Genetic algorithms compared to other techniques for pipe optimisation

by

Simpson, A.R., Dandy, G.C. and Murphy, L.J.

Journal of Water Resources Planning and Management

Citation:
Simpson, A.R., Dandy, G.C. and Murphy, L.J. (1994). "Genetic algorithms compared to other techniques for pipe optimisation", *Journal of Water Resources Planning and Management,* American Society of Civil Engineers, Vol. 120, No. 4, July–August, 423–443.

For further information about this paper please email Angus Simpson at angus.simpson@adelaide.edu.au

Genetic Algorithms Compared to Other Techniques for Pipe Optimization

Angus R. Simpson, Member, ASCE, Graeme C. Dandy and Laurence J. Murphy

ABSTRACT: The genetic algorithm technique is a relatively new optimization technique. In this paper we present a methodology for optimizing pipe networks using genetic algorithms. Unknown decision variables are coded as binary strings. We investigate a three-operator genetic algorithm comprising reproduction, crossover, and mutation. Results are compared with the techniques of complete enumeration and nonlinear programming. We apply the optimization techniques to a case study pipe network. The genetic algorithm technique finds the global optimum in relatively few evaluations compared to the size of the search space.

INTRODUCTION

The construction and maintenance of pipelines for water supply costs many millions of dollars every year. As funds for the development of new infrastructure become increasingly scarce, there is an increasing desire to achieve the highest level of effectiveness for each dollar spent. Traditionally, the design of water distribution networks has been based on experience. However, there is now a significant (and growing) body of literature devoted to optimization of pipe networks.

Much of the research to date has applied deterministic optimization techniques (including linear programming, dynamic programming, and nonlinear programming) to the problems of network design. A new and developing field involves the application of stochastic optimization techniques (such as genetic algorithms and simulated annealing) to large combinatorial problems. This paper applies genetic algorithms to the problem of designing pipe networks and compares its performance with the techniques of complete enumeration and nonlinear programming.

PIPE NETWORK OPTIMIZATION PROBLEM

In its simplest form, the problem of pipe network design for gravity systems is usually formulated in the following way. For a given layout of pipes and specified demands at the nodes, find the combination of pipe sizes that gives the minimum cost, subject to the following constraints:

1. Continuity of flow must be maintained at all junctions or nodes in the network.
2. The head loss in each pipe is a known function of the flow in the pipe, its diameter, length, and hydraulic properties.
3. The total head loss around a loop must equal zero or the head loss along a path between two reservoirs must equal the elevation difference.
4. Minimum and maximum pressure head limitations must be satisfied at certain nodes in the network.
5. Minimum and maximum diameter constraints may apply to certain pipes in the network.

In addition, there may be existing pipes in the system with known diameters. One may usually assume steady state flow conditions in the network, although more than one loading condition may need to be considered. Extensions of the problem allow for valves, pumps, and storage tanks to be sized or selected.

Goulter (1987) suggested that the minimum cost design for a given layout and single loading case is a branched network (i.e., a network with no loops). In practice, loops are an essential feature of actual distribution systems as they provide an alternative flow path if there is pipe failure or for maintenance. One can achieve a degree of redundancy in pipe network optimization by ensuring that the layout has appropriate loops and by specifying minimum diameters for all pipes.

DETERMINISTIC SOLUTION TECHNIQUES

A large literature exists on the optimization of pipe networks. Lansey and Mays (1989b) provide a comprehensive review of the published literature up to 1988. The following review will concentrate on the more recent papers. The traditional method for designing pipe networks is by trial and error guided by experience. In design of pipe networks, designers often make use of commercial simulation packages such as KYPIPE (Wood 1980), WATSYS [or WATERMAX in the United States (Olde 1985)] or WATER (Fowler 1990). A common technique is to ensure for each pipe in the system that the slope of the hydraulic grade line lies within reasonable bounds. Monbaliu et al. (1990) have proposed a type of gradient search technique to achieve an efficient design. Initially, they set all pipes at their minimum diameters and a simulation package was used to determine the pressures at all nodes in the network. If the minimum pressure constraints were not satisfied, the pipe with the maximum head loss per unit length was increased to the next available size and a further simulation was carried out. They repeated this process until all pressure constraints were satisfied. They obtained near-optimal solutions in two test cases.

Enumeration

Complete enumeration is one approach for the optimization of pipe networks. The technique simulates every possible combination of discrete pipe sizes. One selects the cheapest cost network that satisfies the pressure constraints. The main drawback of this technique is the amount of computer time involved. For example, a relatively small system with eight pipes and eight possible sizes for each has 16,777,216 possible solutions. Gessler (1985) has proposed the use of selective enumeration of a severely pruned search space to optimize the design of a pipe network. One has to base the pruning of the search space on experience. Unfortunately, the global optimum may be eliminated in the process of pruning. Loubser and Gessler (1990) suggested guidelines for pruning the search space to reduce the amount of computational effort involved in enumeration; these included: (1) Grouping sets of pipes and assuming that a single diameter will be used for each group; (2) progressively storing the lowest cost solution which satisfies the constraints and eliminating all other solutions of higher cost; and (3) checking on combinations that violate the constraints. One eliminates all combinations that include the same or smaller pipe sizes.

The use of guidelines 2 and 3 removes the need to check for hydraulic feasibility of particular networks since this is computationally demanding. Despite these aids, one requires a considerable amount of computer time for large networks and there is no guarantee that the optimal solution will remain in the pruned search space after applying these heuristics.

Linear Programming

A number of researchers have used linear programming to optimize a design of a pipe network. Researchers have developed two principal approaches (Alperovits and Shamir 1977; Quindry et al. 1979). These are reviewed in Lansey and Mays (1989b).

Nonlinear Programming

One can apply a number of nonlinear optimization packages to the network design problem. They include MINOS (Murtagh and Saunders 1987), GINO (Liebman et al. 1986), and GAMS (Brooke et al. 1988). All these packages use a constrained generalized reduced gradient technique to identify a local optimum for the network problem. Constraints can be included explicitly in the model. Examples include the continuity equations, head losses around loops or between reservoirs, minimum and maximum pressure limitations, and minimum and maximum diameters. Costs can be expressed as any nonlinear function of pipe diameter and length. The limitations of the technique are as follows: (1) Because the pipe diameters are continuous variables the optimal values will not necessarily conform to the available pipe sizes; thus, rounding of the final solution is required; (2) only a local optimum is obtained; and (3) there is a limitation on the number of constraints and hence the size of network that can be handled.

Researchers have reported a number of applications of nonlinear optimization to pipe network problems (EI-Bahrawy and Smith 1985, 1987; Su et al. 1987; Lansey and Mays 1989a; Lansey et al.

1989; Duan et al. 1990). EI-Bahrawy and Smith (1985) applied MINOS to the design of water collection and distribution systems. Their model included a preprocessor to set up the data files and a postprocessor to round off the pipe sizes to commercial diameters. The model for distribution systems included pumps, check valves, and pressure-reducing valves. They obtained the optimal solution to a 33- pipe network in a reasonable amount of computer time. E1-Bahrawy and Smith (1987) applied the aforementioned optimization model to a number of case studies. They demonstrated its ability to: (1) Handle pumps and valves; (2) to find the optimal location of booster pumps and their optimal lifts; and (3) to address the optimal layout problem.

Su et al. (1987) used nonlinear programming to optimize looped pipe networks. In addition they included reliability constraints. They based the optimization model on the generalized reduced gradient (GRG) technique. A steady-state simulation model [KYPIPE, Wood (1980)] was used at each iteration to calculate pressure heads throughout the system. A separate model was used to compute the reliability of both system and nodes. They defined reliability as the probability of the design pressure being maintained at appropriate nodes in the system, given the possibility of some pipes being unavailable because of breakage. The model cannot include other elements such as pumps, valves, and storage tanks. The inclusion of constraints on reliability usually produced looped networks.

Lansey et al. (1989) considered the optimal design of pipe networks where there is uncertainty in the nodal demands, Hazen-Williams coefficients and the minimum nodal heads. They used a chance-constrained approach to convert the probabilistic constraints into deterministic ones. The constraints included the probability of the system being able to satisfy the specified nodal demands and heads. The GRG technique identified the optimum pipe sizes. The method tended to produce branched pipe networks. Lansey and Mays (1989a) used nonlinear programming to find the optimum design and layout of pipe networks. Their model was able to simulate pumps, tanks, and multiple loading cases. They embedded a simulation package [KYPIPE, Wood (1980)] in the model to ensure that the continuity and head loss constraints were satisfied. A GRG technique identified the optimum solution and the augmented Lagrangian method was used to include minimum head and other constraints. The model often resulted in branched optimum networks.

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

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

OVERVIEW OF GENETIC ALGORITHMS

A genetic algorithm is a search algorithm based on natural selection and the mechanisms of population genetics (Holland 1975; Goldberg 1989). Genetic algorithms (GAs) differ from the traditional approaches of existing optimization techniques. The simple ideas of the GA search have their roots in the biological processes of survival and adaptation. The result is an efficient algorithm with the flexibility to search complex spaces such as the solution space for the design of a looped pipe network.

To implement a GA one codes the decision variable set describing a trial solution as a string or "chromosome." Usually a binary alphabet is used for the coding. The genetic algorithm evaluates the trial solution and computes a measure of worth or "fitness" for the string. The GA successively evaluates and regenerates a collection of trial solutions called a "population." The GA creates new populations from old populations. A simple, yet powerful GA comprises three operators: reproduction, crossover, and mutation. Reproduction is a survival-of-the-fittest selection process. Crossover is the partial exchange of corresponding segments of bits between two parent strings to produce two offspring strings. Mutation is the occasional flipping of bit values to prevent the loss of a potentially useful genetic trait.

Researchers have applied GAs to a diverse range of scientific, engineering, economic, and also artistic search problems including: (1) Structural optimization (Goldberg and Samtani 1986; Sved et al. 1991); (2) pipeline operation optimization (Goldberg and Kuo 1987); (3) control system optimization for aerospace applications (Krishnakumar and Goldberg 1990); and (4) musical composition (Horner and Goldberg 1991).

Goldberg and Kuo (1987) applied GAs to the optimization of the operation of a gas pipeline operation. They considered a steady-state serial pipeline consisting of 10 pipes and 10 compressor stations each containing four pumps in series. Their objective was to minimize power, while supplying a specified flow and maintaining allowable pressures. The simple threeoperator GA found near-optimal pump operation alternatives after evaluating a fraction of the total possible number of solutions (about 3,500 from 1.10×10^{12} possible combinations).

Genetic algorithms are robust and have been proven theoretically and empirically to be able to efficiently search complex solution spaces. Gas are much less likely to restrict the search to a local optimum compared with point to point movement optimization techniques. Goldberg (1989) refers to the processing leverage of GAs as implicit parallelism.

Cembrowicz and Krauter (1977) used graph theory, linear programming, and a search procedure based on concepts from biological evolution (evolutionary strategy) to optimize pipe networks. The evolutionary strategy was developed by Rechenberg (1973) in Germany in parallel with the development of genetic algorithms by Holland (1975) in the United States. Cembrowicz and Krauter considered tree configurations of pipes that correspond to a local cost minimum. A tree network is a network that contains no loops and every node is connected. The search procedure based on biological evolution is composed of five major operators similar in nature to the genetic algorithm operators. The researchers used the evolutionary strategy technique to generate tree networks and linear programming to optimize the generated tree networks. One can determine the flows in the tree network, thus the objective cost function is subject to a set of linear constraints.

GENETIC ALGORITHMS APPLIED TO PIPE NETWORK OPTIMIZATION

The genetic algorithm technique requires that the set of decision variables be represented by a coded string of finite length (Goldberg and Kuo 1987). We have used binary coding (the minimum alphabet) in this paper. In the case of a pipe network each discrete pipe diameter (e.g., 150 mm, 250 mm, etc.) is assigned a binary code (Hadji and Murphy 1990; Murphy and Simpson 1992). As an example, consider the optimal sizing of a network with eight pipes. If eight different pipe sizes are available then a binary substring of three bits can be used to represent the options for each of the eight decision variables (see Table 1). A 24-bit binary string represents the network to be optimized made up of the eight by three-bit substrings for each decision variable.

Genetic algorithms provide a stochastic optimization approach (Goldberg 1989). One can use GAs to solve large combinatorial optimization problems. GAs use probabilistic transition rules rather than deterministic rules. The network design problem described above has 224 (16,777,216) alternatives. Complete enumeration of each of the alternatives to find the optimum configuration is possible for this number of alternatives. However, when the binary string length exceeds a length of about 30 then the number of combinations becomes very large and complete enumeration is not possible. The GA technique then becomes particularly useful.

Table 1. Options for Each Decision Variable and Associated Binary Code

Options for Each Decision Variable		Binary Coding (3)
If existing pipe (1)	If new pipe (mm) (2)	
Leave as exists	152	000
Duplicate with 152 mm	203	001
Clean existing pipe	254	010
Duplicate with 203 mm	305	011
Duplicate with 254 m	356	100
Duplicate with 305 mm	407	101
Duplicate with 356 mm	458	110
Duplicate with 407 mm	509	111

Genetic algorithms have a number of advantages over other mathematical programming techniques (Goldberg 1989). In the context of optimization of pipe network design some advantages include the following.

1. GAs deal directly with a population of solutions at any one time. These are spread throughout the solution space, so the chance of reaching the global optimum is increased significantly.
2. Each solution consists of a set of discrete pipe sizes. One does not have to round diameters up or down to obtain the final solution.
3. GAs identify a set of solutions of pipe network configurations that are close to the minimum cost solution. These configurations may correspond to quite different designs that can be then compared in terms of other important but nonquantifiable objectives.
4. GAs use objective function or fitness information only, compared with the more traditional methods that rely on existence and continuity of derivatives or other auxiliary information.

Genetic algorithms do not necessarily guarantee that the global optimum solution will be reached, although experience indicates that they will give near-optimal solutions after a reasonable number of evaluations.

To implement the genetic algorithm technique, the following parameters need to be selected.

- Population size (n) – usually 30-200.
- Probability of crossover (p_c) – usually 0.7-1.0.
- Probability of mutation (p_m) – usually 0.01-0.05. Guidelines for computing p_m are: $p_m \geq 1/n$ and $p_m \leq 1/l$ where n = population size and l = length of string (Goldberg and Koza 1990).

CASE STUDY

In this paper we consider a case study in pipe network optimization. Results from the GA technique are compared to those from complete enumeration and nonlinear optimization. We have evaluated the efficiency and effectiveness of the methods.

The pipe network used for our case study was that studied by Gessler (1985). The layout of the network is shown in Fig. 1. We converted Gessler's problem to metric units for this study. The case study network has some interesting features, including selection of diameters of five new pipes; three existing pipes may be cleaned, duplicated, or left alone; three demand patterns must be satisfied; and two supply sources are available. Fig. 1 shows solid lines representing the existing system and dashed lines depicting the new pipes. Elevations, pipe lengths, diameters and Hazen- Williams C coefficients are also given in Fig. 1. Table 2 shows the pipe costs and available diameters. We have added the two largest pipe sizes to make up eight alternatives for each decision variable for the genetic algorithm formulation. Three demand patterns (including two fire loading cases) and the associated minimum pressure head constraints are shown in Table 3.

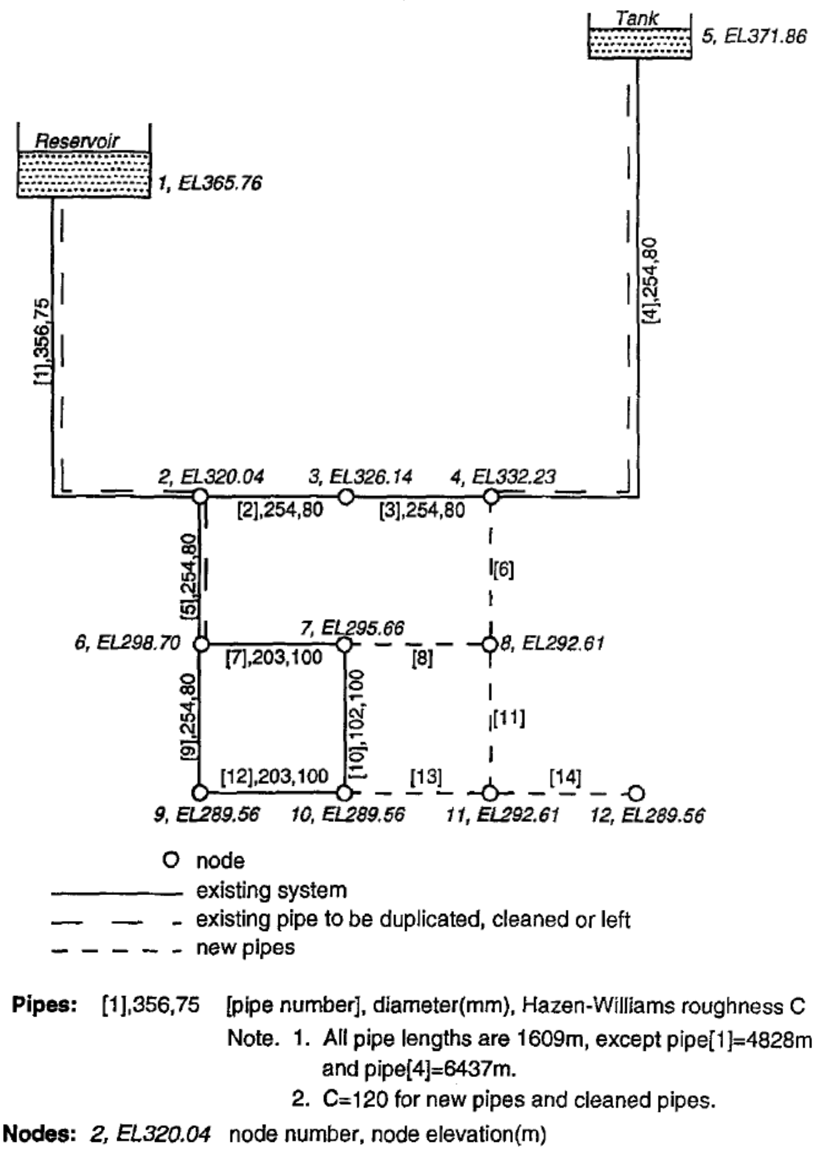

Figure 1. Layout of Gessler (1985) Problem

Table 2. Available Pipe Sizes and Associated Costs for Case Study Network

Diameter (mm) (1)	Cost of new pipe (dollars/m) (2)	Cost of cleaning existing pipe (dollars/m) (3)
152[a]	49.54	47.57
203	63.32	51.51
254	94.82	55.12
305	132.87	58.07
356	170.93	60.70
407	194.88	63.00
458	232.94[b]	-
509	264.10[b]	-

[a]Taken as the lower bound for new pipes to be sized in all optimization formulations.
[b]Approximated using linear extrapolation.

Table 3. Demand Patterns and Associated Minimum Pressures for Case Study Network

Node (1)	Demand Pattern 1		Demand Pattern 2		Demand Pattern 3	
	Demand (L/s) (2)	Minimum pressure head (m) (3)	Demand (L/s) (4)	Minimum pressure head (m) (5)	Demand (L/s) (6)	Minimum pressure head (m) (7)
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

IMPLEMENTATION OF GENETIC ALGORITHM TECHNIQUE

The genetic algorithm procedure involves the following steps.

Generation of Initial Population

The GA generates the initial population of solutions (of, say, size n = 100) using a random number generator. Each bit position in the 24-bit string for the case study problem takes on a value of either 1 or 0. Each successive three bits represent a specific option for the eight pipes under consideration (i.e., for pipes [11, [4], [51, [61, [81, [11], [13], and [141). For example, if the first six bits were 010110, then following Table 1 notation, the existing pipe [1] has to be cleaned and pipe [4] has a parallel pipe of diameter 356 mm. Each of the 100 strings represents a possible combination of pipe sizes and thus represents a different configuration of a pipe network. We used a recurrence called a linear congruential generator to generate sequences of random numbers (Barnard and Skillicorn 1988).

Computation of Network Cost

The GA considers each of the 100 strings in the population in turn. It decodes each substring into the corresponding pipe size and computes the total material and construction cost. One can also incorporate operation and maintenance costs if desired. The GA determines the costs of each network in the initial population. For generations 1, 2, 3 , . . . , etc., if crossover does not occur for a pair of strings (only for $p_c < 1.0$) the previously computed pipe cost is used.

Hydraulic Analysis of Each Network

A steady-state solver computes the heads and discharges under the three specified demand patterns shown in Table 3 for each of the networks in the population. For the initial generation 100 networks are analyzed. The actual heads are compared with the minimum allowable pressure heads and any pressure deficits are noted.

Computation of Penalty Cost

The GA assigns a penalty cost for each loading case if a network does not satisfy the minimum pressure constraints. We have used the pressure violation at the worst node. The maximum pressure deficit is multiplied by a penalty factor (e.g., K = $70,000/m of head). The GA uses the previously computed penalty costs if crossover does not occur (for generations after the initial generation). The penalty multiplier is a measure of the worth per meter attributed to pressure heads below the allowable minimum pressure head. The penalty cost should be such that near-optimal infeasible solutions are highly fit so that the optimum solution will be approached from both above and below. The optimum solution lies on the boundary between feasible and infeasible solutions (Richardson et al. 1989).

Computation of Total Network Cost

The total cost of each network in the population is taken as the sum of the network cost plus the penalty cost.

Computation of Fitnesses

The fitness of the string is taken as some function of the objective function. The GA computes the fitness for each network in the population using (1) and the total cost from "Computation of Total Network Cost." The GA searches for the minimum cost configuration in the case of a pipe network. Thus the objective function must be minimized. The inverse of the "total network cost" is one example of a form of the fitness function selected to ensure the lowest cost strings survive; it has the following form:

$$f_i = \text{Fitness}_i = \frac{1}{\text{Total Network Cost}_i} \quad \text{for strings } i = 1, 2, \ldots, n \tag{1}$$

One may select many other forms of fitness function; however, we have found the form of (1) provides the most effective solutions from the GA search.

Generation of New Population Using Reproduction Operator

The GA generates new members of the new generation by a selection scheme. We have used a proportionate selection method [also referred to as weighted roulette wheel (Goldberg 1989)]. A weighted roulette wheel has slots that are sized according to the fitness of each member in the population. The selection operator assigns each string in the population to a segment of the roulette wheel. The size of the segment is proportional to the fitness f~ of the string. The probability of selection of a particular string is

$$p_i \frac{f_i}{\sum_{j=1}^{n} f_i} \tag{2}$$

Strings with higher fitnesses (i.e., lower costs) have a higher probability of being selected.

Crossover Operator

Crossover is the partial exchange of bits between two parent strings to form two offspring strings. Consider a crossover probability of p_c = 0.70. The GA randomly picks two strings from the new population. A uniformly distributed random number is then generated in the range 0.0 to 1.0. The GA applies the crossover operator if the random number is less than 0.7. Otherwise, the GA does not apply crossover for these two particular strings. To perform crossover, a crossover point is randomly selected along the strings (e.g., position 19). The crossover operator moves the last five digits of the first string in the pair to the digit positions 20 to 24 of the second string, while the last five digits of the second string are moved to end part of the first string. On average np_c or 70 (= 100 × 0.7) strings are crossed over in each generation.

Mutation Operator

Mutation ensures no important genetic material is lost. Consider a mutation probability of p_m = 0.01. The GA considers each string bit by bit in the new generation formed as a result of reproduction and crossover. A uniformly distributed random number in the range 0.0 to 1.0 is then generated. The GA applies mutation if the random number is less than 0.01. Otherwise the GA does not apply mutation to that particular bit. The mutation operator changes the value of the bit to the opposite value (i.e., 0 to 1 or a 1 to 0).

Production of Successive Generations

Goldberg (1989) refers to the application of the three operators of reproduction, crossover, and mutation as a standard genetic algorithm. The foregoing procedure has produced a new generation using steps in the sections headed "Computation of Network Cost" through "Mutation Operator." The GA repeats the process to generate successive generations. The least-cost strings (e.g., the best 20) are stored and updated as cheaper cost alternatives are generated. The size of the network in terms of the number of decision variables determines the total number of generations to be evaluated. Typically a GA will evaluate 100-1,000 generations,

COMPLETE ENUMERATION RESULTS

It was feasible to use a complete enumeration of every possible set of pipe combination in the case study network. We were thus able to find the global optimum network configuration that satisfied the pressure constraints. For the original six alternative pipe sizes (as specified by Gessler) in Table 2 (152 mm to 407 mm) there are 3,981,312 combinations (65 by 83--including both the cleaning option and the option of not duplicating for pipes [1], [4], and [5]). Gessler (1985) used selective enumeration of a severely pruned search space of approximately 900 alternatives to optimize this problem. Gessler found a least-cost network with a cost of $1,833,700.

Complete enumeration of every alternative (network cost, hydraulic analysis, penalty function cost) for each of the three loading cases was carried out. It took 82 central processing unit (CPU) hours on a SUN 4/280 computer for 11,940,000 hydraulic network evaluations. There are two least-cost networks with a cost of $1,750,300. Table 4 shows the 39 lowest cost solutions obtained. A number of different configurations have very similar costs.

NONLINEAR OPTIMIZATION RESULTS

In addition to complete enumeration, we also solved the case study problem using the nonlinear optimization package GINO assuming that continuous sizes were available for all pipes. We carried out the GINO analysis independently of both the complete enumeration and the genetic algorithm analysis.

A cost function for new pipes was fitted to the data in Table 2. We found that the cost per meter length was approximately linear with pipe diameter for new pipes. The linear curve fitted for the new pipe cost data is not extremely accurate, especially for the smallest pipe size. The process of fitting a cost function to the data is a weakness in the application of nonlinear optimization to pipe networks. Table 5 shows the fitted cost functions for new and duplicated pipes.

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

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

The constraints for each of the three loading cases are as follows. Continuity of flow at each node (10 linear equations)

$$\sum_{j=1}^{NPJ} Q_j + q_i = 0 \quad \text{for all nodes } i = 1, \ldots, NJ \tag{3}$$

where Q_j = flows in each of the *NPJ* pipes connected to node *i* if flows away from the node are taken as positive. The demand at the node is q_i.

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

$$H_i - H_k = \frac{10.675 L_j Q_j |Q_j|^{0.852}}{C_j^{1.852} D_j^{4.8704}} \quad \text{for all pipes } j = 1, \ldots, NP \tag{4}$$

where L_j = length of pipe j; and C_j = Hazen-Williams coefficient for pipe j.

Table 4. Best 39 Ranked Case Study Problem Configurations that Satisfy Pressure Constraints

Number (1)	Total cost (dollars) (2)	Pipe Selections (mm diameter)							
		Pipe[b a] [1] (3)	Pipe[d] [4] (4)	Pipe[bcd] [5] (5)	Pipe [6] (6)	Pipe [8] (7)	Pipe [11] (8)	Pipe [13] (9)	Pipe [14] (10)
1	1,750,300	leave	dup 356	leave	305	203	203	152	254
2	1,750,300	leave	dup 356	leave	305	203	254	152	203
3	1,772,500	leave	dup 356	leave	305	203	203	203	254
4	1,772,500	leave	dup 356	leave	305	203	254	203	203
5	1,791,000	leave	dup 356	dup 203	254	203	203	152	254
6	1,799,900	leave	dup 356	clean	254	203	203	203	254
7	1,801,000	leave	dup 356	leave	305	203	254	152	254
8	1,801,000	leave	dup 356	leave	305	254	203	152	254
9	1,801,000	leave	dup 356	leave	305	254	254	152	203
10	1,811,500	leave	dup 356	leave	356	203	203	152	254
11	1,811,500	leave	dup 356	leave	356	203	254	152	203
12	1,811,500	leave	dup 356	leave	305	203	305	152	203
13	1,811,500	leave	dup 356	leave	305	203	203	152	305
14	1,813,100	leave	dup 356	dup 203	254	203	203	203	254
15	1,823,200	leave	dup 356	leave	305	203	203	254	254
16	1,823,200	leave	dup 356	leave	305	203	254	203	254
17	1,823,200	leave	dup 356	leave	305	254	203	203	254
18	1,823,200	leave	dup 356	leave	305	203	254	254	203
19	1,823,200	leave	dup 356	leave	305	254	254	203	203
20	1,828,500	leave	dup 356	clean	254	203	254	152	254
21	1,830,000	leave	dup 356	dup 152	305	203	203	152	254
22	1,830,000	leave	dup 356	dup 152	305	203	254	152	203
23[a]	1,833,700	leave	dup 356	leave	305	203	305	203	203
24	1,833,700	leave	dup 356	leave	356	203	254	203	203
25	1,833,700	leave	dup 356	leave	356	203	203	203	254
26	1,833,700	leave	dup 356	leave	305	203	203	203	305
27	1,838,500	clean	dup 305	leave	254	254	254	152	254
28	1,839,000	clean	dup 305	dup 203	254	203	203	152	254
29	1,839,000	leave	dup 356	clean	305	203	203	152	254
30	1,839,000	leave	dup 356	clean	305	203	254	152	203
31	1,839,000	leave	dup 356	clean	254	203	305	152	203
32	1,839,000	clean	dup 305	dup 152	254	203	203	203	254
33	1,839,000	clean	dup 305	dup 203	254	203	254	152	203
34	1,841,700	leave	dup 356	dup 203	254	203	254	152	254
35	1,841,700	leave	dup 356	dup 254	254	203	203	152	254
36	1,841,700	leave	dup 356	dup 254	254	203	254	152	203
37	1,841,700	leave	dup 356	dup 203	254	254	203	152	254
38	1,841,700	leave	dup 356	dup 152	254	254	203	203	254
39	1,848,000	clean	dup 305	clean	254	203	203	203	254

[a]Optimum solution as found by Gessler (1985).
[b]Leave = leave existing pipe as it is.
[c]Clean = clean existing pipe.
[d]Dup 356 = duplicate existing pipe with a 356 mm pipe.

Table 5. Fitted Cost Functions for Nonlinear Optimization Model

Pipes (1)	Cost function (dollars/m length) (2)
New	$c_1 = 611.0D - 52.9$
Pipe [1]	$c_2 = 420.1 \log(D_e) + 447.9$ (for $D_e \geq 0.356$ m)
Pipes [4] and [5]	$c_3 = 278.3 \log(D_e) + 385.0$ (for $D_e \geq 0.254$ m)

Minimum pressure head constraints at each node as given in Table 3 (10 linear inequalities for each loading case)

$$H_i \geq \overline{H}_i \quad \text{for all nodes } i = 1, \ldots, NJ \tag{5}$$

The foregoing three items give 34 constraints per loading case.

In addition there are eight lower bounds for the pipe diameters.

$$D_{\text{new}} \geq \overline{D_{\text{new}}} \quad \text{for all new pipes}, new = 1,2, \ldots, \text{NEW}_{\text{total}} \tag{6}$$

The minimum diameter for the five new pipes in the case study problem is 152 mm (refer to Table 1). The minimum equivalent diameter for each pipe that could be duplicated (pipes [1], [4], and [5]) is its existing diameter (Fig. 1). Upper bounds on pipe diameters were not required for this problem.

There were a total of 110 constraints for the problem (i.e., 3 × 34 + 8) as the three loading cases were considered simultaneously. The objective function (to be minimized) is the total cost of the network, i.e., the cost per unit length for each pipe (Table 5) multiplied by its length and summed over all pipes in the network. We optimized the following objective function in GINO

$$\text{Minimize Cost} \quad c_1 L_1 + \sum_{j \in \text{new}} c_j L_j + \sum_{k \in [4].[5]} c_k L_k \tag{7}$$

where c_1, c_j, c_k~ = cost functions in Table 5.

GINO uses a generalized reduced gradient approach for solving nonlinear optimization problems. Liebman et al. (1986) discusses the details of the method. Table 6 shows the optimum solution obtained with an estimated cost of $1,760,000. One needs to round the pipe sizes up or down to the nearest available diameter. This may involve considerable judgment, particularly for a large network, because it is not clear that the solution after rounding will automatically satisfy the minimum pressure constraints at all nodes. In this case the rounding was carried out by trial and error, with each rounded solution being checked to see if it satisfied the minimum pressure constraints for all three loading cases.

Table 6. Solution from GINO Nonlinear Optimization for Case Study Network

Pipe number (1)	Existing diameter (mm) (2)	Diameter from non-linear optimization (mm) (3)	Rounded solution [equivalent diameter (mm) and/or whether rounded up or down] (4)
[1]	356	356 (existing)	Don't duplicate
[4]	254	445	Duplicate with 356mm (455, round up)
[5]	254	254 (existing)	Don't duplicate
[6]	-	297	305 (round up)
[8]	-	205	203 (round down)
[11]	-	223	254 (round up)
[13]	-	152 (minimum)	152
[14]	-	220	254 (round up)
[Cost]		$1,760,000	$1,801,000

Table 6 shows the rounded solution that we determined for this problem. The duplication of pipe [4] with a 356-mm-diameter pipe gives an equivalent diameter of 455 mm that is slightly larger than the optimum continuous size of 445 mm. Four pipes have been rounded up in size, one has been rounded down, and three do not require rounding as they are at the respective lower bounds. The cost of this rounded solution is $1,801,000. The solution obtained by rounding up pipe [11] and pipe [14] corresponds to solution 7 in Table 4 and is about 3% more expensive than the optimum.

We modified the nonlinear programming model to calculate the pressure heads throughout the network for the particular set of pipe sizes. This was accomplished by replacing the lower bounds on pipe diameters by equations that define the diameters and deleting the minimum head constraints. We verified that the rounded solution matched the results from a commercial network solver [WATER (Fowler 1990)].

The most significant rounding in this solution was for pipes [11] and [14], whose sizes fell midway between the commercial sizes of 203 mm and 254 mm. In the foregoing solution both were rounded up in size. We tested three further alternative rounding solutions.

1. Round both pipes [11] and [14] down to 203 mm diameter.
2. Round pipe [11] down to 203 mm and pipe [14] up to 254 mm.
3. Round pipe [11] up to 254 mm and pipe [14] down to 203 mm.

Solutions 2 and 3 both satisfied the minimum pressure constraints, but solution 1 violated the constraints. The costs of solutions 2 and 3 are identical and both equal to $1,750,300. We identified these solutions as the likely optimum solutions to the problem. Comparison with the solutions obtained by complete enumeration (Table 4) shows that the two solutions obtained by nonlinear optimization are indeed the minimum cost solutions to the problem.

The optimization run took 6.8 CPU min on a SUN 4/280 for this problem. Identification of the global optima after rounding in this case could have been fortuitous as there is no guarantee that rounding from a continuous solution will give the optimum discrete sizes.

GENETIC ALGORITHMS RESULTS

In the genetic algorithm formulation the eight decision variables in the case study network were each represented by a three-bit binary substring representing eight possible alternatives as either new pipes, duplicated pipes or cleaning of an existing pipe (see Table 1). We have arranged the options available for existing pipes in Table 1 in terms of increasing cost (S/m). The GA deals with a 24-bit binary string comprising the eight by three-bit substrings representing the pipe network to be optimized. The penalty cost for each loading case was taken as the product of the maximum of all the pressure head constraint violations times a specified penalty multiplier, K (S/m). We chose a value of K = $70,000/m for the penalty function multiplier.

We developed a PASCAL computer program of a three-operator GA coupled with a Newton-Raphson network solver. The program performs a hydraulic network analysis at each function evaluation to determine the flows and pressure heads. We formulated the loop equations in the hydraulic solver to minimize computation time. The solver uses sparse matrix routines. The pipe flows and node pressures for the optimal solution are shown in Tables 7 and 8.

We performed ten GA runs for the case study network problem using different random number seeds. We permitted a maximum of 50,000 function evaluations (0.298% of the 16,777,216 possible combinations) for each run. Each GA run took 45 min CPU time on a SUN 4/280 computer. The GA finds one of the two global optimum solutions, verified by the complete enumeration (Table 4), in eight out of the 10 runs, and always identifies near-optimal solutions. The lowest-cost solutions achieved in each run are presented in Table 9.

Fig. 2 shows a plot of the cost of the best network solution in each generation and the average cost of each generation against the expected number of evaluations up to this generation for a seed of 6,000. In the first 7,000 to 8,000 evaluations, selection and crossover dominate, with the average cost being driven down from close to $4,500,000 to approximately $2,600,000. For further evaluations both crossover and mutation are driving the process. The GA finds the minimum cost configuration of $1,750,300 on three occasions. The curves in Fig. 2 show increasingly fitter populations as the run progresses, although this tends to level out after approximately 6,000 evaluations.

Table 7. Pipe Flows for Optimal Solution ($1,750,300 – Solution #1 in Table 4)

Pipe (1)	Upstream node (2)	Downstream node (3)	Flow (L/s)		
			Demand pattern 1 (4)	Demand pattern 2 (5)	Demand pattern 3 (6)
[1]	1	2	47.46	72.65	61.46
[2]	2	3	-5.70	-7.74	-3.88
[3]	3	4	-18.32	-20.09	-16.50
[4]	4	5	-97.65	-135.55	-121.50
[5]	2	6	40.54	67.50	52.73
[6]	4	8	79.33	115.46	105.00
[7]	6	7	-2.62	25.58	-2.79
[8]	7	8	-24.98	-53.93	-27.54
[9]	6	9	24.24	22.99	36.59
[10]	7	10	3.43	-2.50	5.83
[11]	8	11	35.42	42.60	58.53
[12]	9	10	11.62	10.37	23.97
[13]	10	11	-3.87	-11.05	10.87
[14]	11	12	12.62	12.62	50.47

Table 8. Allowable Pressure Heads and Actual Pressure Heads for Optimal Solution

Node (1)	Demand Pattern 1		Demand Pattern 2		Demand Patter 3	
	Allowable pressure head (m) (2)	Actual pressure head (m) (3)	Allowable pressure head (m) (4)	Actual pressure head (m) (5)	Allowable pressure head (m) (6)	Actual pressure head (m) (7)
2	28.18	36.28	14.09	24.95	14.09	30.48
3	17.61	30.47	14.09	19.32	14.09	24.52
4	17.61	26.84	14.09	16.15	14.09	20.46
6	35.22	46.87	14.09	18.68	14.09	34.34
7	35.22	50.06	10.57	12.75	14.09	37.54
8	35.22	59.23	14.09	41.29	14.09	47.93
9	35.22	51.88	14.09	24.06	14.09	34.61
10	35.22	49.79	14.09	22.38	14.09	26.65
11	35.22	47.53	14.09	24.82	14.09	18.27
12	35.22	50.00	14.09	27.29	10.57	13.72

Table 9. Results of Genetic Algorithm Runs for 50,000 Evaluations

Run number (1)	Lowest-Cost Network			
	Random number seed (2)	Best total network cost (dollars) (difference from optimum %) (3)	Evaluation number achieved (4)	Solution number (Table 4) (5)
1	1,000	1,722,500 (1.27%)	29,070	3
2	2,000	1,750,300	10,350	1
3	3,000	1,750,300	43,740	2
4	4,000	1,811,500 (3.5%)	40,860	10
5	5,000	1,750,300	17,190	2
6	6,000	1,750,300	11,070	2
7	7,000	1,750,300	10,080	1
8	8,000	1,750,300	41,490	2
9	9,000	1,750,300	12,510	1
10	10,000	1,750,300	19,890	1

Table 10 shows GA results for both 15,000 (0.0894% of search space; evaluation time of 15 min CPU) and 25,000 (0.149% of search space; evaluation time of 25 rain CPU) total evaluations. An additional 10 GA runs were performed to assess the sensitivity of the technique to parameter settings. Table 11 shows the GA parameters and lowest cost results. The GA identified the optimum solution in 9 out of 10 runs. These results do not seem to be highly sensitive to the selected GA parameters.

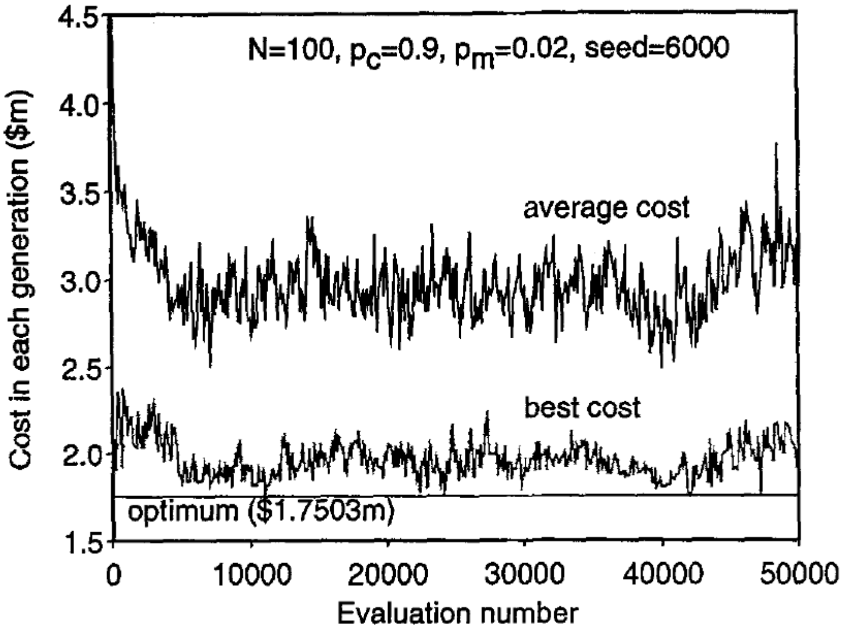

Figure 2. Generation Cost Statistics against Evaluation Number Achieved

Table 10. Results of Genetic Algorithm Runs for 15,000 and 25,000 Evaluations

Run number (1)	Random number seed (2)	LOWEST-COST NETWORK			
		15,000 Evaluations		25,000 Evaluations	
		Cost (dollars) (difference from optimum) (3)	Evaluation number achieved (4)	Cost (dollars) (difference from optimum) (5)	Evaluation number achieved (6)
1	1,000	1,811,500 (3.5%)	14,040	1,791,000 (2.3%)	23,400
2	2,000	1,750,300	10,350	1,750,300	10,350
3	3,000	1,848,000 (5.6%)	8,100	1,841,700 (5.2%)	22,410
4	4,000	1,878,100 (7.3%)	9,000	1,839,000 (5.1%)	15,660
5	5,000	1,801,000 (2.9%)	11,106	1,750,300	17,190
6	6,000	1,750,300	11,070	1,750,300	11,070
7	7,000	1,750,300	10,080	1,750,300	10,080
8	8,000	1,799,900 (2.8%)	4,410	1,799,900 (2.8%)	4,410
9	9,000	1,750,300	12,510	1,750,300	12,510
10	10,000	1,839,000 (5.1%)	7,200	1,750,300	19,890

Table 11. Results of 10 Genetic Algorithm Runs

Run number (1)	Population size (n) (2)	Probability of crossover (p_c) (3)	Probability of mutation (p_m) (4)	Lowest-cost network found (dollars) (% difference from optimum) (5)	Achieved at evaluation number (6)
1	20	0.7	0.02	1,750,300	3,962
2	50	0.7	0.02	1,750,300	20,755
3	100	0.7	0.02	1,750,300	28,000
4	150	0.7	0.02	1,750,300	9,765
5	100	0.1	0.02	1,750,300	12,920
6	100	0.5	0.02	1,799,900 (2.8%)	5,400
7	100	1.0	0.02	1,750,300	13,800
8	100	0.7	0.0	1,750,300	1,960
9	100	0.7	0.005	1,750,300	10,150
10	100	0.7	0.03	1,750,300	2,660

Note: Global minimum = $1,750,300

COMPARISON OF OPTIMIZATION METHODS

We have used three methods in this study to find the least-cost alternative for the expansion of a pipe network. These methods are complete enumeration, nonlinear optimization, and genetic algorithms.

Complete enumeration was applied to the case study pipe network involved approximately 11,940,000 hydraulic analyses and used more than 82 CPU hours of computer time. This method finds the global optimum. However, for larger network problems complete enumeration would be infeasible even on the fastest computer.

Nonlinear optimization is the fastest of the three optimization techniques applied to the case study network taking only 6.8 CPU min. One needs to fit a cost function to the discrete pipe cost data and this may lead to some inaccuracies. The designer would also have to round the continuous solution found either up or down to the nearest discrete pipe sizes. Additional computer runs are required to ensure that the rounded solution actually satisfies the pressure constraints. For large systems the round up/round down process becomes a second optimization problem.

We found that the genetic algorithm method using the standard three operators was particularly effective in finding global optimal or near-optimal solutions for the case study network. The GA required only a relatively small number of evaluations to find the optimum solution (<0.3% of the total search space for each run). The computer times are comparatively long, with a total of 45 CPU min being required for 50,000 evaluations of each of 10 genetic algorithm runs with different starting seeds, However, the GA was still effective in finding the global solution for 15,000 evaluations for each run. One difficulty with the GA technique is knowing how many evaluations will be sufficient. One may also encounter difficulties if a continuous variable needs to be represented with a binary representation, especially for large numerical values. The genetic algorithm technique is ideally suited to discrete problems such as selection of commercially available pipe sizes.

CONCLUSIONS

We have presented a methodology for the application of the genetic algorithm technique to pipe network optimization. The GA codes the pipe sizes available for selection as binary strings. We have used a simple three- operator genetic algorithm comprising reproduction, crossover, and *muta*tion. Results presented in this paper show that the genetic algorithm technique is very effective in finding near-optimal or optimal solutions for a case study network in relatively few evaluations.

The results from the genetic algorithm technique have been compared with both complete enumeration and nonlinear optimization. One may only use complete enumeration for pipe networks with relatively few pipes. Nonlinear optimization is an effective technique when applied to a small network expansions such as for the case study network; however, the problem of rounding up and down of the continuous solution to discrete pipe sizes must be addressed. The nonlinear programming method only generates one solution. The GA technique generates a whole class of alternative solutions close to the optimum. One of these alternative solutions may actually be preferred to the optimum solution based on other nonquantifiable measures. This is a major benefit of the genetic algorithm method. The genetic algorithm technique is in its infancy, and further developments should provide improvement in these search methods for practical problems.

ACKNOWLEDGMENT

The Australian Research Council partly funded this research. We gratefully acknowledge this support.

APPENDIX I. REFERENCES

Alperovits, A., and Shamir, U. (1977). "Design of optimal water distribution systems." *Water Resour. Res.,* 13(6), 885-900.

Barnard, D. T., and Skillicorn, D. B. (1988). *Pascal for engineers.* Allyn and Bacon, Inc., Boston, Mass.

Brooke, A., Kendrick, D., and Meeraus, A. (1988). *GAMS: a user's guide,* The Scientific Press, Redwood City, Calif.

Cembrowicz, R. G., and Krauter, G. E. (1977). "Optimization of urban and regional water supply systems." *Conf. Proc.; Systems Approach for Development,* IFAC, Cairo, Arab Republic of Egypt.
Dnan, 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.
E1-Bahrawy, A., and Smith, A. A. (1985). "Application of MINOS to water collection and distribution networks." *Cir. Engrg. Systems,* Vol. 2, 38-49.
EI-Bahrawy, A., and Smith, A. A. (1987). "A methodology for optimal design of pipe distribution networks." *Can. J. of Civ. Engrg.,* 14, 207-215.
Fowler, A. J. (1990). *Water.* Municipal Hydraulics, Vancouver, British Columbia, Canada.
Gessler, J. (1985). "Pipe network optimization by enumeration." *Proc., Computer Applications for Water Resources,* ASCE, New York, N.Y., 572-581.
Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley Publishing Co., Inc., Reading, Mass.
Goldberg, D. E., and Koza, J. R. (1990). "Genetic algorithms in search, optimization and machine learning." Workshop Notes, Computer Science Department, Stanford University, August 6-10.
Goldberg, D, E., and Kuo, C. H. (1987). "Genetic algorithms in pipeline optimization." *J. Computing in Civ. Engrg.,* ASCE, 1(2), 128-141.
Goldberg, D. E., and Samtani, M. P. (1986). "Engineering optimization via genetic algorithm." *Proc., 9th Conf. on Electronic Computation,* ASCE, New York, N.Y., 471-482.
Goulter, I. C. (1987). "Current and future use of systems analysis in water distribution network design." *Civ. Engrg. Systems,* Vol. 4, 175-184.
Hadji, G., and Murphy, L. J. (1990). "Genetic algorithms for pipe network optimization." *4th Year Student Civ. Engrg. Res. Rep.,* University of Adelaide, Australia.
Holland, J. H. (1975). *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, Mich.
Horner, A., and Goldberg, D. E. (1991). "Genetic algorithms and computer-assisted music composition." *Proc., 4th Int. Conf. on Genetic Algorithms,* University of California, San Diego, Calif., 437-441.
Krishnakumar, K., and Goldberg, D. E. (1990). "Control system optimization using genetic algorithms." *Proc., AIAA Guidance, Navigation, and Control Conf.,* American Institute of Aeronautics and Astronautics (AIAA).
Lansey, K. E., and Mays, L. W. (1989a). "Optimization model for water distribution system design." *J. Hydr. Engrg.,* ASCE, 115(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., Duan, N., Mays, L. W., and Tung, Y. K. (1989). "Water distribution system under uncertainties." *J. Water Resour. Plnng. and Mgmt.,* ASCE, 115(5), 630-644.
Liebman, J. S., Lasdon, L., Schrage, L., and Waren, A. (1986). *Modeling and optimization with GINO.* The Scientific Press, Palo Alto, Calif.
Loubser, B. F., and Gessler, J. (1990). "Computer-aided optimization of water distribution networks." *The Cir. Engr. in South Africa,* (Oct.), 413-422.
Monbaliu, J., Jo, J. H., Fraisse, C. W,, and Vadas, R. G. (1990). "Computer aided design of pipe networks." *Water resource systems application,* S. P. Simonovic, I. C. Goulter, D. H. Burn, and B. J. Lence, eds., Friesen Printers, Winnipeg, Canada.
Murphy, L. J., and Simpson, A. R. (1992). "Pipe optimization using genetic algorithms." *Res. Rep. No. R93,* Department of Civil Engineering, University of Adelaide, Australia.
Murtagh, B. A., and Saunders, M. A. (1987). *MINOS 5.1 user's guide.* Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, Stanford, Calif.
Olde, M. (1985). *WATSYS user's manual.* HCP Pty Ltd., Sydney, Australia.
Quindry, G. E., Brill, E. D., and Liebman, J. C. (1981). "Optimization of looped water distribution systems." *J. Envir. Engrg. Div.,* ASCE, 107(4), 665-679.

Richardson, J. T., Palmer, M. R., Liepins, G., and Hilliard, M. (1989). "Some guidelines for genetic algorithms with penalty functions." *Proc., 3rd Int. Conf. on Genetic Algorithms,* J. D. Schaffer, ed., M. Kaufmann Publishers, San Mateo, Calif.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinziplen der Biolgishen Evoltuion.* Frommann-Holzboog, Stuttgart, Germany (in German).

Su, Y. C., Mays, L. W., Duan, N., and Lansey, K. E. (1987). "Reliability-based optimization model for water distribution systems." *J. Hydr. Engrg.,* ASCE, 114(12), 1539-1556.

Sved, G., Schmid, L. J., and Simpson, A. R. (1991). "Minimum weight structures designed by genetic algorithms." *Computational mechanics; Vol.* I, Y. K. Cheung et al., eds., A. A. Balkema, Rotterdam, The Netherlands.

Wood, D. J. (1980). *User's manual computer analysis of flow in pipe networks including extended period simulations.* University of Kentucky, Lexington, Ky.

APPENDIX II. NOTATION

The following symbols are used in this paper:

C_j	=	Hazen-Williams coefficient for pipe *j*;
c_j	=	cost of pipe *j*;
D_e	=	equivalent pipe diameter (meters);
D_{new}	=	pipe diameter of pipe *new* (meters);
$\overline{D_{\mathrm{new}}}$	=	minimum allowable diameter for pipe *new* (meters);
f_i	=	fitness of member *i* in population;
H_i	=	pressure head at node *i* (meters);
H_i	=	minimum allowable pressure head at node *i* (meters);
K	=	penalty cost multiplier ($/meter);
L_j	=	length of pipe j (meters);
l	=	length of binary string;
n	=	population size for genetic algorithm;
NJ	=	number of nodes in pipe network;
NP	=	number of pipes in pipe network;
NPJ	=	number of pipes connected to junction *i*;
p_c	=	probability of crossover
p_i	=	probability of selection of member *i* of population;
p_m	=	probability of mutation;
Q_j	=	flow in pipe *j* (m3/s); and
q_i	=	demand at node *i* (m3/s).