the surge tank, thus the tank cost is formulated as

$$C_{tank} = \pi c_{tank} D_{tank} H_{tank} \tag{9.2}$$

where c_{tank} = the cost of unit perimeter area of the surge tank, H_{tank} = the height of the surge tank determined by the maximum transient pressure head and D_{tank} = the diameter of the tank. The diameter of the surge tank is the variable to be optimised.

The total cost C_{total} of the pipeline networks is the sum the pipeline cost C_{pipe} and the surge tank cost C_{tank}, given as

$$C_{total} = C_{pipe} + C_{tank} \tag{9.3}$$

The model for the optimisation of transient design for water distribution systems is formulated to minimise the total cost of the water distribution network by searching for the optimal pipe diameters, pipe wall thicknesses (i.e. pipe classes) and the surge tank diameter such that the design alternative meets the minimum allowable pressure head requirements and the maximum transient pressure heads. It can be mathematically written as follows.

searching for (D, CLS, D_{tank})

minimising C_{total}

subject to

$$\begin{aligned}
&\forall d_n \in D^0 = \{d_k^0, k = 0, \ldots, K\} \\
&\forall cls_n \in CLS^0 = \{cls_{ci}^0, ci = 0, \ldots, CI\} \\
&\forall D_{tank} \in D_{tank}^0 = \{d_{tn}^0, tn = 1, \ldots, TN\} \\
&H_{i,j} \geq H_j^{min}, j = 1, \ldots, J \quad i = 1, \ldots, I \\
&H_{tr,n}^{srg} < H_n^{max}, n = 1, \ldots, N \quad tr = 1, \ldots, TR
\end{aligned}$$

where d_k^0 = k-th commercially available pipe diameter in set D^0; K = number of the commercially available pipe sizes; cls_n = pipe class of the n-th pipe; cls_{ci}^0 = the ci-th commercially available pipe class; CI = the number of the commercially available pipe classes; $H_{i,j}$ = hydraulic grade at node j under steady state loading i; I = number of steady state loadings; H_j^{min} = minimum allowable hydraulic grade at node j under steady state demand loadings; and J = number of nodes in system (excluding fixed grade nodes); $H_{tr,n}^{srg}$ = maximum transient pressure head of pipe n under transient loading tr; TR = number of transient loadings; and H_n^{max} = maximum allowable transient pressure head of pipe n.

9.4 Incorporation of HAMMER into fmGANET

Optimisation of the water distribution systems, formulated earlier, is required to satisfy the minimum pressure heads under steady state demand conditions and to withstand the maximum pressure heads at water hammer events, Thus both steady state simulation and transient simulation are required for providing the pressure head information for the optimisation procedure. The simulation of water hammer provides the maximum transient pressure for each pipe. This information of the maximum pressure of the pipes is used to guide the optimisation procedure to search for the optimal combination of pipe wall thickness (or classes), while the pressure heads at normal steady state operation conditions, simulated by steady state solver such as EPANET, are used to search for the optimal combination of pipe diameters.

The transient model developed and verified in Chapter 8 has been incorporated into the fast messy GA (fmGANET) to enable the optimal transient design of water distribution systems. The integrated system is called fmGAHAM as shown in Figure 9-1. The fmGAHAM is an extension of fmGANET that was implemented by integrating the fast messy GA with EPANET in Chapter 6. The extended fmGANET is able to carry out optimisation of design and rehabilitation under steady and unsteady state loadings by searching for the cost effective combination of pipe diameters and classes for new pipes to be added to the system, the optimal rehabilitation strategy for the existing pipes and the best sizes of surge tanks for protection of transient pressure surge. Figure 9-1 shows that the fmGAHAM has a similar programme structure to fmGANET except the transient model HAMMER is integrated into the system. For a problem of design that requires sizing the pipe classes and surge devices, the transient simulations are performed for each of the design alternatives (combinations of possible pipe classes and surge protection devices) before evaluating the cost of the solution. The cost of the transient design solution is associated with not only pipe diameters but also the pipe classes. The total cost is the sum of the pipe cost and surge tank cost as described in last section.

The integrated approach for optimisation of water distribution system including water hammer loading has been applied to two case studies in the following sections. The first case study is a hypothetical and simple pipeline system, which is used to test the effectiveness of the implemented model. The second case study is for a real irrigation network in South Australia.

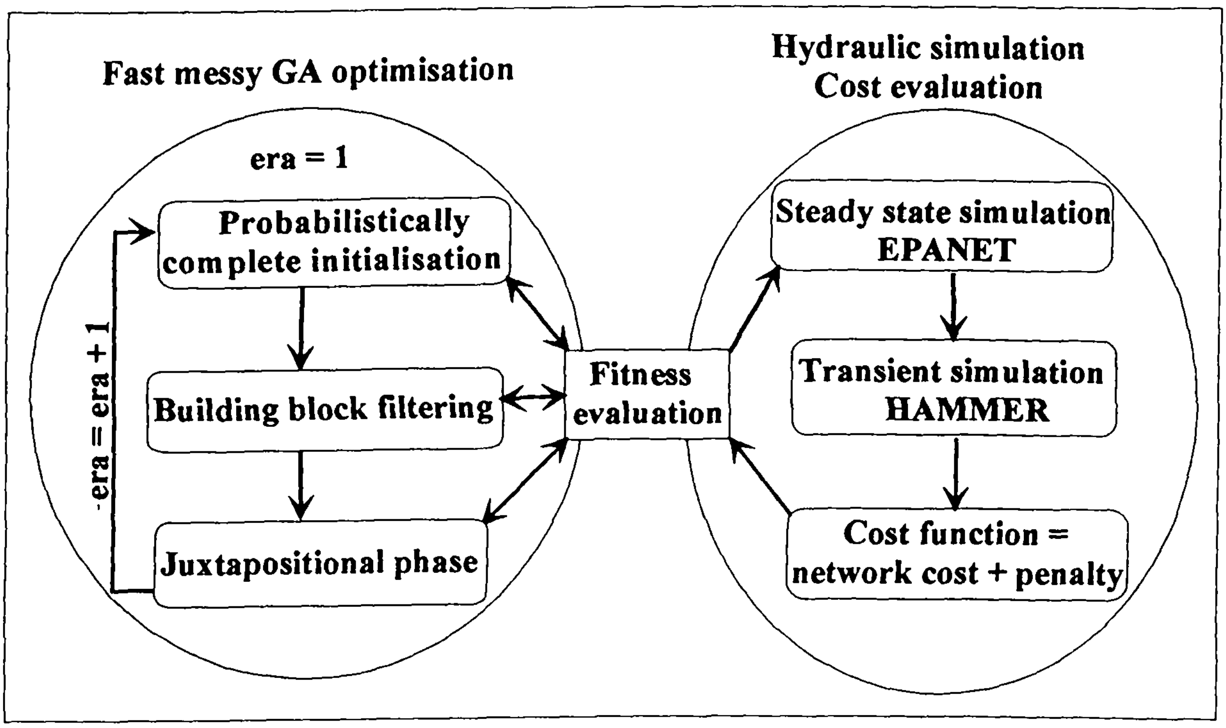


Figure 9-1 Conceptual Structure of The Program fmGAHAM for Optimal Transient Design

9.5 A Simple Pipeline Case Study

A pipeline, as shown in Figure 9-2, is a simple hypothetical water system established to test the integrated approach for the optimal transient design. The pipeline system consists of a reservoir at the upstream end, a surge tank between the junction and the reservoir, and also two branches at the downstream of the surge tank.

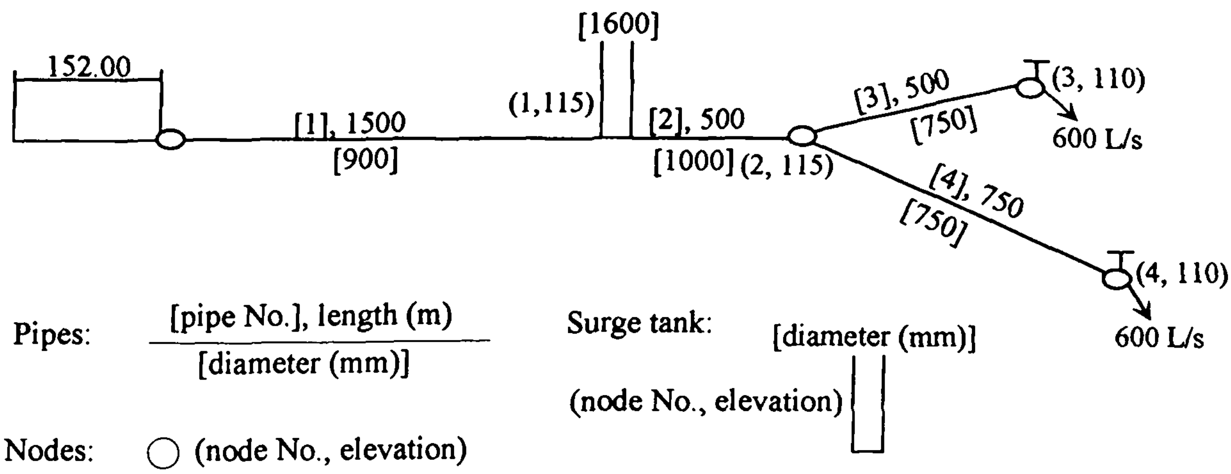


Figure 9-2 Layout of a Simple Hypothetical Pipeline System

Table 9-1 Steady State Flow of the Hypothetical System Under Different Demand Loadings

Flow velocities (m/s)				Hydraulic grade (m)			
Pipe Id	Demand nodes: Node 4	Node 3	Node 3 and 4	Node Id	Demand nodes: Node 4	Node 3	Node 3 and 4
[1]	0.94	0.94	1.89	1	151.06	151.06	148.56
[2]	0.76	0.76	1.53	2	150.87	150.87	147.88
[3]	0.00	1.36	1.36	3	150.87	150.11	147.12
[4]	1.36	0.00	1.36	4	149.72	150.87	146.73

It is Assumed that there are two outlet valves at downstream of the system. Each of the two valves is to supply 600 L/s. A Hazen-William coefficient C of 120 is assumed for all the pipes. The steady state flow simulated by using EPANET is given in Table 9-1. The system will be designed to meet a minimum allowable pressure head of 30 metres at all nodes under a steady state operating condition and to withstand the maximum water hammer pressure at transient events generated by operating the two outlet valves in 10 seconds. The available pipe sizes and associated classes are given in Table 9-2 while the available sizes for surge tank are given in Table 9-3. The maximum allowable pressure heads for class 4, 6 and 10 are 40, 60 and 100 meters respectively.

Table 9-2 Available Pipe Sizes and Associated Cost

Diam.	Class 4		Class 6		Class 10	
(mm)	cost ($/m)	thickness (mm)	cost ($/m)	thickness (mm)	cost ($/m)	thickness (mm)
750	233.94	16.00	250.13	15.00	296.4	15.00
900	273.60	17.00	294.39	17.00	343.88	16.00
1000	325.70	19.00	354.56	18.00	413.70	18.00
1200	425.02	23.00	459.23	22.00	555.10	21.00

Hobas pipe (as shown in Table 9-2) is made stronger by adding more resin to the pipe wall mixture rather than necessarily by increasing the pipe wall thickness.

Table 9-3 Available Surge Tank Sizes and Associated Costs

Surge tank diameter (m)	Unit height cost ($/m)
1.00	600.00
1.5	800.00
2.00	1000.00
2.50	1200.00

9.5.1 Critical transient loading

In order that the hypothetical water system is designed to be able to withstand the most severe water hammer pressure, the transient event that generates the maximum transient pressure in the system needs to be identified. All the water hammer events that can possibly occur in this simple system are as follows.

(1) Event 1: valve 3 is open while valve 4 is shut in 10 seconds;

(2) Event 2: valve 3 is shut in 10 second while valve 4 is kept open;

(3) Event 3: no demand at valve 3 while valve 4 is shut in 10 seconds;

(4) Event 4: no demand at valve 4 while valve 3 is shut in 10 seconds.

A water hammer model has been established by using the transient simulation program HAMMER developed in Chapter 8. It is assumed that the pipes are made of steel (Young's modulus $E = 2.07 \times 10^{11}$) and are class of 4. The corresponding wave speed of each pipe is computed by Eq. (8.3) in Chapter 8. The information of the pipes and nodes is shown as Figure 9-2. The minimum available pipe diameter of 750mm is used for pipe [3] and [4], which is believed to generate the severe transient pressure at the upstream of the outlet valves. The four transient events given above are simulated and the results of the maximum transient pressure head for each pipe over four water hammer loadings are shown in Figure

9-3. It is observed that the transient events 3 and 4 are the severe water hammer loading. Events 3 and 4 are considered for the optimal transient design of the hypothetical system.

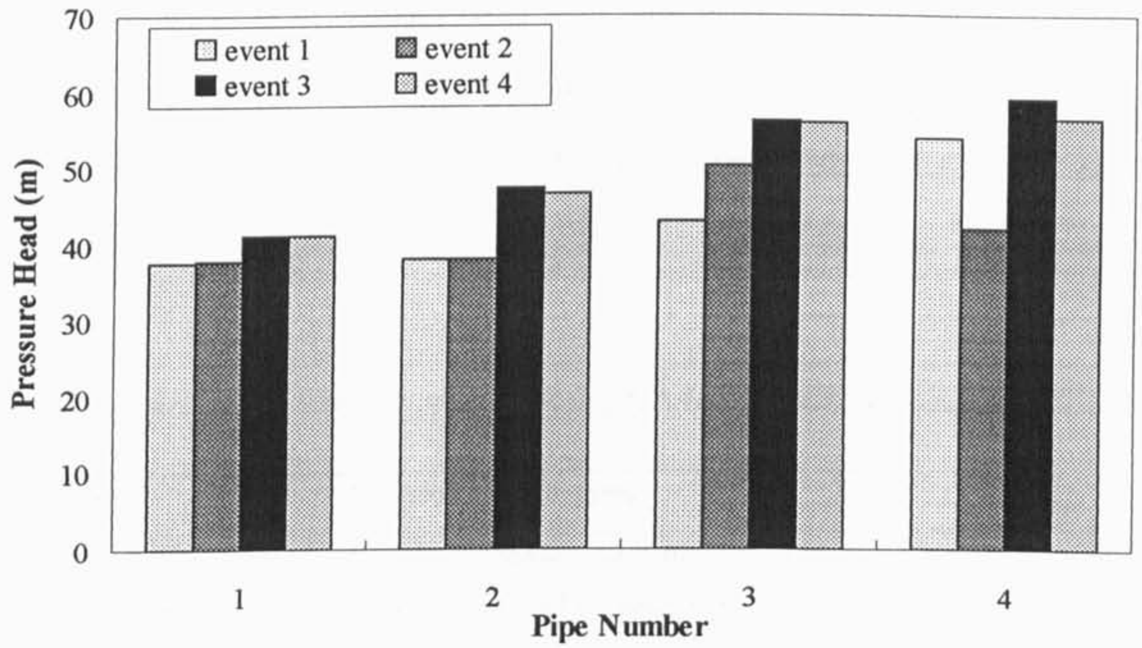


Figure 9-3 The Maximum Transient Pressure Head of the Hypothetical Pipeline System

9.5.2 Optimal solution

The optimal transient design of the hypothetical pipeline system has been found by using fmGAHAM under severe transient event 3 and 4 as identified in section 9.5.1. Two binary bits were used to code 3 pipe classes for each of 4 pipes while 2 binary bits were used to code 4 pipe diameter sizes for each of 4 pipes and another 2 binary bits were used to code 4 surge tank diameters. A total of 18 binary bits were used for the genotype coding of a solution for the hypothetical pipeline system. The fast messy GA parameters used for this case study are given in Table 9-4. The edge search of the genetic algorithm optimisation by adapting the penalty factor as described in Chapter 7 has been employed in this case study.

The initial lower and upper bounds of penalty factor are \$5000/m and \$100,000/m respectively. The final penalty adapted for the optimal solution is \$20,000/m.

The optimal solution of the system including transient design for the hypothetical system is shown in Table 9-5. The steady state pressure heads at nodes and the maximum transient pressure heads for each pipe are given in Table 9-6 and 9-7. It shows that the solution found by fmGAHAM meets all the steady state and transient design criteria conditions.

Table 9-4 Parameters Used by the Fast Messy GA for the Simple Case Study

Parameters	Value	Parameters	Value
Splice probability	1.000	Maximum number of eras	3
Cut probability	0.016	Initial population size	90
Allelic mutation probability	0.030	juxtapositional population size	100
Genic mutation probability	0.030	Maximum number of generations in a era	20
Thresholding	on	random seed	0.76

This optimal result has been verified by enumerating all the possible alternatives for the hypothetical system. Table 9-8 shows top 20 optimal solutions for the transient design of the simple system. The enumerative solutions verify the effectiveness of the fast messy GA approach for the optimal transient design of water distribution systems. The convergence of the fast messy GA approach has been shown in Figure 9-4. It shows that the fast messy GA found the optimal solution by evaluating about 1,300 possible solutions. The total search space for the simple problem is 82,944 design alternatives. The fast messy GA reached the global optimal solution by searching about 1.5% of the search space.

Table 9-5 Optimal Transient Design Solution of the Hypothetical System

Components	Diameter (mm)	Pipe class or surge tank height (m)
Pipe 1	750	4
Pipe 2	900	6
Pipe 3	750	6
Pipe 4	750	6
Surge tank	2500	39.7

Table 9-6 Steady State Pressure of the Optimal Transient Design Solution

Node	Pressure head (m)	Minimum pressure head (m)	Pressure excess
1	34.70	30.00	4.70
2	34.39	30.00	4.39
3	39.39	30.00	9.39
4	38.23	30.00	8.23

Table 9-7 The Maximum Water Hammer Pressure of the Optimal Transient Design

Pipe ID	Maximum pressure head (m)	Pipe class	Allowable maximum pressure head (m)	Allowable pressure head residual (m)
1	39.70	4	40.00	0.30
2	46.98	6	60.00	13.02
3	56.49	6	60.00	3.51
4	59.04	6	60.00	0.96

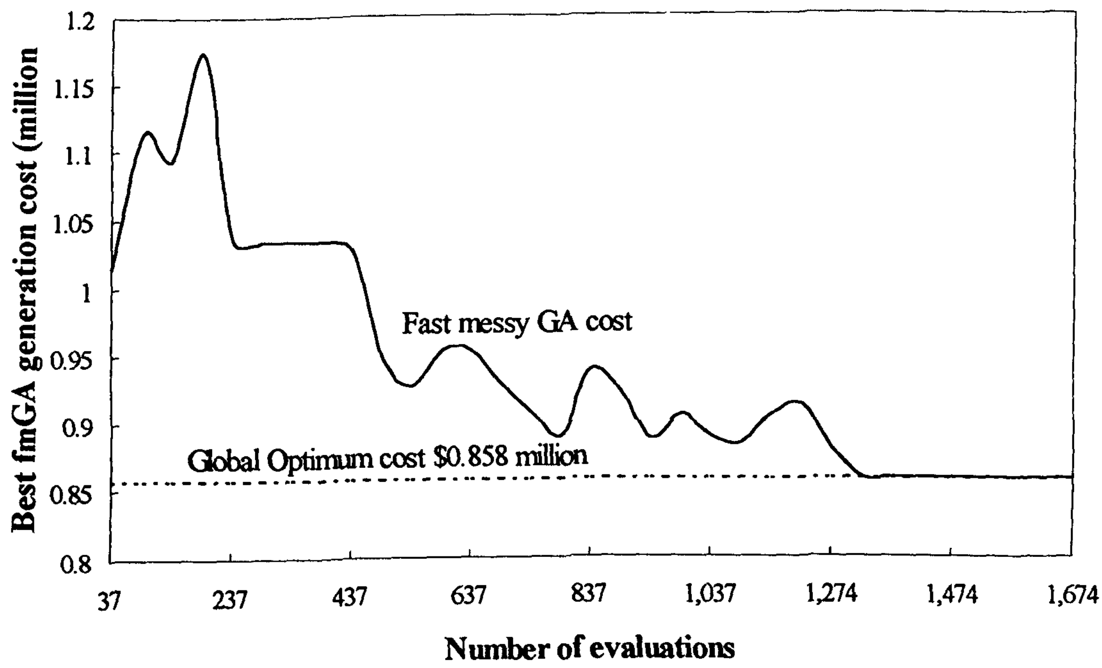


Figure 9-4 Convergence of the fmGA Search for the Optimal Transient Design of the Hypothetical Pipeline System

Table 9-8 Lowest Cost 20 Transient Solutions for the Simple Hypothetical Pipeline System by Complete Enumeration

Solution rank	Pipe 1		Pipe 2		Pipe 3		pipe 4		Surge tank		Cost (Dollars)
	Diam. (mm)	Class	Diam. (mm)	Class	Diam. (mm)	Class	Diam. (mm)	Class	Height (m)	Diam. (mm)	
1	750	4	900	6	750	6	750	6	39.7	2,500	858,406
2	750	4	750	6	750	6	750	10	39.7	2,500	866,656
3	750	4	900	6	750	10	750	6	39.7	2,500	878,661
4	750	4	900	6	900	6	750	6	39.7	2,500	880,530
5	750	6	900	6	750	6	750	6	39.7	2,500	882,691
6	750	4	900	10	750	6	750	6	39.7	2,500	883,151
7	750	6	750	6	750	6	750	10	41	2,000	884,354
8	750	4	750	6	750	10	750	10	39.7	2,500	886,911
9	750	4	750	10	750	6	750	10	39.7	2,500	886,911

Solution rank	Pipe 1		Pipe 2		Pipe 3		pipe 4		Surge tank		Cost (Dollars)
	Diam. (mm)	Class	Diam. (mm)	Class	Diam. (mm)	Class	Diam. (mm)	Class	Height (m)	Diam. (mm)	
10	750	4	1000	6	750	6	750	6	39.7	2,500	888,488
11	750	4	750	6	900	6	750	10	39.7	2,500	888,777
12	750	4	900	6	750	6	750	10	39.7	2,500	888,789
13	750	6	750	6	750	6	750	10	39.7	2,500	890,941
14	750	4	900	6	750	6	900	6	39.7	2,500	891,590
15	750	6	750	6	750	10	750	10	46.9	1,000	891,688
16	750	6	750	6	750	10	750	10	43	1,500	897,944
17	750	6	1000	6	750	6	750	6	43	1,500	899,520
18	750	6	750	6	900	6	750	10	43	1,500	899,824
19	750	6	900	6	750	10	750	6	39.7	2,500	902,946
20	750	4	900	10	750	10	750	6	39.7	2,500	903,406

9.6 A Real System Case Study — Loveday Irrigation Network

The comprehensive transient analysis for Loveday network in Chapter 8 shows that there is no single transient event that governs the design of the network. The network must be designed under multiple transient loadings, namely the transient events generated by closing one of three outlet valves 28, 31 and 14 with the other two valves shut are considered in optimal transient design. The fast messy GA has been employed to carry out the optimal transient design in two stages. It has been observed that optimisation of pipe diameters, pipe classes and surge tank in one model becomes very complicated and also requires massive computation. Thus a two-stage optimisation approach is proposed to solve the problem of optimal transient design of the real network. In the first stage, the optimal design is carried out by just considering steady state demand loadings to search for the optimal pipe diameters. Then the optimal pipe diameters are locked in and the second stage allows the optimal model search for the optimal pipe classes and surge tank dimensions by considering the three transient loadings.

9.6.1 Stage-one optimisation

In the first stage, the optimisation of design of Loveday network was carried out by just considering the steady state demand loadings as given in Table 9-9. Available pipe sizes (diameters), as shown in Table 9-10, are the combination of the concrete pipes and the uPVC pipes. There are 3 classes available for each pipe size, but only class I pipes (the minimum cost class) were considered in this stage. The concrete pipes provide large sizes for the trunk main of the network while the uPVC pipes are mainly used for the branches in the downstream portion of the network.

The boundary fast messy GA search approach developed in Chapter 7 has been employed for the stage-one optimisation of Loveday network. There are 54 pipes to be optimised in the stage-one optimisation. Four binary bits were used for each of 54 pipes to represent 11 pipe sizes, and 4 binary bits were used to coding the penalty factor in an interval from 5,000,000 to 10,000,000. A total of 220 binary bits were used for the genotype coding of an alternative for stage-one optimisation. The fast messy GA parameters were initial population sizes = 200; juxtapositional population size = 500; maximum era = 20; thresholding = on; tiebreaking = off; bitwise cut probability = 0.0166; splice probability = 0.95; and mutation probability = 0.01.

Table 9-9 Steady State Demand Loadings for Loveday Network

Node Id	Demand (L/s) loading 1	Demand (L/s) loading 2	Node Id	Demand (L/s) loading 1	Demand (L/s) loading 2
2	0.00	0.00	28	10.00	80.00
3	0.00	0.00	29	30.00	16.00
4	0.00	0.00	30	90.00	47.00
5	0.00	0.00	31	20.00	120.00
6	60.00	31.00	32	0.00	0.00
7	60.00	31.00	33	130.00	67.00
8	60.00	31.00	34	10.00	120.00
9	0.00	0.00	35	40.00	21.00
10	80.00	41.00	36	0.00	0.00
11	55.00	120.00	37	140.00	73.00
12	0.00	0.00	38	30.00	16.00
13	0.00	0.00	39	0.00	0.00
14	40.00	120.00	40	25.00	120.00
15	60.00	31.00	41	30.00	16.00
16	0.00	0.00	42	0.00	0.00
17	0.00	0.00	43	120.00	120.00
18	0.00	0.00	44	60.00	120.00
19	0.00	0.00	45	45.00	0.00
20	80.00	41.00	46	45.00	120.00
21	0.00	0.00	47	30.00	0.00
22	80.00	41.00	48	30.00	120.00
23	90.00	47.00	49	240.00	240.00
24	20.00	10.00	50	60.00	31.00
25	60.00	31.00	51	0.00	0.00
26	60.00	31.00	52	60.00	31.00
27	70.00	36.00	-	-	-

Table 9-10 Available Pipe Sizes and Associated Cost for Stage-One Optimisation

Pipe diameter (mm)	Material	Cost ($/m)
233.60	uPVC	50.00
258.20	uPVC	74.00
310.30	uPVC	77.00
384.40	uPVC	100.00
457.00	Concrete	128.00
534.00	Concrete	143.00
610.00	Concrete	153.00

Pipe diameter (mm)	Material	Cost ($/m)
762.00	Concrete	170.00
915.00	Concrete	252.00
1066.00	Concrete	305.00
1219.00	Concrete	410.00

Table 9-11 Optimal Solutions for Loveday Network Under Steady State Demand Loadings

Pipe Id	Optimal diameters (mm)						
	fmGA1	fmGA2	fmGA3	fmGA4	fmGA5	fmGA6	fmGA7
1	1219.0	1219.0	1219.0	1219.0	1219.0	1219.0	1219.0
2	1219.0	1219.0	1219.0	1219.0	1219.0	1219.0	1219.0
3	310.3	310.3	310.3	310.3	310.3	310.3	310.3
4	1219.0	1219.0	1219.0	1219.0	1219.0	1219.0	1219.0
5	534.0	610.0	1066.0	610.0	915.0	762.0	762.0
6	534.0	534.0	384.4	384.4	534.0	534.0	534.0
7	1219.0	1219.0	915.0	1219.0	1066.0	1219.0	1066.0
8	762.0	762.0	1066.0	762.0	915.0	762.0	762.0
9	534.0	384.4	1066.0	457.0	915.0	762.0	762.0
10	384.4	384.4	1066.0	384.4	915.0	384.4	762.0
11	310.3	233.6	1066.0	384.4	762.0	384.4	762.0
12	233.6	233.6	915.0	310.3	762.0	384.4	610.0
13	233.6	233.6	1066.0	384.4	762.0	384.4	610.0
14	762.0	762.0	915.0	762.0	534.0	534.0	233.6
15	457.0	457.0	384.4	534.0	384.4	762.0	534.0
16	1219.0	1219.0	1066.0	1219.0	1066.0	1219.0	1219.0
17	1066.0	1219.0	762.0	1066.0	1066.0	1066.0	1066.0
18	310.3	310.3	310.3	384.4	384.4	310.3	310.3
19	1219.0	1066.0	1066.0	1066.0	915.0	1066.0	1066.0
20	1219.0	1219.0	762.0	1066.0	1066.0	1066.0	1066.0
21	310.3	310.3	310.3	310.3	310.3	310.3	310.3
22	1219.0	1219.0	762.0	1219.0	1066.0	1219.0	1066.0
23	1066.0	1219.0	762.0	1219.0	1066.0	1066.0	1066.0
24	534.0	310.3	762.0	610.0	762.0	1066.0	915.0
25	1219.0	1219.0	762.0	1066.0	762.0	1066.0	762.0
26	534.0	457.0	762.0	1066.0	762.0	915.0	1066.0
27	384.4	233.6	762.0	1066.0	762.0	915.0	915.0
28	233.6	762.0	610.0	915.0	610.0	762.0	1066.0
29	762.0	762.0	233.6	610.0	384.4	762.0	762.0
30	762.0	762.0	762.0	610.0	610.0	762.0	762.0

Pipe Id	Optimal diameters (mm)						
	fmGA1	fmGA2	fmGA3	fmGA4	fmGA5	fmGA6	fmGA7
31	762.0	610.0	762.0	534.0	534.0	534.0	762.0
32	457.0	457.0	457.0	534.0	534.0	534.0	457.0
33	457.0	457.0	384.4	384.4	384.4	384.4	384.4
34	310.3	310.3	384.4	384.4	384.4	384.4	384.4
35	384.4	384.4	457.0	457.0	384.4	384.4	384.4
36	534.0	384.4	384.4	762.0	233.6	534.0	762.0
37	384.4	384.4	310.3	384.4	233.6	534.0	384.4
38	384.4	258.2	310.3	310.3	384.4	258.2	258.2
39	233.6	610.0	610.0	762.0	457.0	762.0	762.0
40	1066.0	1066.0	762.0	610.0	762.0	384.4	233.6
41	915.0	915.0	762.0	384.4	534.0	233.6	384.4
42	762.0	1066.0	762.0	384.4	310.3	233.6	384.4
43	762.0	384.4	384.4	233.6	610.0	233.6	233.6
44	610.0	384.4	457.0	233.6	610.0	233.6	233.6
45	384.4	310.3	310.3	384.4	384.4	384.4	384.4
46	1066.0	1219.0	233.6	762.0	610.0	762.0	384.4
47	384.4	384.4	384.4	384.4	384.4	310.3	384.4
48	384.4	384.4	384.4	384.4	384.4	384.4	384.4
49	310.3	310.3	310.3	310.3	384.4	384.4	310.3
50	384.4	384.4	457.0	384.4	310.3	384.4	384.4
51	310.3	310.3	310.3	310.3	310.3	310.3	310.3
52	384.4	384.4	457.0	457.0	384.4	384.4	384.4
53	381.0	384.4	310.3	258.2	310.3	310.3	310.3
54	384.4	384.4	384.4	384.4	384.4	384.4	384.4
Deficit (m)	0.00	0.00	**0.01**	0.00	**0.03**	0.00	**0.06**
Total cost	6,036,782	6,057,513	5,916,330	5,904,404	5,695,566	5,686,397	5,492,674
Evaluations	395,120	541,820	544,320	271,710	536,820	535,320	538,820

Table 9-12 Pressure Heads and Deficits of Optimal Solutions fmGA4, 5, 6 and 7 for Loveday Network at Critical Nodes (m)

Node	fmGA4		fmGA5		fmGA6		fmGA7	
ID	head	excess	head	excess	head	excess	head	excess
10	3.86	0.86	5.47	2.47	3.11	0.11	3.81	0.81
11	3.02	0.02	3.78	0.78	3.02	0.02	2.94	**-0.06**
12	3.97	0.97	6.46	3.46	3.96	0.96	3.88	0.88
23	3.38	0.38	3.79	0.79	4.04	1.04	3.84	0.84
24	3.15	0.15	2.97	**-0.03**	3.29	0.29	3.65	0.65
26	3.56	0.56	3.38	0.38	4.14	1.14	4.57	1.57
28	3.33	0.33	3.33	0.33	3.52	0.52	3.67	0.67
31	5.05	2.05	3.64	0.64	3.15	0.15	3.23	0.23
34	3.02	0.02	3.01	0.01	3.37	0.37	2.99	**-0.01**
37	5.77	2.77	6.08	3.08	3.01	0.01	3.09	0.09
38	4.38	1.38	4.41	1.41	5.28	2.28	3.19	0.19
39	4.36	1.36	4.16	1.16	4.16	1.16	2.95	**-0.05**
40	5.30	2.30	5.09	2.09	5.97	2.97	3.88	0.88
43	4.35	1.35	3.17	0.17	3.17	0.17	3.17	0.17
46	3.11	0.11	7.66	4.66	3.15	0.15	2.94	**-0.06**
49	3.36	0.36	2.97	**-0.03**	3.02	0.02	3.31	0.31

The fmGANET has been run by using different random seeds, a set of cost effective solutions have been obtained as given in Table 9-11. It shows that the boundary fast messy GA reached the optimal or near-optimal solutions by evaluating about 540,000 alternatives in a total possible combination of 2^{220}, which is about 1.68 x 10^{66}. The boundary fast messy GA has been shown very efficient at searching for the optimal and/or near-optimal solutions.

The optimal or near-optimal solutions found in this stage are observed to be very close to the boundary of the hydraulic pressure requirements as shown in Table 9-12. The pressure heads of 4 best solutions at critical nodes are given in Table 9-12. Although the minimum cost solution fmGA 7 has the maximum pressure head violation of 0.06 metres, it is considered the optimal solution from the first stage optimisation and will be adopted in stage-two optimisation.

Table 9-13 Material Data of Pipe Class I

Diameter (mm)	Cost ($/m)	Wall thickness (mm)	Allowable pressure (m)
233.60	50.00	9.70	90.00
258.20	74.00	10.70	90.00
310.30	77.00	12.90	90.00
384.40	100.00	15.90	90.00
457.00	128.00	38.00	52.00
534.00	143.00	41.00	41.20
610.00	153.00	44.00	30.00
762.00	170.00	51.00	32.00
915.00	252.00	63.00	32.00
1066.00	305.00	63.00	32.00
1219.00	410.00	63.00	32.00

Table 9-14 Material Data of Pipe Class II

Diameter (mm)	Cost ($/m)	Wall thickness (mm)	Allowable pressure (m)
233.60	60.00	12.00	120.00
258.20	86.00	14.00	120.00
310.30	92.00	17.00	120.00
384.40	130.00	20.90	120.00
457.00	140.00	54.00	53.00
534.00	148.00	51.00	47.10
610.00	158.00	57.00	41.20
762.00	210.00	63.00	37.50
915.00	280.00	76.00	37.50
1066.00	370.00	76.00	37.50
1219.00	545.00	76.00	34.30

Table 9-15 Material Data of Pipe Class III

Diameter (mm)	Cost ($/m)	Wall thickness (mm)	Allowable pressure (m)
233.60	75.00	16.80	160.00
258.20	102.00	18.50	160.00
310.30	112.00	22.20	160.00
384.40	140.00	27.50	160.00
457.00	150.00	70.00	55.90
534.00	158.00	61.00	51.00
610.00	178.00	76.00	48.00
762.00	255.00	76.00	42.00
915.00	312.00	89.00	42.00
1066.00	425.00	89.00	42.00
1219.00	680.00	89.00	42.00

9.6.2 Stage-two optimisation

In the second stage, the pipe classes and the surge tank of the Loveday network have been optimised by using the optimal pipe diameters found in stage-one. There are 3 classes for each of 11 available pipe diameters as shown in Table 9-13, 9-14 and 9-15. The surge tank is allocated at node 23 and a cost of $\$245/m^2$ of the tank perimeter surface area. The unit tank cost is based on the construction cost information provided by the pipeline contractors in South Australia. It was used for evaluation of the tank alternatives. Two cases including optimisation of pipe classes and optimisation of pipe classes and surge tank have been carried out for Loveday network in this stage. The optimal diameters of solution fmGA7 obtained in stage-one is fixed in stage-two optimisation.

The boundary GA search strategy has also been employed in stage-two optimisation of Loveday network. Two binary bits were used for the representation of 3 pipe classes for each of 54 pipes, 4 binary bits were used for coding the surge tank diameters in range from 1.0 metre to 3.0 metres in diameter and other 4 binary bits were used for coding the penalty factor in a interval of [500,000 1,000,000]. Thus 112 binary bits were used for the optimisation of the pipe class in stage two, and 116 bits were used for the optimisation of the pipe class and the surge tank. The critical transient loadings, as identified in Chapter 8, are the independent instantaneous closure of valves 28, 31 and 14. The optimal transient design solutions have been achieved by using the same set of the fast messy GA parameters as in the first stage.

Table 9-16 Optimal Solutions of Stage-two Optimisation

Pipe ID	Pipe diameters (mm)	Minimum thickness (mm)	Optimal thickness	
			no tank (mm)	with tank (mm)
1	1219.0	63.0	63.0	63.0
2	1219.0	63.0	63.0	63.0
3	310.3	12.9	12.9	12.9
4	1219.0	63.0	63.0	63.0
5	762.0	51.0	51.0	51.0
6	534.0	41.0	41.0	41.0
7	1066.0	63.0	63.0	63.0
8	762.0	51.0	51.0	51.0
9	762.0	51.0	51.0	51.0
10	762.0	51.0	51.0	51.0
11	762.0	51.0	51.0	51.0
12*	**610.0**	**44.0**	**57.0**	**57.0**
13	**610.0**	**44.0**	**57.0**	**57.0**
14	233.6	9.7	9.7	9.7
15	534.0	41.0	41.0	41.0
16	1219.0	63.0	63.0	63.0
17	1066.0	63.0	63.0	63.0
18	310.3	12.9	12.9	12.9
19	1066.0	63.0	63.0	63.0
20	1066.0	63.0	63.0	63.0
21	310.3	12.9	12.9	12.9
22	1066.0	63.0	63.0	63.0
23	1066.0	63.0	63.0	63.0
24	915.0	63.0	63.0	63.0
25	762.0	51.0	51.0	51.0
26	1066.0	63.0	63.0	63.0
27	915.0	63.0	63.0	63.0
28	1066.0	63.0	63.0	63.0
29	**762.0**	**51.0**	**63.0**	**63.0**

Pipe ID	Pipe diameters (mm)	Minimum thickness (mm)	Optimal thickness	
			no tank (mm)	with tank (mm)
30	762.0	51.0	51.0	51.0
31	**762.0**	**51.0**	**76.0**	**76.0**
32	**457.0**	**38.0**	**54.0**	**54.0**
33	384.4	15.9	15.9	15.9
34	384.4	15.9	15.9	15.9
35	384.4	15.9	15.9	15.9
36	**762.0**	**51.0**	**76.0**	**63.0**
37	384.4	15.9	15.9	15.9
38	258.2	10.7	10.7	10.7
39	**762.0**	**51.0**	**76.0**	**76.0**
40	233.6	9.7	9.7	9.7
41	384.4	15.9	15.9	15.9
42	384.4	15.9	15.9	15.9
43	233.6	9.7	9.7	9.7
44	233.6	9.7	9.7	9.7
45	384.4	15.9	15.9	15.9
46	384.4	15.9	15.9	15.9
47	384.4	15.9	15.9	15.9
48	384.4	15.9	15.9	15.9
49	310.3	12.9	12.9	12.9
50	384.4	15.9	15.9	15.9
51	310.3	12.9	12.9	12.9
52	384.4	15.9	15.9	15.9
53	310.3	12.9	12.9	12.9
54	384.4	15.9	15.9	15.9
Surge tank Diameter (m)		-	-	1.0
Surge tank Hight (m)		-	-	18.0
Pipe cost		5,492,674	5,737,509	5,700,384
Surge tank cost		-	-	24,090
Total cost		5,492,674	5,737,509	5,724,474

*pipes chosen with a higher class than the minimum class are in bold.

The optimal solutions of transient design of Loveday network have been obtained by fmGAHAM for the design without the surge tank and with the surge tank as given as in Table 9-16. The first stage optimal design was obtained by using the cost information of the class I pipes, which correspond to the minimum pipe wall thickness. Thus it resulted in a lower cost solution than the optimal transient design solutions in the second stage. The maximum water hammer pressure heads of the stage-one optimal solution are greater than the allowable transient pressure heads at node 12, 13, 29 and 39. This indicates that the optimal solution under steady state demand loadings has no safety guarantee for the pressure surge protection although it may provide a lower cost design solution.

Table 9-16 and Table 9-17 show that the transient design including the surge tank is the optimal solution for Loveday network, both in terms of the capital cost and the surge pressure protection. The second stage optimal design considering pipe classes without a surge tank chooses higher pipe classes for a few pipes as highlighted in bold in Table 9-16 than the first stage optimal design. Consequently the safety of water hammer protection is guaranteed as shown in Table 9-17. The total cost of the optimal design solutions increased from 5,492,674 in stage one to 5,737,509 in stage two. The optimal transient design solution considering the surge tank at node 23 brings the class down from class III to class II for pipe 36, as shown as in Table 9-16, namely the pipe wall thickness for pipe 36 is reduced from 76.0 mm to 63.00 mm. A surge tank with the height of 18.0 metres and the diameter of 1.0 metre is chosen for the pressure surge protection. Although the tank costs $24,090 the pipe cost decreases more than the cost of the surge tank. The total cost of the optimal transient design with surge tank still provides a lower cost solution than that without a surge tank. It also provides more safety margin for the protection of the water hammer pressures as given in Table 9-17. Therefore we are able to conclude that the optimal transient design is not only able to provide a lower cost design solution but also a safer solution in terms of pressure surge protection.

Table 9-17 Comparison of Transient Pressure Head Residual of Optimal Solutions

Pipe ID	Allowable transient head residual (m)			Pipe ID	Allowable transient head residual (m)		
	Stage-one	Stage-two			Stage-one	Stage-two	
	minimum class	no tank	with tank		minimum class	no tank	with tank
1	13.02	16.69	16.69	28	16.73	7.91	8.58
2	11.71	12.69	12.69	29	**-2.74**	2.72	2.73
3	73.32	69.64	69.64	30	5.38	0.23	0.26
4	11.78	12.69	12.69	31	3.00	2.03	2.05
5	14.76	16.69	16.69	32	8.73	2.21	2.29
6	25.39	25.89	25.89	33	72.92	43.49	41.78
7	14.32	16.69	16.69	34	48.26	36.74	35.33
8	14.80	18.68	18.68	35	50.73	54.51	54.74
9	11.47	18.68	18.68	36	0.95	3.75	1.26
10	11.82	18.68	18.68	37	51.97	51.03	51.79
11	6.88	9.66	12.10	38	17.65	19.15	19.15
12	**-10.56**	5.47	7.96	39	**-7.89**	2.66	2.66
13	**-10.64**	5.47	7.96	40	66.41	52.91	45.78
14	65.67	64.32	66.88	41	66.38	52.62	52.63
15	14.12	20.15	22.26	42	55.26	55.22	55.23
16	12.48	6.25	10.55	43	70.73	52.58	53.05
17	11.81	6.25	10.55	44	64.94	48.37	48.33
18	70.48	59.60	66.11	45	73.73	62.90	67.13
19	16.30	8.92	13.46	46	75.16	62.10	67.64
20	16.28	9.21	14.16	47	66.78	58.62	64.59
21	71.72	61.77	68.02	48	75.16	50.31	65.13
22	16.39	9.40	14.16	49	60.20	55.24	63.50
23	18.03	9.94	14.85	50	59.93	54.63	64.52
24	13.92	7.75	13.35	51	69.23	54.82	59.61
25	16.03	7.64	12.81	52	77.94	76.64	76.64
26	13.92	6.53	12.74	53	69.72	57.62	65.58
27	12.09	6.00	12.56	54	77.03	60.24	67.19

9.7 Sensitivity of Optimal Transient Solution

Section 9.6 shows that the two-stage optimisation is effective at finding a low cost solution of the transient design of Loveday network. The optimal transient design, however, is based on the optimal diameters obtained in stage-one by using the minimum thickness (class I) pipes. A question arises as to how sensitive the optimal transient solution is to the pipe class being fixed in stage-one optimisation. In other words, does the approach produce a similar low cost solution of the transient design based on optimal diameters found in stage-one using pipes of another class. In this section, the optimisation of transient design of Loveday network has been carried out by using the stage-one optimal diameters of class II pipes. First of all, the stage-one optimal diameters of Loveday irrigation network have been obtained by using the cost information of class II pipes. The stage-one optimal diameters are then used for the stage-two optimisation of transient design of Loveday network. The results from both stage one and two are compared with that obtained in Section 9.6 using the cost information of class I pipes in stage one.

Table 9-18 Comparison of Stage-one Optimal Solutions Using Pipe Class I and II Cost

Pipe ID	Optimal pipe diameter using class I unit cost (mm)	Optimal pipe diameter using class II unit cost (mm)	Pipe ID	Optimal pipe diameter using class I unit cost (mm)	Optimal pipe diameter using class II unit cost (mm)
1	1219	1219	28	1066	762
2	1219	1219	29	762	762
3	310.3	310.3	30	762	762
4	1219	1219	31	762	534
5	762	762	32	457	534
6	534	384.4	33	384.4	384.4
7	1066	1066	34	384.4	384.4
8	762	762	35	384.4	610
9	762	762	36	762	233.6
10	762	610	37	384.4	233.6

Pipe ID	Optimal pipe diameter using class I unit cost (mm)	Optimal pipe diameter using class II unit cost (mm)	Pipe ID	Optimal pipe diameter using class I unit cost (mm)	Optimal pipe diameter using class II unit cost (mm)
11	762	762	38	258.2	310.3
12	610	762	39	762	457
13	610	762	40	233.6	610
14	233.6	233.6	41	384.4	610
15	534	457	42	384.4	233.6
16	1219	1066	43	233.6	610
17	1066	1066	44	233.6	534
18	310.3	384.4	45	384.4	384.4
19	1066	1066	46	384.4	762
20	1066	1066	47	384.4	384.4
21	310.3	310.3	48	384.4	384.4
22	1066	1066	49	310.3	384.4
23	1066	1066	50	384.4	381
24	915	915	51	310.3	310.3
25	762	762	52	384.4	457
26	1066	762	53	310.3	310.3
27	915	915	54	384.4	384.4
Pressure deficit (m)				0.06	0.03
Total cost evaluated by using class I pipe cost				5,492,674	5,596,651*
Total cost evaluated by using class II pipe cost				6,764,152*	6,710,189
Number of genetic algorithm evaluations				538,820	496,940

*total cost evaluated after optimisation for comparison.

The stage-one optimisation model of Loveday network was rerun by using the cost information of class II pipes. The cost variation of class II pipes, as shown in Figure 9-5, is similar to that of class I pipes. The other model parameters in the genetic algorithm optimisation are the same as used in Section 9.6.1. A set of lower cost solutions have been found for Loveday irrigation network. The optimal solution (see Appendix C for all the solutions) has been compared, as shown in Table 9-18, with the least cost solution fmGA7 using class I pipes. The solutions are evaluated by using the cost of both class I and class II pipes. It shows that the optimal solution corresponds to a lower cost than that of the solution evaluated by using the cost of the other class pipes. For example, the optimal solution of class I pipes is less costly than that evaluated by using the cost of class II pipes. This indicates that the stage-one optimisation model has been effective at achieving the optimal solution according to the cost information being used in the model.

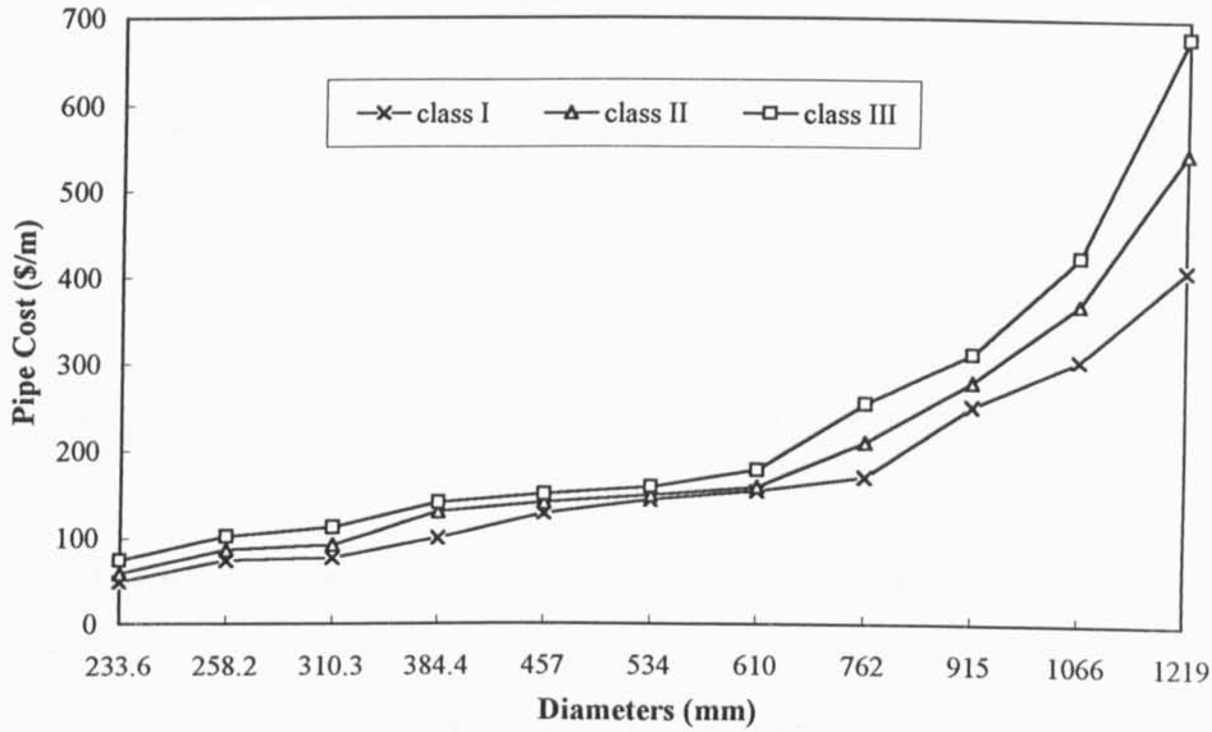


Figure 9-5 Cost Structure of Class I, II and III Pipes

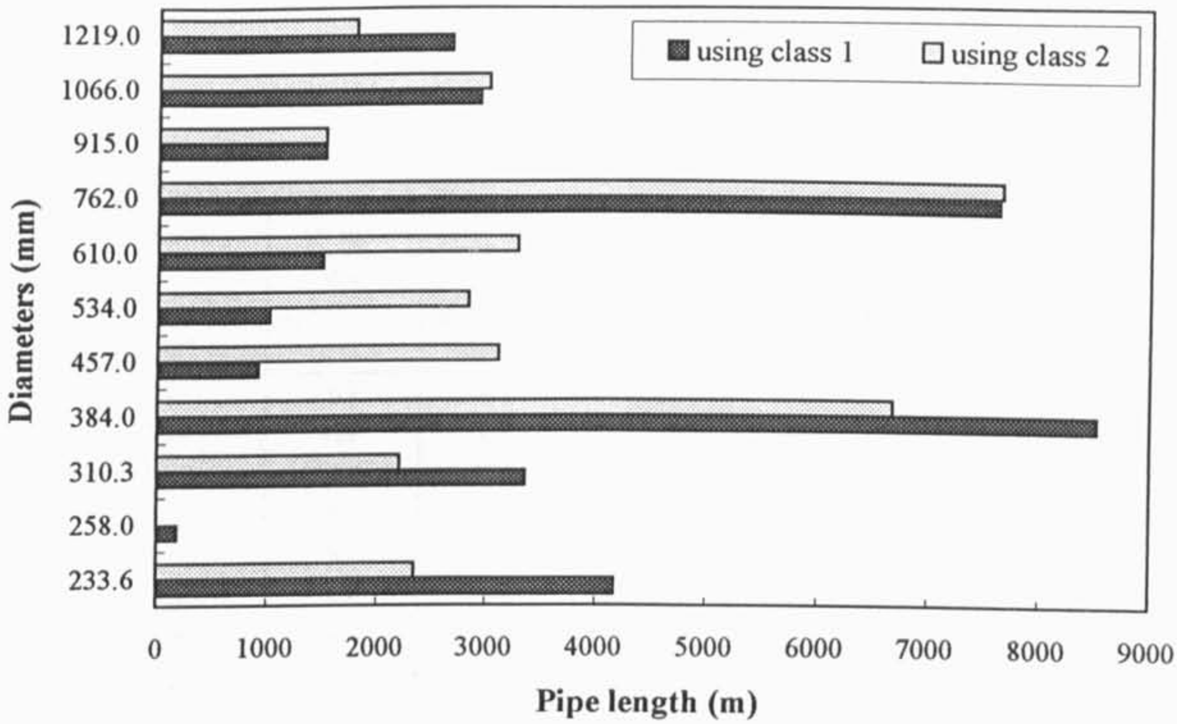


Figure 9-6 Pipe Length of Optimal Diameters of Stage-one Solutions Using Class I and II Pipe Cost

The stage-one optimal pipe sizes of both class I and class II pipes are classified by the plot of the pipe length of a individual pipe diameter as shown in Figure 9-6. It demonstrates that the stage-one optimisation of both class I and II pipes has chosen similar pipe sizes for the large pipes of diameters of 762.0, 915.0, 1006 and 1219 millimetres. The class I pipes favour small uPVC pipes of diameters from 233.6 to 384.4 millimetres, while the class II pipes favour reinforced concrete pipes of diameters of 457.0, 534.0 and 610.0 millimetres. The optimal solution of class II pipes is fixed for optimal selection of pipe thicknesses (pipe classes) and the size of the surge tank for Loveday network under water hammer loading. The results from the stage-two optimisation are compared with the optimal solutions presented in Section 9.6.2.

Table 9-19 Comparison of Stage-two Optimisation Based on Stage-one Solutions Using Class I and II Pipe Cost

Pipe	Based on stage-one optimisation by using class I pipe cost			Based on stage-one optimisation by using class II pipe cost		
	Diameters	no tank	with tank	Diameters	no tank	with tank
ID	(mm)	wall thickness (mm)	wall thickness (mm)	(mm)	wall thickness (mm)	wall thickness (mm)
1	1219	63	63	1219	63	63
2	1219	63	63	1219	63	63
3	310.3	12.9	12.9	310.3	12.9	12.9
4	1219	63	63	1219	63	63
5	762	51	51	762	51	51
6	534	41	41	384.4	15.9	15.9
7	1066	63	63	1066	63	63
8	762	51	51	762	51	51
9	762	51	51	762	51	51
10	762	51	51	610	44	44
11	762	51	51	762	51	51
12*	**610**	**57**	**57**	**762**	**63**	**63**
13	**610**	**57**	**57**	**762**	**63**	**63**
14	233.6	9.7	9.7	233.6	9.7	9.7
15	534	41	41	457	38	38
16	1219	63	63	1066	63	63
17	1066	63	63	1066	63	63
18	310.3	12.9	12.9	384.4	15.9	15.9
19	1066	63	63	1066	63	63
20	1066	63	63	1066	63	63
21	310.3	12.9	12.9	310.3	12.9	12.9
22	1066	63	63	1066	63	63
23	1066	63	63	1066	63	63
24	915	63	63	915	63	63
25	762	51	51	762	51	51
26	1066	63	63	762	51	51
27	915	63	63	915	63	63
28	1066	63	63	762	51	51
29	**762**	**63**	**63**	**762**	**63**	**63**
30	762	51	51	**762**	**63**	**63**
31	762	76	76	**534**	**61**	**61**
32	457	54	54	**534**	**61**	**61**
33	384.4	15.9	15.9	384.4	15.9	15.9
34	384.4	15.9	15.9	384.4	15.9	15.9
35	384.4	15.9	15.9	**610**	**57**	**57**

Pipe ID	Based on stage-one optimisation by using class I pipe cost			Based on stage-one optimisation by using class II pipe cost		
	Diameters (mm)	no tank wall thickness (mm)	with tank wall thickness (mm)	Diameters (mm)	no tank wall thickness (mm)	with tank wall thickness (mm)
36	**762**	**76**	**63**	233.6	9.7	9.7
37	384.4	15.9	15.9	233.6	9.7	9.7
38	258.2	10.7	10.7	310.3	12.9	12.9
39	**762**	**76**	**76**	457	38	38
40	233.6	9.7	9.7	**610**	**57**	**57**
41	384.4	15.9	15.9	**610**	**57**	**57**
42	384.4	15.9	15.9	233.6	9.7	9.7
43	233.6	9.7	9.7	**610**	**57**	**57**
44	233.6	9.7	9.7	534	41	41
45	384.4	15.9	15.9	384.4	15.9	15.9
46	384.4	15.9	15.9	762	51	51
47	384.4	15.9	15.9	384.4	15.9	15.9
48	384.4	15.9	15.9	384.4	15.9	15.9
49	310.3	12.9	12.9	384.4	15.9	15.9
50	384.4	15.9	15.9	384.4	15.9	15.9
51	310.3	12.9	12.9	310.3	12.9	12.9
52	384.4	15.9	15.9	457	38	38
53	310.3	12.9	12.9	310.3	12.9	12.9
54	384.4	15.9	15.9	384.4	15.9	15.9
Pipe cost		5,737,509	5,700,384		5,701,021	5,701,021
Surge tank diam. (m)		-	1.0		-	1.0
Surge tank height (m)		-	18.0		-	18.2
Surge tank cost		-	24,090		-	24,293
Total cost		5,737,509	5,724,474		5,701,021	5,725,314
Minimum transient head residual (m)		0.23	0.26		0.6	0.6

*pipes chosen higher class than the minimum class are in bold.

The stage-two optimal selection of pipe classes and surge tank sizes based on the stage-one optimal diameters of class II pipes has been carried out for the case without a surge tank and the case with surge tank at node 23. The optimal pipe thicknesses and/or surge tank sizes are tabulated in Table 9-19 and compared with the optimal transient design solutions in Section 9.6.2. It shows that the stage-two optimisation model based on stage-one optimal diameters of class II pipes has chosen the same optimal pipe thicknesses for both the case without a surge tank and the case with a surge tank in the system. A minimum diameter of 1.0 metre has been chosen for the surge tank, which is associated with a cost of \$24,293. Thus, the optimal transient design solution for the case with a surge tank in the system is more costly than the optimal transient solution without a surge tank. However, the cost of the optimal transient design solution based on the optimal diameter of class II pipes is similar to that based on the optimal diameters of class I pipes. This implies that the optimal transient solution without a surge tank, based on the optimal diameters of class II pipes, is economically and hydraulically equivalent to the optimal transient design solution with a surge tank based on the optimal diameters of class I pipes. This is because the stage-one optimisation model using class II pipe costs has chosen more reinforced concrete pipes of diameters of 457.0, 534.0 and 610.0 mm, as shown in Figure 9-6, than that using class I pipe costs. It is the large reinforced concrete pipes that provide a similar transient protection capacity to that provided by the surge tank of the optimal transient design solution based on the optimal diameters of class I pipes. This indicates that the two-stage optimisation approach for transient design of water distribution systems is effective at finding the least cost solution with a satisfied security margin of water hammer protection, and also that the cost of the optimal transient design solution is not really sensitive to the pipe class being fixed in stage-one.

9.8 Summary

A model of optimisation of transient design of water distribution systems has been formulated in this Chapter and implemented by incorporating the water hammer simulation model HAMMER into the computer program fmGANET developed in Chapter 6. The boundary GA search strategy based on self-adaptive penalty has also been incorporated into the integrated computer program fmGAHAM for optimal transient design. This approach has been applied to two case studies to investigate the effectiveness and efficiency of the methodology.

The first example is a simple hypothetical pipeline system. A complete enumeration of the transient design alternatives has been carried out. It provides a comparison basis for verifying the optimal solution by fmGAHAM. The results obtained shows that fmGAHAM is effective at finding the optimal solution. The second example is a real water distribution system in South Australia. It is a low head irrigation system and consists of 54 pipes to be sized. The optimal transient design has been carried out in two stages. In the first stage, only the pipe diameters were optimised by using a fixed thickness (or a class) pipes under steady state demand loadings. A set of lower cost solutions have been found for the system. In the second stage, the pipe class and the surge tank have been optimised by using the optimal diameter given in stage one. Although the total cost of the optimal transient designs are higher than that without considering water hammer events, the transient design provides the safety guarantee for the pressure surge protection. The optimal transient design including the pipe classes and the surge tank is not only able to demonstrate the economic benefit but also to ensure the safety of the transient pressure surge that is generated by operation of the water distribution system.

The optimisation of the transient designs based on optimal diameters of class II pipes has also been carried out for Loveday irrigation network and compared with the optimal transient designs based on the optimal diameters of class I pipes. It shows that the model at stage one is able to produce the optimal design solutions according to the cost information being used in the model. The total cost of the optimal transient design produced at stage two optimisation does not appear to be sensitive to the pipe class being fixed in stage-one optimisation. Thus it is concluded that the approach for the optimisation of transient design of water distribution system is effective and efficient at finding the optimal solution for design of water distribution systems including water hammer loadings.

10. CONCLUSIONS AND RECOMMENDATIONS

10.1 Introduction

This dissertation was motivated to achieve two goals:

- To develop an efficient and effective optimisation technique for design and rehabilitation of water distribution systems.
- To develop a generic methodology for optimal design of water distribution systems considering water hammer loadings.

The goals have been achieved by the accomplishments outlined in this thesis that have been carried out as follows:

- The development of a generalised genotype representation and mapping scheme for genetic algorithm optimisation of water distribution systems.
- The development of a discrete pipe model for optimisation of design and/or rehabilitation of water distribution systems.
- The analysis of a continuous pipe formulation and development of a split pipe model for optimisation of water distribution systems.
- The implementation of both the discrete pipe and split pipe models by integrating the messy GA with the hydraulic network solver. The integrated approach has been applied to a number of previously studied networks.
- The comparison of GA paradigms for optimisation of water distribution systems.
- The development of the fast messy GA for optimisation of water distribution system and its application to a large-scale real network.
- The development of a boundary search of genetic algorithm optimisation by coevolution of self-adaptive penalty functions.

- The development and analysis of water hammer simulation of water distribution systems.
- The development of an approach for optimal transient design of water distribution systems under steady state and water hammer loading.

The conclusions drawn from this work and recommendations made for the future work will be discussed in the following sections.

10.2 Conclusions

The conclusions that can be drawn from this work are:

1. **The messy genetic algorithm is more efficient and effective at optimisation of water distribution system than the standard GA and/or improved GA.**

A generalised formulation for genetic algorithm optimisation of water distribution systems has been developed in Chapter 3. It includes (1) a discrete pipe model; (2) a genotype and phenotype representation and (3) a scheme for mapping a genotype (a string) to a phenotype (a solution). The optimisation problem was solved by using the original messy GA. The messy GA has been integrated with the hydraulic network solver EPANET. The integrated program mGANET has been tested on previously studied examples, the two reservoir network and the New York city tunnels problem. The messy GA uses variable-length strings, thresholding selection and messy operators of cut and splice. It starts by enumerating a certain order of building blocks and is followed by a primordial phase and a juxtapositional phase. In the primordial phase, tournament thresholding selection occurs only to allow highly fit strings to be enriched in a population. The strings are evolved in the juxtapositional phase by performing the messy genetic operations of cut and splice. The results have shown that the messy GA approach to optimal sizing and rehabilitation of water distribution networks is more efficient than the standard GA and/or the improved

GA approach. The approach will be able to provide the decision-maker a set of lower cost solutions for the design of the water distribution systems.

2. **The genetic algorithm split pipe model is able to produce lower cost solutions than GA discrete pipe optimisation for design and rehabilitation of water distribution systems.**

A split pipe formulation for the genetic algorithm optimisation of water distribution systems has been developed in Chapter 4. An analysis showing a comparison of the continuous pipe formulation and the split pipe formulation has been presented. It indicates that there is no guarantee, in general, that the continuous pipe size model using a fitted convex cost function will reach the optimal split pipe solution. The split pipe optimisation model, allowing the split pipe sizes to be used in the optimisation procedure, has been formulated to search for the optimal split pipe solution. This model has been implemented in a genetic algorithm formulation and tested on two previously studied examples of optimisation of water supply networks. The results have shown that the genetic algorithm split pipe model for optimal sizing and rehabilitation of water distribution networks is able to give lower cost solutions than genetic algorithm discrete pipe optimisation.

3. **The messy genetic algorithm has been shown to be particularly suitable to solve the problems of optimal rehabilitation of water distribution systems.**

The model developed for optimal design of water distribution systems in Chapter 3 has been extended to deal with optimal rehabilitation of the systems as discussed in Chapter 5. The optimal rehabilitation model has been formulated to select a rehabilitation action for each pipe from a set of possible rehabilitation actions such as duplicating, cleaning, relining, replacing or just leaving a pipe as it is. The pipe size associated with a rehabilitation action is chosen from a list of commercially available discrete pipe sizes, and a pumping capacity is employed so that the water demand and the minimum allowable hydraulic pressures at all the nodes are satisfied while the total cost of the rehabilitation is minimised. The total cost considered includes both the pipe rehabilitation cost and the pumping cost. To minimise the total cost, an integrated program mGANET, which has coupled the messy genetic algorithm and a hydraulic network solver EPANET, has been employed to search for the optimal rehabilitation strategy. The results of the case study have shown that the messy GA is efficient at searching for the optimal solution. It provides a decision-maker with a set of least cost solutions, which then can be judged by using other non-quantifiable criteria.

4. The fast messy genetic algorithm is the most efficient genetic algorithm at optimisation of water distribution system among the GA paradigms.

The standard genetic algorithm has been compared with the messy genetic algorithm in Chapter 6. The standard GA defines the relations and classes implicitly by using a fixed-length representation. It combines the relation space, the class space and the sample space all together, thus a poor and noisy decision process occurs. Increasing the tournament selection pressure can improve the search efficiency of the standard GA, but too much pressure may leads the search to a local optimal.

Messy GAs emphasise searching for appropriate relations. The original messy GA uses a competitive template and explicit enumeration of good classes—building-blocks—to ensure correct decision making. However, using an initialisation procedure where building-blocks are explicitly enumerated essentially limits the messy GA so that it cannot be applied to a highly dimensional problem. The probabilistically complete initialisation and the building-block filtering process are introduced into the fast messy GA to detect better classes from better relations. An empirical comparison study of the messy GAs for optimisation of pipeline networks has been carried out and shows that the fast messy GA is the most efficient algorithm among the genetic-based search paradigms. It eliminates the major bottleneck of the original messy GA—the explicitly enumerative initialisation and thus provides a promising optimisation algorithm for solving highly dimensional discrete optimisation problems. The fast messy GA has been implemented and integrated with hydraulic network solver EPANET. The integrated approach has been applied to the optimal rehabilitation of a real water system in Morocco. This application has demonstrated that the fast messy GA is very efficient at solving large-scale optimisation problems.

5. The boundary search approach by co-evolution of self-adaptive penalty improves the efficiency and effectiveness of genetic algorithm optimisation.

A strategy of boundary search of GA optimisation has been developed by co-evolution of the penalty factor and self-adaptation of the penalty factor, and also applied to optimisation of water distribution systems. The optimal solution of design and rehabilitation of water networks has been observed to be achieved at the boundary of the feasible and infeasible regions of the search space. It is a boundary optimisation problem. A heuristic rule is specifically proposed for adapting the penalty factor range in such a way that the genotype population is forced towards the boundary of the feasible and infeasible regions. Optimisation of New York city tunnels problem has been chosen to demonstrate the application of this boundary GA search strategy. The results obtained by using the boundary GA search strategy have been carefully analysed and compared with the conventional penalty GA optimisation approach. It has been found that the conventional approach is quite sensitive to the penalty factor. Too large a penalty value may preserve the feasibility of the genotypes and the search normally converges to an internal point of the feasible region, which is not real optimal solution. Too small a penalty value forces the GA to select an alternative solution outside the feasible region, which eventually leads the search to converge to an infeasible solution. The boundary search strategy has been shown to be effective and efficient at adapting the feasibility of the GA population within a large range of the penalty factors. It automatically adjusts the penalty range and co-evolves the penalty factor during the GA optimisation process. The boundary search strategy is not only generally applicable to the optimisation of water distribution system, but also to other boundary optimisation problems, which are often found in engineering design. It has been successfully applied to the optimisation of design of water distribution system including water hammer loadings.

6. Consideration of the transient loading is essential for design of pipeline networks.

A computer model for the water hammer simulation of water distribution systems has been developed and applied to investigation of transient behaviour of low head irrigation systems. The transient model, based on the method of characteristics, has been integrated with the hydraulic solver EPANET. The integrated model HAMMER/EPANET is able to perform the simulation of the steady and unsteady flow in the pipeline network with reservoirs, nodes, junctions, in-line valves, outlet valves and surge tanks. A comprehensive analysis of the transient events has been carried out for the Loveday irrigation pipe system in South Australia. The simulation results obtained show that water hammer events generate a severe transient pressure not only in a simple series pipeline, but also in a complicated networks. An approach for evaluation of the transient events has been developed and is applied to the evaluation of the transient pressure surge by the water hammer events. This provides engineers with a method for identifying the critical water hammer events for the cost effective transient design of water distribution systems.

7. An approach of the two-stage messy genetic algorithm optimisation that has been developed in Chapter 9 is effective at searching for the optimal solution of design of water distribution systems including consideration of water hammer.

A model for the optimisation of transient design of water distribution systems has been formulated and implemented by incorporating the water hammer simulation model into the computer program fmGANET developed in Chapter 6. The boundary GA search strategy based on self-adaptive penalty has also been incorporated into the computer program fmGAHAM for the optimal transient design.

This approach has been applied to two case studies of optimal transient design of water distribution systems. The optimal solution of the first example, a simple hypothetical pipeline system, has been verified by a complete enumeration of the transient design alternatives. The results obtained show that the approach is effective at finding the optimal solution. For a complicated network, it is recommended that the optimisation of transient design of water distribution systems be carried out in two stages. In the first stage, only the pipe diameters are optimised by using the cost information of a fixed pipe class under steady state demand loadings. A set of optimal diameters are found for the system. In the second stage, the diameters are locked in, the pipe classes and water hammer protection devices (e.g. a surge tank) are then optimised.

Optimisation of transient design of Loveday irrigation network, a real water system in South Australia, has been carried out by applying the two-stage approach. It is a low head irrigation system and consists of 54 pipes to be sized. A set of optimal diameters of class I pipes have been found for the system under steady state demand loadings. In the second stage, the pipe class and the surge tank have been optimised by using the optimal diameters found in stage one. Although the total cost of the optimal transient designs based on the optimal diameters of class I pipes are higher than that without considering water hammer events, the optimal transient design provides the safety guarantee for the pressure surge protection. Thus it is concluded that the optimal transient design of water distribution systems is not only able to demonstrate the economic benefit but also to ensure the safety of the transient pressure surge that is generated by operation of the water distribution systems.

The optimisation of the transient design of Loveday irrigation network, based on optimal diameters of class II pipes, has also been carried out. The results have been compared with the optimal transient designs based on the optimal diameters of class I pipes. It shows that the model is able to produce the optimal transient design solutions. The cost of the optimal transient design is observed not being sensitive to the pipe class being fixed in stage-one. The two-stage optimisation process is likely to lead to a suboptimal design solution, although it is shown that the approach is effective and efficient at finding the near optimal or cost effective solutions for the design of water distribution systems including consideration of water hammer loadings.

10.3 Recommendations

To further improve the efficiency and accuracy of optimisation of water distribution systems, a number of research avenues for future consideration are suggested as follows:

- **Incorporation of Pump operation (scheduling) into optimal design and/or rehabilitation of water distribution systems.** Chapter 5 shows that the cost of the rehabilitation is dominated by the pump energy cost, which is almost 90% of the total cost. This implies that the optimal rehabilitation of water distribution system must consider not only pump useful horse power but also the pump operation (scheduling).

The consideration of pump operation will provide more accurate information to evaluate the rehabilitation strategies in an optimal rehabilitation model of the water distribution systems in the future.

- **Transient sensitivity analysis.** It is the simulation of transient loadings that takes the most of computation time to run the optimal transient design program. The sensitivity study will identify what parameters (diameters, pipe wall thickness, materials or surge protection devices) are the most sensitive to the maximum transient pressure in pipes. This will lead to a simplification of the model for optimal transient design and/or the model for transient simulation of water distribution systems.

- **Approximate transient simulation.** A simplified model for simulation of transient events could be developed to improve the efficiency of optimisation of transient design. The approximate simulation model can be formulated by just considering the most sensitive parameters.

- **Optimal allocation of transient surge protection devices.** The model developed in this research could be improved to taking into account selecting the optimal location of a water hammer protection device (eg. a surge tank). This could further increase the economic benefit of the optimal transient design and also provide better protection of the transient pressure surge within pipeline systems.

Work on these suggestions will be fruitful in directly improving the optimisation of water distribution systems or indirectly exploring many other ways to solve real-world optimisation problems.

11. BIBLIOGRAPHY

Abbott, M. B. (1992). *Hydroinformatics: information technology and the aquatic environment.* Avebury Technical.

Alperovits, E. and Shamir, U. (1977). "Design of water distribution systems." *Water Resources Research.* Vol. 13, No. 6, 885-900.

Ahn, T. (1993). "Optimal design of municipal and irrigation water distribution systems." *Ph.D Dissertation,* submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Arfaie, M. and Anderson, A. (1991). "Implicit finite difference for unsteady pipe flow." *Mathematical Engineering in Industry,* Vol. 3, No. 2, 133-151.

Arora, J. S. & Huang M. W. (1994). "Methods for optimization of nonlinear problems with discrete variables: a review." *Structural Optimization* 8, 69-85.

Babovic, V., Wu, Z. Y. and Larsen, L. C. (1994). "Calibrating Hydrodynamic Models by Means of Simulated Evolution." *in Proc. Hydroinformatics'94,* Delft, the Netherlands, 197-200.

Back, T., Hoffmeister, F. & Schwefel, H.-P. (1991). "A survey of evolution strategies." In R. K. Belew & L. B. Booker (Eds.), *Proc. of the Fourth International Conference on Genetic Algorithms* (pp2-9). San Mateo, CA: Morgan Kaufmann.

Bean, J. C. & Hadj-Alounae, A. B. (1992). " A dual genetic algorithm for bounded integer programs." *Technical Report,* TR 92-53, Ann Arbor, MI: University of Michigan, Department of Industrial and Operations Engineering.

Bhave, P.R. (1985). "Optimal expansion of water distribution systems," *J. Environmental Engineering*, ASCE, 111(2), 177-197.

Bhave, P. R. and Sonak, V. V. (1992). " A critical study of the linear programming gradient method for optimal design of water supply networks." *Water Resources. Research.* 28(6), 1577-1584.

Box, M. J. (1965). " A new method of constrained optimization and a comparison with other methods." *J. Computer,* Vol. 8, pp. 42.

Cembrowicz, R. C., and Krauter, G. E. (1977). "Optimization of urban and regional water supply systems." *Conf. Proc. of system approach for development*, IFAC, Cairo, Arab Republic of Egypt.

Chaudhry, M.H. (1987). *Applied Hydraulic Transients.* Litton Educational Publishing Inc., van Nostrand Reinhold Co.

Collin, M., Cooper, L., Helgason, R., Kennington, J. and LeBlanc, L. (1978). "Solving the pipe network analysis problem using optimization techniques." *Management Science,* The Institute of Management Science, 24, 747-760.

Cross, H. (1936). "Analysis of flow in networks of conduits or conductors." Bulletin No. 286, University ofIllinois Eng. Expr. Station, Urbana, Illinois.

Dandy, G. C., Simpson, A. R. and Murphy, L. J. (1993). "A review of pipe network optimization techniques," *Proc. of 2nd Australian Conf. on Computing for the Water Industry Today and Tomorrow,* ACADS, Melbourne, Australia

Dandy, G. C., Simpson, A. R. and Murphy, L. J. (1996). "An improved genetic algorithm for pipe network optimization," *Water Resources Research,* Vol. 32, No. 2, 449-458.

Deb, K. and Goldberg, D.E. (1991) "mGA in C: A Messy Genetic Algorithm in C," *IliGAL Report No. 91008*,Illinois Genetic Algorithms Laboratory, University ofIllinois at Urbana-Champaign, Urbana, IL 61801, USA.

Duan, N., Mays, L. W. and Lansey, K. E. (1990). "Optimal reliability-based design of pumping and distribution systems." *J. Hydr. Engrg.*, ASCE, 116(2), 249-268

Eiger, G., Shamir, U. and Ben-Tal, A. (1994). "Optimal design of water distribution networks." *Water Resour Res.*, 30(9), 2637-2646.

Fiacco, A. V. and McCormick, G. P. (1968). *Nonlinear programming: sequential unconstrained minimisation*, Wiley, New York.

Fujiwara, O., Jenchaimahakoon and Edirisinghe N. C. P., (1987). "A modified linear programming gradient method for optimal design of looped water distribution networks, *Water Resour. Res.*, 23(6), 977-982.

Fujiwara, Q. and Dey, D. (1987). Two Adjacent Pipe Diameters at the Optimal Solution in the Water Distribution Network Models. *Water Resources Research.* Vol. 23, No. 8, 1457-1460.

Fujiwara, O. and Khang, D. B. (1990). "A Two-Phase Decomposition Method for Optimal Design of Looped Water Distribution Networks," *Water Resources Research.* Vol. 26, No. 4, 539-549.

Gessler, J. (1982). "Optimization of pipe networks," *Proc., International Symposium on Urban Hydrology, Hydraulics and Sediment Control,* University of Kentucky, Lexington, Kentucky, 165-171.

Gessler, J. (1985)." Pipe Network Optimization by Enumeration," *Proc., Computer Application for Water Resources.*, ASCE, Buffalo, N.Y., 572-581.

Glover, F. (1977). "Heuristics for integer programming using surrogate constraints." *Decision Sciences,* 8(1), 156-166.

Glover, F. & Kochenberger, G. (1995). "Critical event tabu search for multidimensional knapsack problems." In *Proc. of the International Conference on Metahuristics of Optimization* (pp113-133). Dordrecht, the Netherlands: Kluwer Publishing.

Goldberg, D. E. and Kuo, C. H. (1987). "Genetic algorithm in pipeline optimization", *J. Computing in Civ. Engrg.,* ASCE, 1(2), 128-141.

Goldberg, D.E. (1987). "Simple genetic algorithms and minimal, deceptive problem," In Davis L. (Ed.), *Genetic Algorithms and Simulated Annealing,* (pp. 74-88). San Mateo, CA: Morgan Kaufmann.

Goldberg, D.E. (1989). *Genetic algorithms in search, optimization and machine learning,* Addison-Wesley Publishing Company, Inc., 412pp.

Goldberg, D.E., Korb, B., & Deb, K. (1989). "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Systems, 3,* 493-530.

Goldberg, D. E., Deb, K. & Korb, K. (1990) "Messy genetic algorithms revisited: studies in mixed size and scale," *Complex Systems,* 4, 415-444.

Goldberg, D. E., Deb, K., Kargupta, H., & Harik G. (1993). "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms," *IlliGAL Report No. 93004,*Illinois Genetic Algorithms Laboratory, University ofIllinois at Urbana-Champaign, Urbana, IL 61801, USA.

Goulter, I.C. (1987). "Current and future use of systems analysis in water distribution network design." *Civ. Engng Syst.,* Vol. 4, 174-184.

Goulter, I.C. (1992) "Systems Analysis in Water-Distribution Network Design: From Theory to Practice," *Journal of Water Resources Planning and Management*, ASCE , Vol. 118, No. 3, 238-348

Halhal, D, Walters, G. A., Ouazar, D. and Savic, D. (1997) "Water network rehabilitation with structured messy genetic algorithm." *Journal of Water Resources Planning and Management*, ASCE, Vol. 123, No. 3, 137-146.

Haug, E. J. and Arora, J. S. (1979). *Applied optimal design: mechanical and structural systems*, Wiley, New York.

Holland, J.H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, Michigan.

Homaifar, A., Lai, S. H.-Y. & Qi, X. (1994). "Constrained optimization via genetic algorithms." *Simulations*, 62(4), 242-254.

Joins, J. & Houck, C. (1994). "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs." In Michalewicz Z, Schaffer J. D. Schwefel, H.-P., Fogel D. B. & Kitano, H. (Eds.), *Proc. of the first IEEE International Conference on Evolutionary Computing* (pp579-584). Piscataway, NJ: IEEE Press.

Kally, E. (1971). "Pipeline planning by dynamic computer programming", *J. of Amer. Water Works Association.*, March, 114-118.

Karney, B. and McInnis, D. (1992). "Efficient calculation of transient flow in simple pipe networks." *J. Hydr. Enrg*, ASCE, 118(7), 1014-1030.

Karney, B. and McInnis, D. (1990). "Transient analysis of water distribution systems." *J. AWWA*, 82(7), 62-70.

Kargupta, H. (1995). "SEARCH, Polynomial Complexity, and the Fast Messy Genetic Algorithm." *Ph.D Thesis,* University ofIllinois at Urbana-Champaign.

Kessler, A. and U. Shamir, (1989). "Analysis of the linear gradient method for optimal design of water supply networks." *Water Resour. Res.*, 25(7), 1469-1480.

Kim, J. H. and Mays, L. W. (1994) "Optimal rehabilitation model for water-distribution systems." *Journal of Water Resource Planning and Management*, ASCE, Vol.120, No.5, 982-1000.

Lansey, K. E. and Mays, L.W. (1989a). "Optimization Model for Water Distribution System Design". *J. Hydr. Engrg.*, ASCE, Vol. 115, No. 10, 1401-1418.

Lansey, K. E. and Mays, L.W. (1989b). "Optimization model for design of water distribution systems." *Reliability analysis of water distribution systems,* L. R. Mays, ed., ASCE, New York, N. Y.

Lansey, K. E. and Mays, L.W. (1989c). "Simulation model for analysis of water distribution systems." *Reliability analysis of water distribution systems,* L. R. Mays, ed., ASCE, New York, N. Y.

Loganathan, G. V. Greene, J.J. and Ahn, T.J. (1995) "Design heuristic for global minimum cost water-distribution systems," *Journal of Water Resources Planning and Management,* ASCE, Vol. 121, No. 2, 182-192

McInnis, D. and Karney, B. W. (1995). "Transients in distribution networks: Field test and demand models." *Journal of Hydraulic Engineering,* ASCE, Vol. 121, No. 3, 218-231.

Michalewicz, Z. and Schoenauer, M. (1996). "Evolutionary algorithms for constrained parameter optimization problems." *Evolutionary Computation,* 4(1), 1-32, MIT Press.

Michalewicz, Z. (1995). "Genetic algorithms, numerical optimization and constraints." In Esheelman L. J. (Ed.), *Proc. of the Sixth International Conference on Genetic Algorithms* (pp151-158). San Mateo, CA: Morgan Kaufmann.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs.*, 3rd ed. New York, Springer-Verlag.

Michalewicz, Z and Attia, N. (1994). "Evolutionary optimization of constrained problems." In Sebald A. V. & Fogel L. J. (Eds.), *Proc. of the Third Annual Conference on Evolutionary Programming* (pp98-108). River Edge, NJ: World Scientific.

Michalewicz, Z., Logan, T. & Swaminathan, S. (1994). " Evolutionary operators for continuous convex parameter space. In *Proc. of the Third Annual Conference on Evolutionary Programming* (pp84-97), River Edge, NJ: World Scientific.

Miachlewicz, Z. & Nazhiyath, G. (1995). "Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints." In Fogel D. B. (Ed.), *Proc. of the Second International Conference on Evolutionary Computation* (pp. 647-651). Piscataway, NJ: IEEE Press.

Murphy, L.J. and Simpson, A.R. (1992). "Pipe optimization using genetic algorithms," *Research Report No. R 93,* Dept. of Civil and Envir. Eng., The University of Adelaide, Australia.

Murphy, L.J., Simpson, A.R. and Dandy, G.C.(1993). "Pipe network optimization using an improved genetic algorithm," *Research Report No. R 109,* Dept. of Civil and Envir. Eng., The University of Adelaide, Australia.

Murphy, L.J., Dandy, G.C. and Simpson, A.R.(1994). "Optimal design and operation of pumped water distribution systems," *Proc., Conf. on Hydraulics in Civil Engineering,* Institution of Engineers, Australia, Brisbane, Australia

Morgan, D. R. and Goulter, I.C. (1985). "Water Distribution Design with Multiple Demands," *Proc., Computer Application for Water Resources.*, ASCE, Buffalo, N.Y., 582-590.

Ormsbee, L. and Contractor, D. (1981). "Optimization of hydraulic networks", *International Symposium on Urban Hydrology, Hydraulics and Sediment Control,* Lexington, KY, 255-281.

Oron, G. and Karmeli, "Procedure for economic evaluation of water networks parameters", *Water Res. Bulletin,* Vol. 15, No. 4, 1050-1060.

Parmee, I., & Purchase, G. (1994). "The development of directed genetic search technique for heavily constrained design spaces." In *Proc. of the Conference on Adaptive Computing in Engineering Design and Control* (pp. 97-102). Plymouth, UK: University of Plymouth.

Powell, D. & Skolnick, M. M. (1993). "Using genetic algorithm in engineering design optimization with nonlinear constraints. In Forrest S. (ed.), *Proc. of the Fifth International Conference on Genetic Algorithms* (pp. 424-430).

Quindry, G., Brill, E.D., Liebman, J. C. and Robinson A. R. (1979). Comment on "Design of optimal water distribution systems" by E. Alperovits and U. Shamir, *Water Resour. Res.*, 15(6), 1651-1654.

Quindry, G. E., Brill, E. D., Liebman J.C., and Robinson A. R., (1979). Comment on "Design of optimal water distribution system" by Alperovits E. and Shamir U., *Water Resour. Res.*, 15(6), 1651-1654.

Quindry, G. E., Brill, E.D., and Liebman, J.C. (1981). "Optimization of looped water distribution systems," *J. Environmental Engineering Division,* ASCE, 107(EE4), 665-679.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimuerung Technischer Systeme nach Prinziplen der Biolgishen Evolution.* Frommann-Holzboog, Stuttgart, Germany (in german), being translated by Schwefel, H. P. (1981) and included in *Numerical Optimization of Computer Models,* John Wiley & Son.

Rossman, L.A. (1994) "EPANET user's manual," Risk Reduction Engineering Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, Cincinnati, OH 45268.

Rowell, W. and Barnes, J. "Obtaining layout of water distribution systems", *J. of Hydraul. Div.,* ASCE, Vol. 108, No. HY1, 137-148.

Savic, D.A. and Walters, G. A.(1997). "Genetic Algorithms for the Robust, Least-cost Design of Water Distribution Networks," *J. Water Resour. Plng. and Mgnmt.,* ASCE, 123(2), 67-77.

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, D. (1985). "Multiple objective optimization with vector evaluated genetic algorithms." In Grefensttete J. J. (Ed.), *Proc. of the First International Conference on Genetic Algorithms.* New York: Laurence Erlbaum Associates.

Schoenauer, M. & Xanthakis S. (1993). "Constrained GA optimization." In Forrest S. (Ed.), *Proc. of the Fifth International Conference on Genetic Algorithms* (pp. 573-580). San Mateo, CA: Morgan Kaufmann.

Simpson, A. R., McPheat J. and Gibbard M. J. (1992). "Development of a water hammer computer model for integrated dynamic modelling of hydro-electric power plant." unpublished report, The University of Adelaide.

Simpson, A. R., Murphy, L. J. and Dandy, G.C.(1993). "Pipe Network Optimization Using Genetic Algorithms," *Proc., ASCE, Water Resources Planning and Management Division Specialty Conf.,* ASCE, Seattle, Washington, May.

Simpson, A. R., Dandy, G. C. and Murphy, L.J. (1994) "Genetic algorithms compared to other techniques for pipe optimization. " *Journal of Water Resource Planning and Management,* ASCE, Vol. 120 No. 4, 423-443

Simpson, A. R. and Goldberg, D. E. (1994). "Pipeline optimization via genetic algorithms: from theory to practice." *2nd Int'l. Conf. on Pipeline Systems,* Edinburgh, Scotland.

Su, Y. C., Mays, L.W., Duan, N., and Lansey, K.E. (1987). "Reliability-based optimization model for water distribution system." *J. Hydr. Engrg.,* ASCE, 114(12), 1539-1556.

Surry, P., Radiffe N. & Boyd I. (1995). "A multi-objective approach to constrained optimization of gas supply networks." In Fogarty T. (Ed.), *Proc. of the AISB-95 Workshop on Evolutionary Computing,* Vol. 993 (pp166-180). Berlin: Springer-Verlag.

Taher, S. A. and Labadie, J. W. (1996). "Optimal Design of Water-Distribution Networks with GIS". *Journal of Water Resource Planning and Management,* ASCE, Vol. 122 No. 4, 301-311.

Thierens, D. & Goldberg, D. (1993). "Mixing in Genetic Algorithms". In Forest S. (Ed.), *Proc. of 5th Int'l. Conf. on Genetic Algorithms.* San Mateo, CA, Morgan Kaufmann, 38-45.

Todini, E. and Pilati, S. (1988). "A gradient algorithm for the analysis of pipe networks." *Computer application in water supply (Proc. of an international conference held at Leicester Polytechnic in 1978).* B. Coulbeck and C. Orr, eds., Research Studies Press, Letchworth, Herfordshire, England, 1-20.

U.S. Army Corps of Engineers (1980), "Methodology for area wide planning studies," Engineer Technical letter No. 1110-2-502, Washington, D.C. 20314.

U.S. Army Corps of Engineers (1983), "Engineering and design evaluation of existing water distribution systems." Engineer Technical letter No. 1110-2-278, Washington, D.C. 20314.

Verwey, A. and Yu, J. H. (1993) "A space-compact high-order implicit scheme for water hammer simulation." *XXVth I.A.H.R. Congress,* Vol. 5 of Proceedings, Tokyo, Japan.

Warga, J. (1954). "Determination of the steady state flows and currents in a network." *Instrument Society of America.* Vol. 9, Pt. 5, Paper No. 54-43-4.

Walski, T. M. (1985). "State-of-the-Art Pipe Network Optimization". *Proc., Computer Application for Water Resources.,* ASCE, Buffalo, N.Y., 559-568.

Walski, T. M, Brill, E.D., Gessler, J., Goulter, I.C., Jeppson, R.M., Lansey, K., Lee, H.L., Liebman, J.C., Mays, L, Morgan, D.R., Ormsbee, L.(1987) "Battle of the Network Models: Epilogue". *Journal of Water Resource Planning and Management,* ASCE, Vol. 113, No. 2, 191-203

Walski, T.M. (1995). "Optimization and Pipe-Sizing Decisions". *Journal of Water Resource Planning and Management,* ASCE Vol. 121, No. 4, 340-343

Walters, G.A. (1988). "Optimal design of pipe networks: A Review." *Proc. 1st International conference on computer methods and water resources,* Vol. 2, Computational Hydraulics, Morocco, Edited by Ouazar D., Brebbia C.A. & Barthet H, Computational Mechanics Publications, Southampton, Boston.

Watters, G. Z. (1984). "Analysis and control of unsteady flow in pipelines." Butterworth Publishers, Stoneham, MA.

Wood, D. J. and Charles, A. M. (1972). "Hydraulic network analysis using linear theory." *J. Hydr. Div.*, ASCE, 98(7), 1157-1170.

Wood, D. J. (1980). *User's manual computer analysis of flow in pipe networks including extended simulations.* University of Kentucky, Lexington, Ky.

Wylie, E. B. and Streeter, V. L. (1993). "Fluid transients in systems." Prentice-Hall, Englewood Cliffs, N.J.

Yang, K. T. Liang, and Wu, I. (1975). "Design of conduit system with diverging branches", *J. of Hydraul. Div.*, ASCE, Vol. 101, No. HY1, 167-188.

Wu, Z. Y & Wang, Y. T. (1992). "Arch dam optimization design under strength fuzziness and fuzzy safety measure." *Proc. of Int. Conf. on Arch Dam,* Hehai University, Nanjing, China, pp 129 - 131.

Wu, Z. Y. (1994). "Automatic Model Calibration by Simulating Evolution." *M.Sc. Thesis* HH. 191, IHE, Delft, the Netherlands.

Wu, Z. Y. and Simpson, A. R. (1997). "An efficient genetic algorithm paradigm for discrete optimization of pipeline networks." MODSIM 97, *Proc. of International Congress on Modelling and Simulation,* Vol. 2, 983-988, Hobart, Tasmania, Australia.

Zhu, B. F., Li, Z. M. and Zhang, B. C. (1984). *Structural optimal design: theory and applications.* Hydro-electrical Press, Beijing, China, (in Chinese).

Appendix A OPTIMAL SOLUTIONS FOR REHABILITATION OF THE MOROCCAN NETWORK

A.1 Optimal Solution of fmGA5

Rehabilitation_for_old_pipe

ID	Action	Diam	C coeff.
1	duplication	400.0	130
2	leave	400.0	120
3	leave	200.0	90
4	leave	350.0	120
5	leave	100.0	90
6	duplication	80.0	130
7	leave	80.0	80
8	leave	80.0	90
9	leave	100.0	70
10	leave	100.0	90
11	leave	100.0	90
12	leave	300.0	120
13	leave	350.0	120
14	leave	100.0	90
15	leave	100.0	90
16	leave	80.0	90
17	leave	100.0	80
18	leave	100.0	80
19	leave	100.0	80
20	leave	100.0	70
21	duplication	80.0	130
22	duplication	80.0	130
23	leave	100.0	90
24	leave	100.0	90
25	leave	300.0	120
26	duplication	150.0	130
27	leave	100.0	90
28	leave	100.0	90
29	leave	80.0	100
30	leave	100.0	100
31	duplication	300.0	130
32	duplication	150.0	130
33	leave	80.0	90
34	leave	100.0	90
35	leave	200.0	120
36	leave	100.0	100
37	leave	200.0	120
38	leave	80.0	80
39	leave	100.0	80
40	leave	100.0	90
41	leave	100.0	70
42	duplication	300.0	130
43	leave	100.0	100
44	leave	80.0	100
45	leave	100.0	100
46	leave	100.0	100
47	leave	100.0	100
48	leave	80.0	100
49	leave	100.0	100
50	leave	100.0	100
51	leave	100.0	100
52	leave	80.0	100
53	leave	100.0	100
54	leave	100.0	100
55	leave	100.0	100
56	leave	100.0	100
57	leave	80.0	100
58	duplication	100.0	130
59	leave	80.0	100
60	leave	100.0	80
61	leave	100.0	80
62	leave	100.0	100

63	leave	80.0	100
64	leave	150.0	100
65	reline	150.0	120
66	leave	150.0	80
67	leave	150.0	90
68	leave	100.0	80
69	leave	80.0	100
70	leave	100.0	100
71	leave	100.0	100
72	leave	100.0	100
73	leave	150.0	100
74	leave	100.0	90
75	leave	150.0	90
76	leave	100.0	100
77	leave	150.0	100
78	leave	100.0	100
79	leave	100.0	100
80	leave	100.0	100
81	leave	80.0	100
82	leave	100.0	100
83	leave	100.0	100
84	leave	80.0	80
85	leave	300.0	70
86	leave	300.0	90
87	leave	100.0	100
88	leave	300.0	100
89	duplication	150.0	130
90	leave	400.0	100
91	leave	300.0	100
92	leave	150.0	100
93	leave	150.0	80
94	leave	80.0	80
95	leave	150.0	80
96	leave	150.0	70
97	duplication	150.0	130
98	reline	100.0	120
99	duplication	80.0	130
100	leave	80.0	120
101	leave	100.0	120
102	leave	100.0	110
103	leave	100.0	110
104	leave	60.0	90
105	leave	80.0	120
106	leave	100.0	120
107	leave	80.0	100
108	reline	100.0	120
109	duplication	80.0	130
110	leave	150.0	120
111	leave	100.0	100
112	leave	60.0	120
113	leave	100.0	90
114	leave	80.0	120
115	duplication	80.0	130
116	leave	60.0	120
117	leave	60.0	120
118	leave	60.0	120
119	leave	150.0	120
120	leave	150.0	100
121	leave	150.0	100
122	leave	60.0	120
123	leave	80.0	120
124	leave	80.0	120
125	leave	100.0	120
126	reline	100.0	120
127	leave	80.0	100
128	reline	100.0	120
129	duplication	200.0	130
130	leave	100.0	80
131	leave	60.0	120
132	leave	80.0	100
133	leave	100.0	120
134	leave	60.0	120
135	duplication	150.0	130
136	leave	60.0	120
137	leave	80.0	100
138	leave	400.0	100
139	reline	150.0	120
140	leave	100.0	100
141	leave	350.0	120
142	leave	100.0	100

143	reline	100.0	120
144	leave	100.0	100
145	leave	100.0	100
146	leave	100.0	120
147	leave	150.0	120
148	duplication	80.0	130
149	leave	250.0	120
150	duplication	80.0	130
151	leave	200.0	120
152	leave	80.0	120
153	leave	150.0	120
154	leave	100.0	120
155	leave	100.0	120
156	leave	100.0	120
157	leave	80.0	90
158	leave	100.0	90

Optimum_Size_for_new_pipes

1	80.0
2	150.0
3	80.0
4	80.0
5	80.0
6	80.0
7	150.0
8	80.0
9	80.0

Pressure_at_Nodes

ID	Grade	min_G	Deficit
2	25.54	20.00	5.54
3	25.12	20.00	5.12
4	22.47	20.00	2.47
5	24.58	20.00	4.58
6	28.31	20.00	8.31
7	28.09	20.00	8.09
8	28.68	20.00	8.68
9	27.34	20.00	7.34
10	25.48	20.00	5.48
11	30.47	20.00	10.47
12	35.15	20.00	15.15
13	26.34	20.00	6.34
14	27.70	20.00	7.70
15	28.24	20.00	8.24
16	26.80	20.00	6.80
17	32.96	20.00	12.96
18	32.12	20.00	12.12
19	29.89	20.00	9.89
20	33.03	20.00	13.03
21	33.82	20.00	13.82
22	32.29	20.00	12.29
23	32.01	20.00	12.01
24	33.62	20.00	13.62
25	28.36	20.00	8.36
26	27.99	20.00	7.99
27	27.78	20.00	7.78
28	28.26	20.00	8.26
29	32.24	20.00	12.24
30	32.54	20.00	12.54
31	30.83	20.00	10.83
32	31.89	20.00	11.89
33	31.42	20.00	11.42
34	32.57	20.00	12.57
35	29.79	20.00	9.79
36	28.65	20.00	8.65
37	28.59	20.00	8.59
38	27.66	20.00	7.66
39	31.46	20.00	11.46
40	30.87	20.00	10.87
41	31.45	20.00	11.45
42	27.25	20.00	7.25
43	27.28	20.00	7.28
44	30.91	20.00	10.91
45	29.10	20.00	9.10
46	29.81	20.00	9.81
47	30.09	20.00	10.09
48	29.95	20.00	9.95
49	30.63	20.00	10.63
50	31.49	20.00	11.49
51	29.79	20.00	9.79
52	28.59	20.00	8.59

```
53     28.82 20.00 8.82
54     28.45 20.00 8.45
55     30.77 20.00 10.77
56     31.58 20.00 11.58
57     27.01 20.00 7.01
58     34.41 20.00 14.41
59     20.87 20.00 0.87
60     31.30 20.00 11.30
61     33.20 20.00 13.20
62     34.66 20.00 14.66
63     29.71 20.00 9.71
64     30.13 20.00 10.13
65     27.80 20.00 7.80
66     23.63 20.00 3.63
67     22.42 20.00 2.42
68     22.38 20.00 2.38
69     20.01 20.00 0.01
70     21.43 20.00 1.43
71     21.79 20.00 1.79
72     24.27 20.00 4.27
73     26.22 20.00 6.22
74     26.95 20.00 6.95
75     27.38 20.00 7.38
76     29.83 20.00 9.83
77     30.37 20.00 10.37
78     29.88 20.00 9.88
79     26.92 20.00 6.92
80     23.16 20.00 3.16
81     26.00 20.00 6.00
82     24.00 20.00 4.00
83     26.21 20.00 6.21
84     33.30 20.00 13.30
85     35.49 20.00 15.49
86     28.91 20.00 8.91
87     20.02 20.00 0.02
88     29.77 20.00 9.77
89     29.85 20.00 9.85
90     27.13 20.00 7.13
91     25.57 20.00 5.57
92     25.10 20.00 5.10
93     26.09 20.00 6.09
94     32.36 20.00 12.36
95     32.69 20.00 12.69
96     27.40 20.00 7.40
97     25.10 20.00 5.10
98     22.71 20.00 2.71
99     24.49 20.00 4.49
100    31.11 20.00 11.11
101    32.15 20.00 12.15
102    30.53 20.00 10.53
103    28.74 20.00 8.74
104    26.48 20.00 6.48
105    22.96 20.00 2.96
106    24.00 20.00 4.00
107    23.30 20.00 3.30
108    26.21 20.00 6.21
109    20.49 20.00 0.49
110    30.96 20.00 10.96
111    20.13 20.00 0.13
112    32.60 20.00 12.60
113    25.32 20.00 5.32
114    22.58 20.00 2.58
115    23.74 20.00 3.74
MaxPressureDiff=  0.00

PenaltyCst=0.00
NetworkCst=1194363.00
EnergyCost=0.00
Total_Cost=1194363.00
Fitness    =1194363.00
Repairing pipe cost    = 224083.00
Relining/cleaning cost = 54600.00
Replacing pipe cost    = 0.00
Duplicating pipe cost  = 606480.00
New pipe cost          = 309200.00
```

A.2 Optimal Solution of fmGA6

ID	Action	Diam	C coeff.
1	duplication	300.0	130
2	leave	400.0	120
3	duplication	80.0	130
4	leave	350.0	120
5	leave	100.0	90
6	leave	80.0	80
7	leave	80.0	80
8	leave	80.0	90
9	leave	100.0	70
10	leave	100.0	90
11	leave	100.0	90
12	duplication	200.0	130
13	leave	350.0	120
14	leave	100.0	90
15	leave	100.0	90
16	leave	80.0	90
17	leave	100.0	80
18	leave	100.0	80
19	leave	100.0	80
20	leave	100.0	70
21	leave	100.0	70
22	leave	350.0	120
23	leave	100.0	90
24	leave	100.0	90
25	leave	300.0	120
26	leave	200.0	120
27	leave	100.0	90
28	leave	100.0	90
29	leave	80.0	100
30	leave	100.0	100
31	leave	150.0	120
32	leave	150.0	120
33	leave	80.0	90
34	leave	100.0	90
35	leave	200.0	120
36	leave	100.0	100
37	duplication	300.0	130
38	leave	80.0	80
39	leave	100.0	80
40	leave	100.0	90
41	leave	100.0	70
42	leave	150.0	120
43	leave	100.0	100
44	leave	80.0	100
45	reline	100.0	120
46	leave	100.0	100
47	duplication	80.0	130
48	leave	80.0	100
49	leave	100.0	100
50	leave	100.0	100
51	leave	100.0	100
52	leave	80.0	100
53	leave	100.0	100
54	leave	100.0	100
55	leave	100.0	100
56	leave	100.0	100
57	leave	80.0	100
58	leave	100.0	100
59	leave	80.0	100
60	leave	100.0	80
61	leave	100.0	80
62	leave	100.0	100
63	leave	80.0	100
64	duplication	80.0	130
65	leave	150.0	70
66	leave	150.0	80
67	leave	150.0	90
68	leave	100.0	80
69	leave	80.0	100
70	leave	100.0	100
71	leave	100.0	100
72	leave	100.0	100
73	leave	150.0	100
74	leave	100.0	90
75	duplication	80.0	130
76	leave	100.0	100
77	duplication	80.0	130

78	leave	100.0	100
79	reline	100.0	120
80	leave	100.0	100
81	leave	80.0	100
82	leave	100.0	100
83	leave	100.0	100
84	leave	80.0	80
85	reline	300.0	120
86	leave	300.0	90
87	leave	100.0	100
88	leave	300.0	100
89	leave	300.0	100
90	leave	400.0	100
91	leave	300.0	100
92	leave	150.0	100
93	leave	150.0	80
94	leave	80.0	80
95	leave	150.0	80
96	reline	150.0	120
97	leave	100.0	100
98	leave	100.0	100
99	leave	100.0	100
100	leave	80.0	120
101	leave	100.0	120
102	leave	100.0	110
103	leave	100.0	110
104	leave	60.0	90
105	leave	80.0	120
106	duplication	300.0	130
107	leave	80.0	100
108	reline	100.0	120
109	leave	100.0	120
110	leave	150.0	120
111	leave	100.0	100
112	leave	60.0	120
113	leave	100.0	90
114	leave	80.0	120
115	duplication	80.0	130
116	leave	60.0	120
117	leave	60.0	120
118	leave	60.0	120
119	leave	150.0	120
120	leave	150.0	100
121	leave	150.0	100
122	leave	60.0	120
123	leave	80.0	120
124	leave	80.0	120
125	leave	100.0	120
126	leave	100.0	120
127	leave	80.0	100
128	leave	100.0	80
129	reline	100.0	120
130	reline	100.0	120
131	leave	60.0	120
132	leave	80.0	100
133	leave	100.0	120
134	leave	60.0	120
135	leave	100.0	90
136	replace	80.0	130
137	leave	80.0	100
138	leave	400.0	100
139	duplication	150.0	130
140	leave	100.0	100
141	leave	350.0	120
142	reline	100.0	120
143	duplication	80.0	130
144	leave	100.0	100
145	duplication	80.0	130
146	leave	100.0	120
147	leave	150.0	120
148	leave	200.0	120
149	leave	250.0	120
150	leave	150.0	120
151	leave	200.0	120
152	leave	80.0	120
153	leave	150.0	120
154	leave	100.0	120
155	leave	100.0	120
156	leave	100.0	120
157	leave	80.0	90

```
158        leave   100.0        90
 Optimum_Size_for_new_pipes
1     80.0
2     150.0
3     80.0
4     80.0
5     80.0
6     80.0
7     150.0
8     80.0
9     80.0

Pressure_at_Nodes
ID    Grade Grade Deficit
2     25.04 20.00 5.04
3     24.62 20.00 4.62
4     22.64 20.00 2.64
5     22.60 20.00 2.60
6     26.66 20.00 6.66
7     26.50 20.00 6.50
8     28.68 20.00 8.68
9     27.33 20.00 7.33
10    25.30 20.00 5.30
11    30.41 20.00 10.41
12    34.96 20.00 14.96
13    26.16 20.00 6.16
14    27.63 20.00 7.63
15    28.08 20.00 8.08
16    26.55 20.00 6.55
17    32.98 20.00 12.98
18    32.25 20.00 12.25
19    29.61 20.00 9.61
20    32.82 20.00 12.82
21    33.45 20.00 13.45
22    31.74 20.00 11.74
23    31.47 20.00 11.47
24    33.08 20.00 13.08
25    27.84 20.00 7.84
26    27.49 20.00 7.49
27    27.09 20.00 7.09
28    28.14 20.00 8.14
29    31.74 20.00 11.74
30    32.32 20.00 12.32
31    31.26 20.00 11.26
32    31.94 20.00 11.94
33    31.29 20.00 11.29
34    32.10 20.00 12.10
35    29.32 20.00 9.32
36    28.18 20.00 8.18
37    28.16 20.00 8.16
38    27.18 20.00 7.18
39    30.57 20.00 10.57
40    30.01 20.00 10.01
41    30.61 20.00 10.61
42    26.43 20.00 6.43
43    27.19 20.00 7.19
44    30.94 20.00 10.94
45    27.88 20.00 7.88
46    28.23 20.00 8.23
47    29.97 20.00 9.97
48    29.57 20.00 9.57
49    30.20 20.00 10.20
50    30.53 20.00 10.53
51    29.07 20.00 9.07
52    27.89 20.00 7.89
53    28.31 20.00 8.31
54    27.94 20.00 7.94
55    30.79 20.00 10.79
56    31.64 20.00 11.64
57    26.88 20.00 6.88
58    33.92 20.00 13.92
59    20.07 20.00 0.07
60    30.86 20.00 10.86
61    32.92 20.00 12.92
62    34.10 20.00 14.10
63    29.97 20.00 9.97
64    30.28 20.00 10.28
65    28.06 20.00 8.06
66    23.52 20.00 3.52
67    22.38 20.00 2.38
```

68	22.44	20.00	2.44
69	20.00	20.00	0.00
70	21.45	20.00	1.45
71	21.75	20.00	1.75
72	22.69	20.00	2.69
73	24.50	20.00	4.50
74	27.00	20.00	7.00
75	27.12	20.00	7.12
76	29.28	20.00	9.28
77	30.39	20.00	10.39
78	29.95	20.00	9.95
79	26.79	20.00	6.79
80	22.70	20.00	2.70
81	25.96	20.00	5.96
82	23.03	20.00	3.03
83	24.57	20.00	4.57
84	31.61	20.00	11.61
85	33.14	20.00	13.14
86	27.80	20.00	7.80
87	20.20	20.00	0.20
88	30.00	20.00	10.00
89	30.52	20.00	10.52
90	27.03	20.00	7.03
91	25.54	20.00	5.54
92	25.11	20.00	5.11
93	26.19	20.00	6.19
94	33.09	20.00	13.09
95	32.14	20.00	12.14
96	27.31	20.00	7.31
97	24.66	20.00	4.66
98	22.50	20.00	2.50
99	24.36	20.00	4.36
100	31.26	20.00	11.26
101	31.99	20.00	11.99
102	29.35	20.00	9.35
103	28.27	20.00	8.27
104	24.60	20.00	4.60
105	23.01	20.00	3.01
106	23.56	20.00	3.56
107	22.50	20.00	2.50
108	26.11	20.00	6.11
109	20.58	20.00	0.58
110	29.35	20.00	9.35
111	19.99	20.00	-0.01
112	32.81	20.00	12.81
113	25.42	20.00	5.42
114	22.68	20.00	2.68
115	23.13	20.00	3.13

MaxPressureDiff= 0.00

PenaltyCst=0.00
NetworkCst=1206343.00
EnergyCost=0.00
Total_Cost=1206343.00
Fitness =1206343.00
Repairing pipe cost = 195863.00
Relining/cleaning cost = 173300.00
Replacing pipe cost = 12100.00
Duplicating pipe cost = 515880.00
New pipe cost = 309200.00

A.3 Optimal Solution of fmGA7

Rehabilitation_for_old_pipe

ID	Action	Diam	C coeff.
1	leave	400.0	120
2	duplication	500.0	130
3	replace	80.0	130
4	leave	350.0	120
5	leave	100.0	90
6	leave	80.0	80
7	leave	80.0	80
8	leave	80.0	90
9	leave	100.0	70
10	leave	100.0	90
11	leave	100.0	90
12	leave	300.0	120
13	leave	350.0	120

14	leave	100.0	90
15	leave	100.0	90
16	leave	80.0	90
17	leave	100.0	80
18	leave	100.0	80
19	reline	100.0	120
20	leave	100.0	70
21	replace	80.0	130
22	leave	350.0	120
23	leave	100.0	90
24	leave	100.0	90
25	leave	300.0	120
26	leave	200.0	120
27	leave	100.0	90
28	leave	100.0	90
29	leave	80.0	100
30	leave	100.0	100
31	leave	150.0	120
32	duplication	200.0	130
33	leave	80.0	90
34	leave	100.0	90
35	leave	200.0	120
36	leave	100.0	100
37	duplication	300.0	130
38	leave	80.0	80
39	leave	100.0	80
40	leave	100.0	90
41	leave	100.0	70
42	replace	300.0	130
43	leave	100.0	100
44	leave	80.0	100
45	leave	100.0	100
46	leave	100.0	100
47	duplication	150.0	130
48	leave	80.0	100
49	leave	100.0	100
50	leave	100.0	100
51	leave	100.0	100
52	leave	80.0	100
53	leave	100.0	100
54	duplication	100.0	130
55	duplication	150.0	130
56	leave	100.0	100
57	leave	80.0	100
58	duplication	200.0	130
59	leave	80.0	100
60	duplication	150.0	130
61	leave	100.0	80
62	leave	100.0	100
63	leave	80.0	100
64	duplication	150.0	130
65	leave	150.0	70
66	reline	150.0	120
67	leave	150.0	90
68	leave	100.0	80
69	leave	80.0	100
70	leave	100.0	100
71	leave	100.0	100
72	leave	100.0	100
73	reline	150.0	120
74	leave	100.0	90
75	leave	150.0	90
76	leave	100.0	100
77	leave	150.0	100
78	leave	100.0	100
79	leave	100.0	100
80	leave	100.0	100
81	leave	80.0	100
82	reline	100.0	120
83	leave	100.0	100
84	leave	80.0	80
85	reline	300.0	120
86	leave	300.0	90
87	leave	100.0	100
88	leave	300.0	100
89	leave	300.0	100
90	leave	400.0	100
91	leave	300.0	100
92	reline	150.0	120
93	leave	150.0	80

94	reline	80.0	120
95	leave	150.0	80
96	reline	150.0	120
97	leave	100.0	100
98	leave	100.0	100
99	leave	100.0	100
100	leave	80.0	120
101	leave	100.0	120
102	leave	100.0	110
103	leave	100.0	110
104	leave	60.0	90
105	leave	80.0	120
106	duplication	150.0	130
107	leave	80.0	100
108	replace	300.0	130
109	leave	100.0	120
110	leave	150.0	120
111	leave	100.0	100
112	leave	60.0	120
113	leave	100.0	90
114	leave	80.0	120
115	leave	100.0	90
116	leave	60.0	120
117	leave	60.0	120
118	leave	60.0	120
119	leave	150.0	120
120	leave	150.0	100
121	leave	150.0	100
122	leave	60.0	120
123	leave	80.0	120
124	leave	80.0	120
125	leave	100.0	120
126	leave	100.0	120
127	leave	80.0	100
128	leave	100.0	80
129	replace	100.0	130
130	duplication	80.0	130
131	leave	60.0	120
132	leave	80.0	100
133	leave	100.0	120
134	leave	60.0	120
135	reline	100.0	120
136	leave	60.0	120
137	duplication	100.0	130
138	leave	400.0	100
139	reline	150.0	120
140	leave	100.0	100
141	leave	350.0	120
142	replace	100.0	130
143	leave	100.0	100
144	leave	100.0	100
145	reline	100.0	120
146	leave	100.0	120
147	leave	150.0	120
148	duplication	200.0	130
149	duplication	500.0	130
150	leave	150.0	120
151	leave	200.0	120
152	leave	80.0	120
153	leave	150.0	120
154	leave	100.0	120
155	leave	100.0	120
156	leave	100.0	120
157	leave	80.0	90
158	reline	100.0	120

Optimum_Size_for_new_pipes

1	80.0
2	150.0
3	80.0
4	80.0
5	80.0
6	80.0
7	150.0
8	80.0
9	150.0

Pressure_at_Nodes

ID	Grade	min_G	Deficit
2	23.79	20.00	3.79
3	24.08	20.00	4.08

4	22.87	20.00	2.87
5	22.78	20.00	2.78
6	26.71	20.00	6.71
7	26.48	20.00	6.48
8	27.88	20.00	7.88
9	26.59	20.00	6.59
10	25.32	20.00	5.32
11	29.79	20.00	9.79
12	34.49	20.00	14.49
13	25.65	20.00	5.65
14	26.92	20.00	6.92
15	27.49	20.00	7.49
16	26.14	20.00	6.14
17	32.33	20.00	12.33
18	31.86	20.00	11.86
19	30.17	20.00	10.17
20	33.11	20.00	13.11
21	33.66	20.00	13.66
22	32.67	20.00	12.67
23	32.94	20.00	12.94
24	34.57	20.00	14.57
25	29.94	20.00	9.94
26	29.33	20.00	9.33
27	27.35	20.00	7.35
28	27.60	20.00	7.60
29	31.64	20.00	11.64
30	31.85	20.00	11.85
31	31.35	20.00	11.35
32	31.00	20.00	11.00
33	30.63	20.00	10.63
34	31.50	20.00	11.50
35	28.87	20.00	8.87
36	27.81	20.00	7.81
37	27.82	20.00	7.82
38	26.83	20.00	6.83
39	30.63	20.00	10.63
40	30.02	20.00	10.02
41	30.59	20.00	10.59
42	26.40	20.00	6.40
43	27.12	20.00	7.12
44	30.46	20.00	10.46
45	27.88	20.00	7.88
46	28.27	20.00	8.27
47	30.06	20.00	10.06
48	29.72	20.00	9.72
49	30.38	20.00	10.38
50	30.55	20.00	10.55
51	28.96	20.00	8.96
52	27.74	20.00	7.74
53	28.63	20.00	8.63
54	28.14	20.00	8.14
55	31.72	20.00	11.72
56	32.67	20.00	12.67
57	27.13	20.00	7.13
58	34.25	20.00	14.25
59	20.54	20.00	0.54
60	31.17	20.00	11.17
61	33.03	20.00	13.03
62	33.93	20.00	13.93
63	30.19	20.00	10.19
64	30.16	20.00	10.16
65	28.43	20.00	8.43
66	23.75	20.00	3.75
67	22.37	20.00	2.37
68	22.55	20.00	2.55
69	20.12	20.00	0.12
70	21.57	20.00	1.57
71	21.73	20.00	1.73
72	22.67	20.00	2.67
73	24.69	20.00	4.69
74	26.01	20.00	6.01
75	26.78	20.00	6.78
76	29.08	20.00	9.08
77	31.31	20.00	11.31
78	30.89	20.00	10.89
79	27.04	20.00	7.04
80	22.46	20.00	2.46
81	27.09	20.00	7.09
82	23.35	20.00	3.35
83	24.78	20.00	4.78

```
84     31.63 20.00 11.63
85     32.99 20.00 12.99
86     27.59 20.00 7.59
87     20.02 20.00 0.02
88     30.26 20.00 10.26
89     30.73 20.00 10.73
90     27.18 20.00 7.18
91     25.66 20.00 5.66
92     25.37 20.00 5.37
93     26.98 20.00 6.98
94     32.72 20.00 12.72
95     33.04 20.00 13.04
96     26.62 20.00 6.62
97     24.36 20.00 4.36
98     22.54 20.00 2.54
99     24.51 20.00 4.51
100    31.35 20.00 11.35
101    32.11 20.00 12.11
102    29.17 20.00 9.17
103    27.94 20.00 7.94
104    24.76 20.00 4.76
105    22.86 20.00 2.86
106    23.87 20.00 3.87
107    22.97 20.00 2.97
108    25.54 20.00 5.54
109    21.43 20.00 1.43
110    29.34 20.00 9.34
111    20.25 20.00 0.25
112    32.49 20.00 12.49
113    26.26 20.00 6.26
114    22.92 20.00 2.92
115    23.42 20.00 3.42
MaxPressureDiff=  0.00

PenaltyCst=0.00
NetworkCst=1384082.00
EnergyCost=0.00
Total_Cost=1384082.00
Fitness    =1384082.00
Repairing pipe cost      = 158192.00
Relining/cleaning cost = 277200.00
Replacing pipe cost      = 79120.00
Duplicating pipe cost    = 543570.00
New pipe cost            = 326000.00
```

A.4 Optimal Solution of fmGA8

```
Rehabilitation_for_old_pipe
ID     Action      Diam          C coeff.
1      duplication 500.0         130
2           leave  400.0         120
3      duplication 300.0         130
4           leave  350.0         120
5           leave  100.0         90
6           leave  80.0          80
7           leave  80.0          80
8           leave  80.0          90
9           leave  100.0         70
10          leave  100.0         90
11          leave  100.0         90
12          leave  300.0         120
13          leave  350.0         120
14          leave  100.0         90
15          leave  100.0         90
16          leave  80.0          90
17          leave  100.0         80
18          leave  100.0         80
19          leave  100.0         80
20          leave  100.0         70
21     duplication 150.0         130
22          leave  350.0         120
23          leave  100.0         90
24          leave  100.0         90
25          leave  300.0         120
26     duplication 100.0         130
27          leave  100.0         90
28          leave  100.0         90
29          leave  80.0          100
```

30	leave	100.0	100
31	reline	150.0	120
32	duplication	200.0	130
33	reline	80.0	120
34	leave	100.0	90
35	leave	200.0	120
36	leave	100.0	100
37	leave	200.0	120
38	leave	80.0	80
39	leave	100.0	80
40	leave	100.0	90
41	leave	100.0	70
42	leave	150.0	120
43	leave	100.0	100
44	leave	80.0	100
45	leave	100.0	100
46	leave	100.0	100
47	leave	100.0	100
48	leave	80.0	100
49	leave	100.0	100
50	duplication	80.0	130
51	leave	100.0	100
52	leave	80.0	100
53	leave	100.0	100
54	leave	100.0	100
55	reline	100.0	120
56	leave	100.0	100
57	leave	80.0	100
58	reline	100.0	120
59	leave	80.0	100
60	leave	100.0	80
61	leave	100.0	80
62	leave	100.0	100
63	leave	80.0	100
64	leave	150.0	100
65	leave	150.0	70
66	leave	150.0	80
67	leave	150.0	90
68	leave	100.0	80
69	leave	80.0	100
70	leave	100.0	100
71	leave	100.0	100
72	leave	100.0	100
73	leave	150.0	100
74	leave	100.0	90
75	leave	150.0	90
76	leave	100.0	100
77	leave	150.0	100
78	leave	100.0	100
79	duplication	150.0	130
80	leave	100.0	100
81	leave	80.0	100
82	leave	100.0	100
83	leave	100.0	100
84	reline	80.0	120
85	leave	300.0	70
86	duplication	100.0	130
87	leave	100.0	100
88	leave	300.0	100
89	leave	300.0	100
90	leave	400.0	100
91	leave	300.0	100
92	leave	150.0	100
93	duplication	80.0	130
94	leave	80.0	80
95	leave	150.0	80
96	leave	150.0	70
97	leave	100.0	100
98	reline	100.0	120
99	leave	100.0	100
100	leave	80.0	120
101	leave	100.0	120
102	leave	100.0	110
103	leave	100.0	110
104	leave	60.0	90
105	leave	80.0	120
106	leave	100.0	120
107	leave	80.0	100
108	reline	100.0	120
109	leave	100.0	120

110	leave	150.0	120
111	leave	100.0	100
112	leave	60.0	120
113	leave	100.0	90
114	leave	80.0	120
115	leave	100.0	90
116	leave	60.0	120
117	leave	60.0	120
118	leave	60.0	120
119	duplication	150.0	130
120	duplication	80.0	130
121	leave	150.0	100
122	leave	60.0	120
123	leave	80.0	120
124	leave	80.0	120
125	leave	100.0	120
126	leave	100.0	120
127	reline	80.0	120
128	leave	100.0	80
129	replace	200.0	130
130	duplication	80.0	130
131	leave	60.0	120
132	leave	80.0	100
133	leave	100.0	120
134	leave	60.0	120
135	leave	100.0	90
136	leave	60.0	120
137	leave	80.0	100
138	leave	400.0	100
139	leave	150.0	100
140	leave	100.0	100
141	duplication	80.0	130
142	leave	100.0	100
143	duplication	80.0	130
144	leave	100.0	100
145	leave	100.0	100
146	leave	100.0	120
147	leave	150.0	120
148	leave	200.0	120
149	leave	250.0	120
150	leave	150.0	120
151	leave	200.0	120
152	leave	80.0	120
153	leave	150.0	120
154	leave	100.0	120
155	leave	100.0	120
156	leave	100.0	120
157	duplication	80.0	130
158	leave	100.0	90

Optimum_Size_for_new_pipes

1	80.0
2	300.0
3	80.0
4	80.0
5	80.0
6	100.0
7	300.0
8	80.0
9	80.0

Pressure_at_Nodes

ID	Grade	min_G	Deficit
2	25.82	20.00	5.82
3	25.41	20.00	5.41
4	23.14	20.00	3.14
5	24.01	20.00	4.01
6	28.21	20.00	8.21
7	27.99	20.00	7.99
8	29.01	20.00	9.01
9	27.67	20.00	7.67
10	25.90	20.00	5.90
11	30.78	20.00	10.78
12	35.46	20.00	15.46
13	26.60	20.00	6.60
14	28.06	20.00	8.06
15	28.59	20.00	8.59
16	27.01	20.00	7.01
17	33.42	20.00	13.42
18	32.45	20.00	12.45
19	30.26	20.00	10.26

20	33.67	20.00	13.67
21	34.14	20.00	14.14
22	32.66	20.00	12.66
23	32.37	20.00	12.37
24	33.98	20.00	13.98
25	28.75	20.00	8.75
26	28.41	20.00	8.41
27	27.71	20.00	7.71
28	28.65	20.00	8.65
29	32.34	20.00	12.34
30	32.87	20.00	12.87
31	31.20	20.00	11.20
32	32.49	20.00	12.49
33	31.87	20.00	11.87
34	33.06	20.00	13.06
35	30.25	20.00	10.25
36	29.15	20.00	9.15
37	29.14	20.00	9.14
38	28.17	20.00	8.17
39	31.70	20.00	11.70
40	31.10	20.00	11.10
41	31.61	20.00	11.61
42	27.41	20.00	7.41
43	28.07	20.00	8.07
44	31.24	20.00	11.24
45	28.15	20.00	8.15
46	28.37	20.00	8.37
47	29.54	20.00	9.54
48	29.45	20.00	9.45
49	30.16	20.00	10.16
50	30.71	20.00	10.71
51	29.54	20.00	9.54
52	28.66	20.00	8.66
53	28.33	20.00	8.33
54	27.94	20.00	7.94
55	30.51	20.00	10.51
56	31.33	20.00	11.33
57	27.13	20.00	7.13
58	34.35	20.00	14.35
59	21.42	20.00	1.42
60	31.31	20.00	11.31
61	33.77	20.00	13.77
62	34.97	20.00	14.97
63	30.34	20.00	10.34
64	30.68	20.00	10.68
65	29.31	20.00	9.31
66	24.41	20.00	4.41
67	23.03	20.00	3.03
68	22.91	20.00	2.91
69	20.48	20.00	0.48
70	21.93	20.00	1.93
71	22.40	20.00	2.40
72	24.17	20.00	4.17
73	25.94	20.00	5.94
74	27.26	20.00	7.26
75	27.58	20.00	7.58
76	29.65	20.00	9.65
77	30.10	20.00	10.10
78	29.52	20.00	9.52
79	27.04	20.00	7.04
80	23.72	20.00	3.72
81	26.37	20.00	6.37
82	24.58	20.00	4.58
83	26.03	20.00	6.03
84	33.21	20.00	13.21
85	35.78	20.00	15.78
86	28.79	20.00	8.79
87	20.04	20.00	0.04
88	30.41	20.00	10.41
89	30.98	20.00	10.98
90	27.55	20.00	7.55
91	26.03	20.00	6.03
92	26.04	20.00	6.04
93	26.48	20.00	6.48
94	32.74	20.00	12.74
95	33.06	20.00	13.06
96	27.76	20.00	7.76
97	25.66	20.00	5.66
98	22.84	20.00	2.84
99	24.52	20.00	4.52

```
100    30.67 20.00 10.67
101    31.65 20.00 11.65
102    30.41 20.00 10.41
103    29.24 20.00 9.24
104    26.02 20.00 6.02
105    23.24 20.00 3.24
106    24.01 20.00 4.01
107    23.85 20.00 3.85
108    26.52 20.00 6.52
109    20.66 20.00 0.66
110    30.86 20.00 10.86
111    20.24 20.00 0.24
112    33.04 20.00 13.04
113    25.49 20.00 5.49
114    23.15 20.00 3.15
115    24.93 20.00 4.93
MaxPressureDiff=  0.00

PenaltyCst=0.00
NetworkCst=1395780.00
EnergyCost=0.00
Total_Cost=1395780.00
Fitness    =1395780.00
Repairing pipe cost    = 213990.00
Relining/cleaning cost = 67510.00
Replacing pipe cost    = 22400.00
Duplicating pipe cost  = 614380.00
New pipe cost          = 477500.00
```

Appendix B OPTIMAL SOLUTIONS OF NEW YORK TUNNELS PROBLEM BY BOUNDARY GA OPTIMISATION

B.1 Optimal Solution with Fixed Penalty Factor $\gamma = 5{,}000{,}000$

Rehabilitation_for_old_pipe

ID	Action	Diam	C coeff.
1	duplication	0	100
2	duplication	0	100
3	duplication	0	100
4	duplication	0	100
5	duplication	0	100
6	duplication	0	100
7	duplication	120	100
8	duplication	0	100
9	duplication	0	100
10	duplication	0	100
11	duplication	0	100
12	duplication	0	100
13	duplication	0	100
14	duplication	0	100
15	duplication	0	100
16	duplication	96	100
17	duplication	96	100
18	duplication	84	100
19	duplication	72	100
20	duplication	0	100
21	duplication	72	100

Pressure_at_Nodes

ID	Node_Grade	min_Grade	Deficit
2	294.24	255.00	39.24
3	286.23	255.00	31.23
4	283.88	255.00	28.88
5	281.81	255.00	26.81
6	280.20	255.00	25.20
7	277.68	255.00	22.68
8	276.38	255.00	21.38
9	273.55	255.00	18.55
10	273.52	255.00	18.52
11	273.65	255.00	18.65
12	274.94	255.00	19.94
13	277.93	255.00	22.93
14	285.46	255.00	30.46
15	293.28	255.00	38.28
16	259.84	260.00	-0.16
17	272.64	272.80	-0.16
18	260.96	255.00	5.96
19	254.82	255.00	-0.18
20	260.49	255.00	5.49

MaxPressureDiff= 0.18

PenaltyCst=956772.02
NetworkCst=37629600.00
EnergyCost=0.00
Total_Cost=38586372.02
Fitness 38586372

B.2 Optimal Solution with Fixed Penalty Factor = 7,000,000

Rehabilitation_for_old_pipe

ID	Action	Diam	C coeff.
1	duplication	0	100
2	duplication	0	100
3	duplication	0	100
4	duplication	0	100

5	duplication	0	100
6	duplication	0	100
7	duplication	108	100
8	duplication	0	100
9	duplication	0	100
10	duplication	0	100
11	duplication	0	100
12	duplication	0	100
13	duplication	0	100
14	duplication	0	100
15	duplication	0	100
16	duplication	108	100
17	duplication	108	100
18	duplication	72	100
19	duplication	60	100
20	duplication	0	100
21	duplication	84	100

Pressure_at_Nodes

ID	Node_Grade	min_Grade	Deficit
2	294.26	255.00	39.26
3	286.28	255.00	31.28
4	283.95	255.00	28.95
5	281.89	255.00	26.89
6	280.29	255.00	25.29
7	277.79	255.00	22.79
8	276.21	255.00	21.21
9	273.42	255.00	18.42
10	273.39	255.00	18.39
11	273.55	255.00	18.55
12	274.86	255.00	19.86
13	277.86	255.00	22.86
14	285.41	255.00	30.41
15	293.26	255.00	38.26
16	261.27	260.00	1.27
17	272.80	272.80	0.00
18	265.57	255.00	10.57
19	255.47	255.00	0.47
20	257.74	255.00	2.74

MaxPressureDiff= 0.00

PenaltyCst=0.00
NetworkCst=39415200.00
EnergyCost=0.00
Total_Cost=39415200.00
Fitness 39415200

B.3 Optimal Solution with Fixed Penalty Factor $\gamma = 11,000,000$

Rehabilitation_for_old_pipe

ID	Action	Diam	C coeff.
1	duplication	72	100
2	duplication	0	100
3	duplication	0	100
4	duplication	0	100
5	duplication	0	100
6	duplication	0	100
7	duplication	108	100
8	duplication	0	100
9	duplication	0	100
10	duplication	0	100
11	duplication	0	100
12	duplication	0	100
13	duplication	0	100
14	duplication	0	100
15	duplication	0	100
16	duplication	96	100
17	duplication	96	100
18	duplication	84	100
19	duplication	72	100
20	duplication	0	100
21	duplication	72	100

Pressure_at_Nodes

ID	Node_Grade	min_Grade	Deficit
2	295.05	255.00	40.05
3	286.96	255.00	31.96
4	284.59	255.00	29.59
5	282.49	255.00	27.49
6	280.87	255.00	25.87
7	278.30	255.00	23.30
8	276.67	255.00	21.67
9	273.76	255.00	18.76
10	273.73	255.00	18.73
11	273.85	255.00	18.85
12	275.12	255.00	20.12
13	278.08	255.00	23.08
14	285.55	255.00	30.55
15	293.32	255.00	38.32
16	260.04	260.00	0.04
17	272.85	272.80	0.05
18	261.14	255.00	6.14
19	255.00	255.00	0.00
20	260.69	255.00	5.69

MaxPressureDiff= 0.00

PenaltyCst=0.00
NetworkCst=39694000.00
EnergyCost=0.00
Total_Cost=39694000.00
Fitness 39694000

B.4 Optimal Solution with Fixed Penalty Factor $\gamma = 2,000,000$

Rehabilitation_for_old_pipe

ID	Action	Diam	C coeff.
1	duplication	0	100
2	duplication	0	100
3	duplication	0	100
4	duplication	0	100
5	duplication	0	100
6	duplication	0	100
7	duplication	0	100
8	duplication	0	100
9	duplication	0	100
10	duplication	0	100
11	duplication	0	100
12	duplication	0	100
13	duplication	0	100
14	duplication	0	100
15	duplication	0	100
16	duplication	84	100
17	duplication	96	100
18	duplication	84	100
19	duplication	72	100
20	duplication	0	100
21	duplication	72	100

Pressure_at_Nodes

ID	Node_Grade	min_Grade	Deficit
2	294.46	255.00	39.46
3	286.79	255.00	31.79
4	284.56	255.00	29.56
5	282.60	255.00	27.60
6	281.09	255.00	26.09
7	278.75	255.00	23.75
8	275.32	255.00	20.32
9	272.83	255.00	17.83
10	272.79	255.00	17.79
11	272.94	255.00	17.94
12	274.31	255.00	19.31
13	277.40	255.00	22.40
14	285.12	255.00	30.12
15	293.13	255.00	38.13
16	259.14	260.00	-0.86
17	271.46	272.80	-1.34
18	260.37	255.00	5.37
19	254.24	255.00	-0.76

```
20    259.81      255.00      4.81
MaxPressureDiff=  1.34

PenaltyCst=2676423.92
NetworkCst=32332800.00
EnergyCost=0.00
Total_Cost=35009223.92
Fitness   =35009223
Penalty factor for the solution= 2000000
```

Appendix C DATA FOR TRANSIENT ANALYSIS OF LOVEDAY IRRIGATION SYSTEM

C.1 Node Data

ID	Elev (m)	Demand (L/s)
2	36.00	0.00
3	32.00	0.00
4	36.00	0.00
5	38.00	0.00
6	38.00	0.00
7	38.00	0.00
8	34.00	0.00
9	21.00	0.00
10	36.00	0.00
11	32.00	0.00
12	32.00	0.00
13	32.00	0.00
14	32.00	0.00
15	33.00	0.00
16	36.00	0.00
17	37.00	0.00
18	37.00	0.00
19	38.00	0.00
20	38.00	0.00
21	36.00	0.00
22	36.00	0.00
23	36.00	0.00
24	36.00	0.00
25	25.00	0.00
26	33.00	0.00
27	31.00	0.00
28	31.00	80.00
29	29.00	0.00
30	32.00	0.00
31	31.00	0.00
32	30.00	0.00
33	32.00	0.00
34	32.50	0.00
35	32.00	0.00
36	32.00	0.00
37	33.00	0.00
38	36.00	0.00
39	35.00	0.00
40	30.50	0.00
41	36.00	0.00
42	38.00	0.00
43	38.00	0.00
44	31.00	0.00
45	31.00	0.00
46	31.00	0.00
47	33.00	0.00
48	32.50	0.00
49	36.00	0.00
50	31.00	0.00
51	36.00	0.00
52	38.00	0.00

C.2 Tank Data

ID	water level
1	51.3

C.3 Pipe Data

ID	Start Node	End Node	Length m	Diam mm	Rough. Coeff.
1	1	2	918.0	1219	120

2	2	3	554.0	1219	120
3	2	44	870.0	304	120
4	3	4	317.0	1066	120
5	4	5	475.0	1372	120
6	4	42	800.0	534	120
7	4	51	710.0	915	120
8	5	6	870.0	1372	120
9	6	7	824.0	1372	120
10	7	52	475.0	1372	120
11	52	8	475.0	1219	120
12	8	9	665.0	915	120
13	9	10	840.0	915	120
14	10	38	875.0	762	120
15	10	49	220.0	534	120
16	51	15	875.0	915	120
17	15	16	206.0	915	120
18	15	45	845.0	381	120
19	16	17	160.0	915	120
20	17	18	348.0	915	120
21	17	47	375.0	381	120
22	18	19	554.0	915	120
23	19	20	156.0	915	120
24	20	21	253.0	915	120
25	41	20	630.0	381	120
26	21	22	475.0	915	120
27	22	23	1267.0	762	120
28	23	24	317.0	685	120
29	29	24	760.0	457	120
30	24	50	475.0	685	120
31	50	25	713.0	610	120
32	25	26	917.0	457	120
33	26	27	887.0	457	120
34	27	28	507.0	381	120
39	33	34	300.0	457	120
38	33	32	825.0	229	120
37	32	30	270.0	229	120
36	30	31	187.0	381	120
35	30	29	1109.0	457	120
40	38	37	1700.0	685	120
41	37	36	400.0	685	120
42	36	29	380.0	381	120
43	36	35	400.0	610	120
44	35	33	1200.0	457	120
45	38	39	396.0	381	120
46	38	41	860.0	685	120
47	41	13	525.0	457	120
48	41	14	950.0	381	120
49	13	12	500.0	381	120
50	12	11	300.0	457	120
51	39	40	400.0	304	120
52	42	43	1780.0	457	120
53	45	46	375.0	381	120
54	47	48	970.0	381	120

C.4 Concrete Pipe Data for Transient Simulation

length (m)	Dia (mm)	from node	to node	thickness (mm)	wave speed (m/s)
918	1219	1	2	76	1098
554	1219	2	3	76	1121
870	304	2	44	29	1205
317	1066	3	4	76	1192
475	1372	4	5	76	1042
800	534	4	42	41	1170
710	915	4	51	63	1168
870	1372	5	6	76	1090
824	1372	6	7	76	1084
475	1372	7	52	76	1042
475	1219	52	8	76	1136
665	915	8	9	63	1167
840	915	9	10	63	1163
875	762	10	38	51	1151
220	534	10	49	41	1158
875	915	51	15	63	1151
206	915	15	16	63	1084
845	381	15	45	32	1170
160	915	16	17	63	1053
348	915	17	18	63	1145
375	381	17	47	32	1234

554	915	18	19	63	1121
156	915	19	20	63	1026
253	915	20	21	63	1110
630	381	41	20	32	1184
475	915	21	22	63	1136
1267	762	22	23	51	1111
317	685	23	24	48	1192
760	457	29	24	38	1176
475	685	24	50	48	1136
713	610	50	25	44	1173
917	457	25	26	38	1207
887	457	26	27	38	1167
507	381	27	28	32	1213
300	457	33	34	38	1128
825	229	33	32	25	1206
270	229	32	30	25	1184
187	381	30	31	32	1230
1109	457	30	29	38	1167
1700	685	38	37	48	1147
400	685	37	36	48	1170
380	381	36	29	32	1250
400	685	36	35	48	1170
1200	457	35	33	38	1170
396	381	38	39	32	1158
860	685	38	41	48	1132
525	457	41	13	38	1151
950	381	41	14	32	1190
500	381	13	12	32	1196
300	457	12	11	38	1128
400	304	39	40	29	1170
1780	457	42	43	38	1170
375	381	45	46	32	1234
970	381	47	48	32	1160

C.5 Hobas Pipe Data for Transient Simulation

length (m)	Dia (mm)	from node	to node	thickness (mm)	wave speed (m/s)
918	1200	1	2	17	358.59
554	1200	2	3	19	393.47
870	292	2	44	9	522.84
317	1050	3	4	19	412.76
475	1350	4	5	19	371.09
800	525	4	42	11	446.43
710	975	4	53	19	426.68
870	1350	5	6	19	357.73
824	1350	6	7	19	357.64
475	1350	7	54	19	371.09
475	1200	54	8	19	371.09
665	975	8	9	19	432.94
840	975	9	10	19	437.5
875	750	10	38	19	488.28
220	525	10	49	11	429.69
875	975	53	15	19	427.25
206	975	15	16	19	402.34
845	370	15	45	9	471.54
160	975	16	17	19	416.67
348	975	17	18	19	453.13
375	370	17	47	9	488.28
554	975	18	19	19	432.81
156	900	19	20	19	406.25
253	900	20	21	19	494.14
630	370	41	20	9	447.44
475	900	21	22	19	463.87
1267	750	22	23	19	471.35
317	675	23	24	19	495.31
760	450	29	24	9	424.11
475	675	24	50	11	495.12
713	600	50	25	11	461.98
917	450	25	26	11	477.6
887	450	26	27	11	397.88
507	370	27	28	9	371.09
300	450	33	34	10	481.34
825	256	33	32	10	468.75
270	256	32	30	12	703.13
187	370	30	31	9	585.94
1109	450	30	29	11	486.98
1700	675	38	37	14	494.79

400	675	37	36	14	426.14
380	370	36	29	9	520.83
400	600	36	35	19	446.43
1200	450	35	33	9	442.71
396	370	38	39	9	446.43
860	675	38	41	14	441.96
525	450	41	13	9	447.92
950	370	41	14	9	410.16
500	370	13	12	9	463.87
300	450	12	11	9	488.28
400	292	39	40	6	468.75
1200	450	42	43	9	426.14
375	370	45	46	9	488.28
970	370	47	48	9	473.63

Appendix D STAGE-ONE OPTIMAL SOLUTIONS FOR DESIGN OF LOVEDAY NETWORK USING CLASS II PIPES

D.1 Output of Solution 1

The Optimal Pipe Class for Each Pipe

Pipe_ID	Pipe_Diam	Pipe_thickness
1	1219	63
2	1219	63
3	310.3	13
4	1219	63
5	762	51
6	384.4	16
7	1066	63
8	762	51
9	762	51
10	610	44
11	762	51
12	762	51
13	762	51
14	233.6	10
15	457	38
16	1066	63
17	1066	63
18	384.4	16
19	1066	63
20	1066	63
21	310.3	13
22	1066	63
23	1066	63
24	915	63
25	762	51
26	762	51
27	915	63
28	762	51
29	762	51
30	762	51
31	534	41
32	534	41
33	384.4	16
34	384.4	16
35	610	44
36	233.6	10
37	233.6	10
38	310.3	13
39	457	38
40	610	44
41	610	44
42	233.6	10
43	610	44
44	534	41
45	384.4	16
46	762	51
47	384.4	16
48	384.4	16
49	384.4	16
50	381	32
51	310.3	13
52	457	38
53	310.3	13
54	384.4	16

Optimum Surge Tank Size
Surge tank diameter = 0.00 meter
Surge tank height = 0.00 meter
Steady State Pressure at Nodes

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	16.03	3	13.03
3	11.37	3	8.37
4	8.79	3	5.79
5	7.73	3	4.73
6	6.85	3	3.85
7	9.19	3	6.19

8	21.75	3	18.75
9	6.2	3	3.2
10	4.12	3	1.12
11	5.11	3	2.11
12	6.68	3	3.68
13	5.34	3	2.34
14	10.91	3	7.91
15	7.55	3	4.55
16	6.26	3	3.26
17	5.75	3	2.75
18	3.92	3	0.92
19	3.69	3	0.69
20	5.5	3	2.5
21	4.65	3	1.65
22	3.84	3	0.84
23	3.43	3	0.43
24	13.34	3	10.34
25	4.49	3	1.49
26	3.88	3	0.88
27	3.12	3	0.12
28	10.14	3	7.14
29	4.05	3	1.05
30	3.38	3	0.38
31	5.95	3	2.95
32	3.64	3	0.64
33	3.04	3	0.04
34	5.19	3	2.19
35	5.53	3	2.53
36	4.75	3	1.75
37	3.67	3	0.67
38	3.43	3	0.43
39	4.36	3	1.36
40	4.33	3	1.33
41	6.86	3	3.86
42	4.45	3	1.45
43	10.42	3	7.42
44	10.26	3	7.26
45	6.91	3	3.91
46	6.92	3	3.92
47	4.37	3	1.37
48	5.13	3	2.13
49	8.28	3	5.28
50	9.82	3	6.82
51	5.56	3	2.56

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	15.97	3	12.97
3	11.27	3	8.27
4	8.37	3	5.37
5	6.71	3	3.71
6	5.44	3	2.44
7	7.3	3	4.3
8	19.86	3	16.86
9	4.29	3	1.29
10	8.05	3	5.05
11	8.28	3	5.28
12	8.65	3	5.65
13	8.65	3	5.65
14	11.07	3	8.07
15	7.73	3	4.73
16	6.47	3	3.47
17	5.95	3	2.95
18	4.12	3	1.12
19	3.89	3	0.89
20	5.64	3	2.64
21	4.47	3	1.47
22	3.47	3	0.47
23	3.04	3	0.04
24	12.65	3	9.65
25	3.88	3	0.88
26	4.56	3	1.56
27	4.54	3	1.54
28	9.84	3	6.84
29	5.26	3	2.26
30	6.2	3	3.2
31	7.09	3	4.09
32	4.56	3	1.56
33	4.06	3	1.06
34	5.4	3	2.4
35	5.64	3	2.64

36	4.8	3	1.8
37	4.4	3	1.4
38	5.33	3	2.33
39	9.64	3	6.64
40	5.04	3	2.04
41	6.76	3	3.76
42	4.35	3	1.35
43	16.03	3	13.03
44	11.51	3	8.51
45	10.96	3	7.96
46	9.54	3	6.54
47	9.81	3	6.81
48	3.22	3	0.22
49	7.81	3	4.81
50	9.84	3	6.84
51	3.74	3	0.74

```
Maximum Steady State Pressure Deficit =0.00

PenaltyCst=0.00
Surgecost =0.00
Pipe cost =6710189.00
NetworkCst=6710189.00
Total_Cost=6710189.00
Final lower bound of penalty   = 656624.83
Final upper bound of penalty   = 1313249.65
Penalty factor for the solution= 656624.83

Number of feasible solutions = 17945.00
Number of infeasible solutions = 9056.00
```

D.2 Output of Solution 2

The Optimal Pipe Class for Each Pipe

Pipe_ID	Pipe_Diam	Pipe_thickness
1	1219	63
2	1219	63
3	310.3	13
4	1219	63
5	762	51
6	534	41
7	1219	63
8	610	44
9	762	51
10	762	51
11	534	41
12	762	51
13	762	51
14	384.4	16
15	762	51
16	1066	63
17	1066	63
18	310.3	13
19	1066	63
20	1066	63
21	310.3	13
22	1066	63
23	1066	63
24	915	63
25	762	51
26	915	63
27	915	63
28	915	63
29	762	51
30	534	41
31	762	51
32	534	41
33	534	41
34	310.3	13
35	534	41
36	610	44
37	534	41
38	310.3	13
39	610	44
40	457	38
41	233.6	10
42	233.6	10
43	233.6	10
44	233.6	10

45	384.4	16
46	762	51
47	310.3	13
48	384.4	16
49	534	41
50	384.4	16
51	310.3	13
52	384.4	16
53	384.4	16
54	384.4	16

Optimum Surge Tank Size
Surge tank diameter = 0.00 meter
Surge tank height = 0.00 meter
Steady State Pressure at Nodes

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	16.03	3	13.03
3	11.37	3	8.37
4	8.89	3	5.89
5	6.28	3	3.28
6	5.56	3	2.56
7	7.55	3	4.55
8	20.21	3	17.21
9	4.79	3	1.79
10	3.22	3	0.22
11	4.17	3	1.17
12	4.48	3	1.48
13	6.18	3	3.18
14	11.51	3	8.51
15	8.12	3	5.12
16	6.82	3	3.82
17	6.27	3	3.27
18	4.39	3	1.39
19	4.15	3	1.15
20	5.86	3	2.86
21	5.33	3	2.33
22	4.05	3	1.05
23	3.78	3	0.78
24	13.77	3	10.77
25	4.92	3	1.92
26	6.39	3	3.39
27	4.26	3	1.26
28	10.09	3	7.09
29	4.62	3	1.62
30	3.95	3	0.95
31	6.29	3	3.29
32	3.77	3	0.77
33	3.08	3	0.08
34	4.99	3	1.99
35	6.73	3	3.73
36	6.33	3	3.33
37	4.82	3	1.82
38	4.57	3	1.57
39	5.5	3	2.5
40	5.17	3	2.17
41	8.87	3	5.87
42	3.27	3	0.27
43	10.42	3	7.42
44	5.98	3	2.98
45	4.8	3	1.8
46	7.48	3	4.48
47	4.93	3	1.93
48	4.7	3	1.7
49	7.93	3	4.93
50	10.52	3	7.52
51	5.21	3	2.21

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	15.97	3	12.97
3	11.27	3	8.27
4	8.59	3	5.59
5	4.91	3	1.91
6	4	3	1
7	6.01	3	3.01
8	18.75	3	15.75
9	3.42	3	0.42
10	7.95	3	4.95
11	8.17	3	5.17
12	8.24	3	5.24
13	8.96	3	5.96

14	11.48	3	8.48
15	8.1	3	5.1
16	6.81	3	3.81
17	6.22	3	3.22
18	4.29	3	1.29
19	4.03	3	1.03
20	5.67	3	2.67
21	4.99	3	1.99
22	3.52	3	0.52
23	3.24	3	0.24
24	12.77	3	9.77
25	4	3	1
26	5.73	3	2.73
27	5.69	3	2.69
28	9.72	3	6.72
29	5.22	3	2.22
30	6.16	3	3.16
31	6.97	3	3.97
32	4.57	3	1.57
33	4.07	3	1.07
34	4.36	3	1.36
35	5.68	3	2.68
36	4.69	3	1.69
37	4.88	3	1.88
38	5.81	3	2.81
39	10.11	3	7.11
40	5.35	3	2.35
41	8.76	3	5.76
42	3.17	3	0.17
43	16.03	3	13.03
44	9.06	3	6.06
45	8.86	3	5.86
46	9.88	3	6.88
47	10.15	3	7.15
48	3.33	3	0.33
49	6.98	3	3.98
50	10.44	3	7.44
51	3.6	3	0.6

Maximum Steady State Pressure Deficit =0.00

PenaltyCst=0.00
Surgecost =0.00
Pipe cost =6800357.00
NetworkCst=6800357.00
Total_Cost=6800357.00
Final lower bound of penalty = 437749.88
Final upper bound of penalty = 875499.77
Penalty factor for the solution= 437749.88

Number of feasible solutions = 19541.00
Number of infeasible solutions = 7460.00

D.3 Output of Solution 3

The Optimal Pipe Class for Each Pipe

Pipe_ID	Pipe_Diam	Pipe_thickness
1	1219	63
2	1219	63
3	310.3	13
4	1219	63
5	534	41
6	534	41
7	1066	63
8	534	41
9	457	38
10	384.4	16
11	233.6	10
12	233.6	10
13	233.6	10
14	762	51
15	534	41
16	1219	63
17	1066	63
18	384.4	16
19	1219	63
20	1219	63
21	310.3	13
22	1219	63

23	1066	63
24	915	63
25	1066	63
26	915	63
27	762	51
28	762	51
29	534	41
30	534	41
31	610	44
32	534	41
33	384.4	16
34	384.4	16
35	534	41
36	610	44
37	610	44
38	310.3	13
39	610	44
40	762	51
41	762	51
42	762	51
43	233.6	10
44	233.6	10
45	381	32
46	1066	63
47	310.3	13
48	384.4	16
49	457	38
50	457	38
51	310.3	13
52	384.4	16
53	310.3	13
54	381	32

Optimum Surge Tank Size
Surge tank diameter = 0.000000 meter
Surge tank height = 0.000000 meter
Steady State Pressure at Nodes

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	16.03	3	13.03
3	11.37	3	8.37
4	8.97	3	5.97
5	8.22	3	5.22
6	7.27	3	4.27
7	7.47	3	4.47
8	19.83	3	16.83
9	4.02	3	1.02
10	3.19	3	0.19
11	3.6	3	0.6
12	4.28	3	1.28
13	5.97	3	2.97
14	10.79	3	7.79
15	7.25	3	4.25
16	6.03	3	3.03
17	5.63	3	2.63
18	3.98	3	0.98
19	3.63	3	0.63
20	5.49	3	2.49
21	5.24	3	2.24
22	3.88	3	0.88
23	3.61	3	0.61
24	13.28	3	10.28
25	4.43	3	1.43
26	3.81	3	0.81
27	3.06	3	0.06
28	9.95	3	6.95
29	4.5	3	1.5
30	3.83	3	0.83
31	6.33	3	3.33
32	3.82	3	0.82
33	3.13	3	0.13
34	5.22	3	2.22
35	7.07	3	4.07
36	6.25	3	3.25
37	4.44	3	1.44
38	4.14	3	1.14
39	5.07	3	2.07
40	4.96	3	1.96
41	8.87	3	5.87
42	3.27	3	0.27
43	10.42	3	7.42

44	10.13	3	7.13
45	6.79	3	3.79
46	6.69	3	3.69
47	4.01	3	1.01
48	3.51	3	0.51
49	7.77	3	4.77
50	9.19	3	6.19
51	6.58	3	3.58

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	15.97	3	12.97
3	11.27	3	8.27
4	8.31	3	5.31
5	6.54	3	3.54
6	4.54	3	1.54
7	4.64	3	1.64
8	18.15	3	15.15
9	3.79	3	0.79
10	7.66	3	4.66
11	7.75	3	4.75
12	7.91	3	4.91
13	8.63	3	5.63
14	10.78	3	7.78
15	7.25	3	4.25
16	6.04	3	3.04
17	5.62	3	2.62
18	3.94	3	0.94
19	3.57	3	0.57
20	5.38	3	2.38
21	5.02	3	2.02
22	3.29	3	0.29
23	3	3	0
24	12.13	3	9.13
25	3.36	3	0.36
26	4.04	3	1.04
27	4.03	3	1.03
28	9.7	3	6.7
29	5.29	3	2.29
30	6.22	3	3.22
31	7.17	3	4.17
32	4.81	3	1.81
33	4.31	3	1.31
34	4.81	3	1.81
35	6.8	3	3.8
36	5.94	3	2.94
37	4.45	3	1.45
38	5.38	3	2.38
39	9.68	3	6.68
40	5.02	3	2.02
41	8.76	3	5.76
42	3.17	3	0.17
43	16.03	3	13.03
44	11.22	3	8.22
45	10.67	3	7.67
46	9.11	3	6.11
47	9.37	3	6.37
48	3.29	3	0.29
49	6.74	3	3.74
50	9.14	3	6.14
51	3.37	3	0.37

Maximum Steady State Pressure Deficit =0.00

PenaltyCst=0.00
Surgecost =0.00
Pipe cost =6983174.00
NetworkCst=6983174.00
Total_Cost=6983174.00
Final lower bound of penalty = 437749.88
Final upper bound of penalty = 875499.77
Penalty factor for the solution= 437749.88

Number of feasible solutions = 17639.00
Number of infeasible solutions = 9362.00

D.4 Output of Solution 4

The Optimal Pipe Class for Each Pipe
Pipe_ID Pipe_Diam Pipe_thickness

1	1219	63
2	1219	63
3	310.3	13
4	1219	63
5	534	41
6	534	41
7	1219	63
8	534	41
9	457	38
10	457	38
11	310.3	13
12	233.6	10
13	233.6	10
14	762	51
15	534	41
16	1219	63
17	1219	63
18	310.3	13
19	1219	63
20	1219	63
21	310.3	13
22	1066	63
23	1219	63
24	915	63
25	762	51
26	915	63
27	915	63
28	1066	63
29	762	51
30	534	41
31	534	41
32	762	51
33	457	38
34	310.3	13
35	384.4	16
36	534	41
37	534	41
38	310.3	13
39	762	51
40	233.6	10
41	457	38
42	384.4	16
43	233.6	10
44	233.6	10
45	384.4	16
46	762	51
47	457	38
48	384.4	16
49	310.3	13
50	457	38
51	310.3	13
52	384.4	16
53	384.4	16
54	384.4	16

Optimum Surge Tank Size
Surge tank diameter = 0.00 meter
Surge tank height = 0.00 meter
Steady State Pressure at Nodes

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	16.03	3	13.03
3	11.37	3	8.37
4	8.9	3	5.9
5	8.04	3	5.04
6	6.9	3	3.9
7	9.35	3	6.35
8	20.69	3	17.69
9	3.59	3	0.59
10	3.32	3	0.32
11	3.73	3	0.73
12	8.19	3	5.19
13	5.91	3	2.91
14	11.87	3	8.87
15	8.59	3	5.59
16	7.38	3	4.38
17	6.98	3	3.98
18	4.75	3	1.75
19	4.57	3	1.57
20	6.22	3	3.22
21	5.57	3	2.57

22	4	3	1
23	3.83	3	0.83
24	13.05	3	10.05
25	4.9	3	1.9
26	5.77	3	2.77
27	3.64	3	0.64
28	9.86	3	6.86
29	5.99	3	2.99
30	5.32	3	2.32
31	7.64	3	4.64
32	4.57	3	1.57
33	3.13	3	0.13
34	5.03	3	2.03
35	6.22	3	3.22
36	5.09	3	2.09
37	3.97	3	0.97
38	3.73	3	0.73
39	4.66	3	1.66
40	4.9	3	1.9
41	8.87	3	5.87
42	3.27	3	0.27
43	10.42	3	7.42
44	6.33	3	3.33
45	5.15	3	2.15
46	8.03	3	5.03
47	5.48	3	2.48
48	3.08	3	0.08
49	7.99	3	4.99
50	10.25	3	7.25
51	6.52	3	3.52

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	15.97	3	12.97
3	11.27	3	8.27
4	8.17	3	5.17
5	6.17	3	3.17
6	3.77	3	0.77
7	5.89	3	2.89
8	18.89	3	15.89
9	3.88	3	0.88
10	8.07	3	5.07
11	8.17	3	5.17
12	9.22	3	6.22
13	9	3	6
14	11.84	3	8.84
15	8.57	3	5.57
16	7.37	3	4.37
17	6.95	3	3.95
18	4.67	3	1.67
19	4.48	3	1.48
20	6.02	3	3.02
21	5.14	3	2.14
22	3.19	3	0.19
23	3	3	0
24	11.58	3	8.58
25	3.44	3	0.44
26	4.88	3	1.88
27	4.83	3	1.83
28	9.09	3	6.09
29	5.56	3	2.56
30	6.5	3	3.5
31	7.29	3	4.29
32	4.46	3	1.46
33	3.95	3	0.95
34	3.71	3	0.71
35	4.56	3	1.56
36	3.1	3	0.1
37	4.48	3	1.48
38	5.41	3	2.41
39	9.72	3	6.72
40	5.39	3	2.39
41	8.76	3	5.76
42	3.17	3	0.17
43	16.03	3	13.03
44	9.41	3	6.41
45	9.22	3	6.22
46	10.44	3	7.44
47	10.7	3	7.7
48	3.38	3	0.38
49	6.74	3	3.74

```
50    10.18        3           7.18
51    3.12         3           0.12
Maximum Steady State Pressure Deficit =0.00

PenaltyCst=0.00
Surgecost =0.00
Pipe cost =6781562.00
NetworkCst=6781562.00
Total_Cost=6781562.00
Final lower bound of penalty   = 291833.26
Final upper bound of penalty   = 583666.51
Penalty factor for the solution= 311288.81

Number of feasible solutions = 16560.00
Number of infeasible solutions = 10441.00
```

D.5 Output of Solution 5

```
The Optimal Pipe Class for Each Pipe
Pipe_ID             Pipe_Diam    Pipe_thickness
1                   1219         63
2                   1219         63
3                   310.3        13
4                   1219         63
5                   762          51
6                   384.4        16
7                   1066         63
8                   762          51
9                   762          51
10                  762          51
11                  610          44
12                  534          41
13                  762          51
14                  233.6        10
15                  610          44
16                  1219         63
17                  915          63
18                  457          38
19                  1066         63
20                  1066         63
21                  310.3        13
22                  1066         63
23                  1066         63
24                  1066         63
25                  610          44
26                  1066         63
27                  1066         63
28                  762          51
29                  762          51
30                  610          44
31                  534          41
32                  457          38
33                  457          38
34                  384.4        16
35                  384.4        16
36                  534          41
37                  457          38
38                  258.2        11
39                  762          51
40                  233.6        10
41                  384.4        16
42                  457          38
43                  233.6        10
44                  233.6        10
45                  384.4        16
46                  534          41
47                  384.4        16
48                  384.4        16
49                  384.4        16
50                  384.4        16
51                  310.3        13
52                  534          41
53                  258.2        11
54                  384.4        16
Optimum Surge Tank Size
Surge tank diameter = 0.000000 meter
Surge tank height   = 0.000000 meter
Steady State Pressure at Nodes
```

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	16.03	3	13.03
3	11.37	3	8.37
4	8.82	3	5.82
5	7.81	3	4.81
6	6.98	3	3.98
7	9.53	3	6.53
8	20.23	3	17.23
9	4.72	3	1.72
10	4.13	3	1.13
11	5.07	3	2.07
12	6.64	3	3.64
13	5.31	3	2.31
14	11.79	3	8.79
15	8.01	3	5.01
16	6.72	3	3.72
17	6.19	3	3.19
18	4.35	3	1.35
19	4.12	3	1.12
20	5.95	3	2.95
21	5.64	3	2.64
22	4.88	3	1.88
23	4.03	3	1.03
24	13.65	3	10.65
25	3.85	3	0.85
26	4.72	3	1.72
27	3.97	3	0.97
28	10.05	3	7.05
29	6.2	3	3.2
30	3.11	3	0.11
31	7.48	3	4.48
32	4.45	3	1.45
33	3.01	3	0.01
34	5.26	3	2.26
35	6.73	3	3.73
36	5.4	3	2.4
37	3.65	3	0.65
38	3.4	3	0.4
39	4.34	3	1.34
40	4.29	3	1.29
41	6.86	3	3.86
42	5.73	3	2.73
43	10.42	3	7.42
44	12.65	3	9.65
45	4.46	3	1.46
46	7.37	3	4.37
47	4.83	3	1.83
48	4.45	3	1.45
49	8.59	3	5.59
50	9.8	3	6.8
51	6.57	3	3.57

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	15.97	3	12.97
3	11.27	3	8.27
4	8.43	3	5.43
5	6.88	3	3.88
6	5.7	3	2.7
7	8	3	5
8	18.78	3	15.78
9	3.29	3	0.29
10	8.53	3	5.53
11	8.75	3	5.75
12	9.12	3	6.12
13	9.12	3	6.12
14	11.85	3	8.85
15	8.12	3	5.12
16	6.85	3	3.85
17	6.32	3	3.32
18	4.46	3	1.46
19	4.22	3	1.22
20	6	3	3
21	5.59	3	2.59
22	4.66	3	1.66
23	3.7	3	0.7
24	12.87	3	9.87
25	3.22	3	0.22
26	4.66	3	1.66
27	4.64	3	1.64

```
28    9.78        3          6.78
29    6.28        3          3.28
30    7.13        3          4.13
31    7.74        3          4.74
32    4.98        3          1.98
33    4.47        3          1.47
34    4.75        3          1.75
35    6.06        3          3.06
36    3.95        3          0.95
37    5.06        3          2.06
38    5.99        3          2.99
39    10.29       3          7.29
40    5.51        3          2.51
41    6.76        3          3.76
42    5.63        3          2.63
43    16.03       3          13.03
44    13.18       3          10.18
45    11.85       3          8.85
46    9.92        3          6.92
47    10.19       3          7.19
48    3.02        3          0.02
49    8.04        3          5.04
50    9.8         3          6.8
51    5.17        3          2.17
Maximum Steady State Pressure Deficit =0.00

PenaltyCst=0.00
Surgecost =0.00
Pipe cost =6799206.00
NetworkCst=6799206.00
Total_Cost=6799206.00
Final lower bound of penalty   = 194555.50
Final upper bound of penalty   = 389111.00
Penalty factor for the solution= 207525.87

Number of feasible solutions = 19646.00
Number of infeasible solutions = 7355.00
```

D.6 Output of Solution 6

```
The Optimal Pipe Class for Each Pipe
Pipe_ID            Pipe_Diam   Pipe_thickness
1                  1219        63
2                  1219        63
3                  310.3       13
4                  1219        63
5                  1066        63
6                  384.4       16
7                  915         63
8                  1066        63
9                  1066        63
10                 1066        63
11                 1066        63
12                 1066        63
13                 1066        63
14                 915         63
15                 384.4       16
16                 1066        63
17                 915         63
18                 310.3       13
19                 915         63
20                 762         51
21                 534         41
22                 762         51
23                 915         63
24                 762         51
25                 610         44
26                 762         51
27                 762         51
28                 762         51
29                 310.3       13
30                 762         51
31                 534         41
32                 457         38
33                 384.4       16
34                 457         38
35                 384.4       16
```

36	233.6	10
37	233.6	10
38	258.2	11
39	534	41
40	762	51
41	762	51
42	457	38
43	762	51
44	534	41
45	310.3	13
46	233.6	10
47	384.4	16
48	381	32
49	384.4	16
50	310.3	13
51	310.3	13
52	534	41
53	310.3	13
54	310.3	13

Optimum Surge Tank Size
Surge tank diameter = 0.000000 meter
Surge tank height = 0.000000 meter
Steady State Pressure at Nodes

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	16.03	3	13.03
3	11.37	3	8.37
4	8.87	3	5.87
5	7.96	3	4.96
6	7.14	3	4.14
7	10.28	3	7.28
8	22.74	3	19.74
9	7.05	3	4.05
10	3.67	3	0.67
11	6.34	3	3.34
12	7.91	3	4.91
13	6.45	3	3.45
14	12.19	3	9.19
15	8.9	3	5.9
16	7.68	3	4.68
17	6.83	3	3.83
18	4.47	3	1.47
19	4.32	3	1.32
20	6.11	3	3.11
21	5.71	3	2.71
22	4.87	3	1.87
23	4.71	3	1.71
24	14.63	3	11.63
25	4.82	3	1.82
26	4.2	3	1.2
27	3.88	3	0.88
28	10.49	3	7.49
29	6.21	3	3.21
30	3.13	3	0.13
31	8.22	3	5.22
32	6.23	3	3.23
33	4.79	3	1.79
34	7.99	3	4.99
35	8.11	3	5.11
36	7.43	3	4.43
37	6.33	3	3.33
38	3.8	3	0.8
39	4.73	3	1.73
40	5.57	3	2.57
41	6.86	3	3.86
42	5.73	3	2.73
43	10.42	3	7.42
44	6.65	3	3.65
45	3.31	3	0.31
46	11.44	3	8.44
47	3.29	3	0.29
48	4.56	3	1.56
49	9.56	3	6.56
50	10	3	7
51	6.7	3	3.7

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	15.97	3	12.97
3	11.27	3	8.27
4	8.69	3	5.69

5	7.61	3	4.61
6	6.7	3	3.7
7	9.82	3	6.82
8	22.31	3	19.31
9	6.67	3	3.67
10	8.52	3	5.52
11	9.15	3	6.15
12	9.52	3	6.52
13	9.5	3	6.5
14	12.2	3	9.2
15	8.93	3	5.93
16	7.73	3	4.73
17	6.79	3	3.79
18	4.29	3	1.29
19	4.11	3	1.11
20	5.75	3	2.75
21	5.08	3	2.08
22	3.8	3	0.8
23	3.61	3	0.61
24	13.22	3	10.22
25	3.57	3	0.57
26	4.25	3	1.25
27	4.25	3	1.25
28	10.29	3	7.29
29	6.69	3	3.69
30	7.54	3	4.54
31	8.69	3	5.69
32	6.67	3	3.67
33	6.16	3	3.16
34	7.68	3	4.68
35	7.78	3	4.78
36	7.02	3	4.02
37	6.11	3	3.11
38	6.91	3	3.91
39	11.22	3	8.22
40	5.91	3	2.91
41	6.76	3	3.76
42	5.63	3	2.63
43	16.03	3	13.03
44	9.78	3	6.78
45	9.23	3	6.23
46	11.66	3	8.66
47	11.49	3	8.49
48	4.18	3	1.18
49	8.38	3	5.38
50	9.97	3	6.97
51	6.24	3	3.24

```
Maximum Steady State Pressure Deficit =0.00

PenaltyCst=0.00
Surgecost =0.00
Pipe cost =7154495.00
NetworkCst=7154495.00
Total_Cost=7154495.00
Final lower bound of penalty   = 437749.88
Final upper bound of penalty   = 875499.77
Penalty factor for the solution= 496116.53

Number of feasible solutions = 18769.00
Number of infeasible solutions = 8232.00
```

D.7 Output of Solution 7

The Optimal Pipe Class for Each Pipe

Pipe_ID	Pipe_Diam	Pipe_thickness
1	1219	63
2	1219	63
3	310.3	13
4	1219	63
5	762	51
6	384.4	16
7	1219	63
8	762	51
9	610	44
10	762	51
11	762	51
12	610	44

13	762	51
14	310.3	13
15	762	51
16	1219	63
17	1066	63
18	310.3	13
19	1066	63
20	1066	63
21	310.3	13
22	1066	63
23	1066	63
24	762	51
25	915	63
26	762	51
27	610	44
28	610	44
29	457	38
30	534	41
31	762	51
32	534	41
33	457	38
34	310.3	13
35	384.4	16
36	233.6	10
37	310.3	13
38	384.4	16
39	384.4	16
40	915	63
41	762	51
42	610	44
43	762	51
44	534	41
45	384.4	16
46	762	51
47	384.4	16
48	384.4	16
49	310.3	13
50	384.4	16
51	310.3	13
52	457	38
53	310.3	13
54	381	32

Optimum Surge Tank Size
Surge tank diameter = 0.00 meter
Surge tank height = 0.00 meter
Steady State Pressure at Nodes

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	16.03	3	13.03
3	11.37	3	8.37
4	8.83	3	5.83
5	7.83	3	4.83
6	5.4	3	2.4
7	8.66	3	5.66
8	20.47	3	17.47
9	4.97	3	1.97
10	3.26	3	0.26
11	4.2	3	1.2
12	8.66	3	5.66
13	7.33	3	4.33
14	12.54	3	9.54
15	9.17	3	6.17
16	7.87	3	4.87
17	7.35	3	4.35
18	5.5	3	2.5
19	5.27	3	2.27
20	7.06	3	4.06
21	6.66	3	3.66
22	4.2	3	1.2
23	3.74	3	0.74
24	13.73	3	10.73
25	4.89	3	1.89
26	5.76	3	2.76
27	3.62	3	0.62
28	10.56	3	7.56
29	3.02	3	0.02
30	3.44	3	0.44
31	5.19	3	2.19
32	5.26	3	2.26
33	3.82	3	0.82

34	7.51	3	4.51
35	7.67	3	4.67
36	6.98	3	3.98
37	4.77	3	1.77
38	4.52	3	1.52
39	5.46	3	2.46
40	6.31	3	3.31
41	6.86	3	3.86
42	4.45	3	1.45
43	10.42	3	7.42
44	7	3	4
45	3.66	3	0.66
46	8.53	3	5.53
47	5.85	3	2.85
48	4.88	3	1.88
49	7.9	3	4.9
50	10.55	3	7.55
51	5	3	2

Node	Node_Grade	min_Grade	Deficit
1	13.18	3	10.18
2	15.97	3	12.97
3	11.27	3	8.27
4	8.48	3	5.48
5	7.02	3	4.02
6	3.78	3	0.78
7	6.93	3	3.93
8	18.9	3	15.9
9	3.45	3	0.45
10	8.77	3	5.77
11	8.99	3	5.99
12	10.05	3	7.05
13	10.05	3	7.05
14	12.52	3	9.52
15	9.16	3	6.16
16	7.89	3	4.89
17	7.34	3	4.34
18	5.46	3	2.46
19	5.21	3	2.21
20	6.92	3	3.92
21	6.37	3	3.37
22	3.43	3	0.43
23	3.01	3	0.01
24	12.55	3	9.55
25	3.78	3	0.78
26	5.21	3	2.21
27	5.16	3	2.16
28	10.08	3	7.08
29	4.94	3	1.94
30	5.92	3	2.92
31	7.02	3	4.02
32	5.89	3	2.89
33	5.38	3	2.38
34	7.15	3	4.15
35	7.27	3	4.27
36	6.59	3	3.59
37	4.63	3	1.63
38	5.56	3	2.56
39	9.86	3	6.86
40	6.44	3	3.44
41	6.76	3	3.76
42	4.35	3	1.35
43	16.03	3	13.03
44	10.09	3	7.09
45	9.55	3	6.55
46	10.96	3	7.96
47	11.21	3	8.21
48	3.36	3	0.36
49	6.76	3	3.76
50	10.49	3	7.49
51	3.29	3	0.29

Maximum Steady State Pressure Deficit =0.00

PenaltyCst=0.00
Surgecost =0.00
Pipe cost =7021474.00
NetworkCst=7021474.00
Total_Cost=7021474.00
Final lower bound of penalty = 291833.26
Final upper bound of penalty = 583666.51
Penalty factor for the solution= 389111.01

```
Number of feasible solutions = 18038.00
Number of infeasible solutions = 8963.00
```