A self-adaptive boundary search genetic algorithm and its application to water distribution systems

La recherche de limite auto-adaptative d'un algorithme génétique et son application aux systèmes de distribution d'eau

ZHENG Y. WU, *Lead Engineer, Haestad Methods, Inc., 37 Brookside Road, Waterbury, CT 06708, USA.*

ANGUS R. SIMPSON, *member, IAHR, Associate Professor, Department of Civil & Environmental Engineering, University of Adelaide, SA 5005, Australia.*

ABSTRACT
The success of the application of genetic algorithms (GA) or evolutionary optimization methods to the design and rehabilitation of water distribution systems has been shown to be an innovative approach for the water industry. The optimal design and rehabilitation of water distribution systems is a constrained non-linear optimization problem. Constraints (for example, the minimum pressure requirements) are generally handled within genetic algorithm optimization by introducing a penalty cost function. The optimal or near optimal solution is found when the pressures at some nodes are close to the minimum required pressure or at the boundary of critical constraints. This paper presents a new approach called the self-adaptive boundary search strategy for selection of penalty factor within genetic algorithm optimization. The approach co-evolves and self-adapts the penalty factor such that the genetic algorithm search is guided towards and preserved around constraint boundaries. Thus it reduces the amount of simulation computations within the GA search and enhances the efficacy at reaching the optimal or near optimal solution. To demonstrate its effectiveness, the self-adaptive boundary search strategy is applied to a case study of the optimization of a water distribution system in this paper. It has been shown that the boundary GA search strategy is effective at adapting the feasibility of GA populations for a wide range of penalty factors. As a consequence, the boundary GA has been able to successfully find the least cost solution in the case study more effectively than a GA without the boundary search strategy. Thus a reliable least cost solution is guaranteed for the GA optimization of a water distribution system.

RÉSUMÉ
Le succès de l'application des algorithmes génétiques (AG) ou des méthodes d'optimisation évolutives, à la conception et la réhabilitation des systèmes de distribution d'eau, montre que c'est une approche innovante pour l'industrie de l'eau. La conception optimale et la réhabilitation des systèmes de distribution d'eau est un problème d'optimisation non linéaire avec contraintes. Les contraintes (par exemple, les exigences de pression minimum) sont généralement prises en compte dans le cadre d'optimisation par algorithme génétique en introduisant une fonction coût de pénalisation. La solution optimale ou proche de l'optimum est obtenue quand les pressions en certains nœuds sont proches du minimum requis ou dans les limites des contraintes critiques. Cet article présente une nouvelle approche appelée la stratégie auto-adaptative de recherche de limite pour la sélection de facteurs de pénalisation dans le cadre de l'optimisation par algorithme génétique. L'approche fait évoluer conjointement et auto-adapte le facteur de pénalisation de telle sorte que la recherche d'algorithme génétique est guidée vers et maintenue autour des limites de contrainte. Cela réduit donc le volume des calculs de simulation dans la recherche par AG et rend plus efficace l'obtention de la solution optimale ou proche de l'optimum. Afin de démontrer son efficacité, la stratégie auto-adaptative de recherche de limite est appliquée, dans cet article, à une étude de cas pour l'optimisation d'un système de distribution d'eau. On montre que la stratégie de recherche de limite pour AG sert efficacement à adapter l'utilisation de populations d'AG pour un large éventail de facteurs de pénalisation. En conséquence l'AG limite est capable d'obtenir avec succès la solution de moindre coût dans le cas étudié, plus efficacement qu'un AG sans la stratégie de recherche limite. Une solution fiable de moindre coût est donc garantie pour l'optimisation par AG d'un système de distribution d'eau.

Key Words: water distribution systems, messy genetic algorithms, boundary search, optimal design and rehabilitation.

Introduction

The use of computer modeling techniques for the simulation of water distribution systems is based on a trial-and-error approach to the design of these systems. These simulation models solve the governing hydraulic equations for the flow behavior of a water distribution system. There are many user-friendly and menu-driven computer programs available today for modeling of a water distribution system. These models have become everyday tools for water engineers. Computer simulation models enable engineers to explore *what-if* questions for design, rehabilitation and operation of water systems, however, due to the extremely large search space of possible solutions, it is practically impossible to locate the lowest cost solution by a trial-and-error design approach.

Optimization of water distribution systems is considerably more difficult than simulation of these systems. It is a constrained non-linear search problem. Typical constraints include the minimum allowable hydraulic pressures at demand nodes. Continuity and energy must also be conserved for the network. The pipe friction loss is a nonlinear function of both pipe diameter and flow. Optimization of a water distribution system may require not only the sizing of pipe diameters, but also the selection of pump capacities, valve locations and setting, and also tank locations and sizes. Many of these decision variables are discrete variables. Thus the problem is a mixed continuous and discrete constrained non-linear optimization problem that is often highly dimensional. There may be many local optima in a search space. No single optimization model or a search algorithm has been able to solve the prob-

Revision received November 2, 2000. Open for discussion till August 31, 2002.

lem without compromising computational efficiency, solution accuracy and problem completeness.
During the past decade genetic algorithm (GA) optimization has been successfully applied to the optimization of water distribution system design (Murphy and Simpson 1992; Simpson et al. 1994; Dandy et al. 1996; Savic & Walters 1997; Hahal et al. 1997 and Lippai et al. 1999). Optimal or near optimal solutions for the design of water distribution systems occur when the pressures at some nodes are close to the minimum allowable pressures or at the boundary of critical constraints. These nodes are often called critical nodes. It is not possible, a priori, to predict the number or location of critical nodes. The GA technique uses a penalty function approach to penalize the fitness of solutions when a constraint is violated. The selection of penalty factors for computing the penalty cost is usually determined by trial and error. If the penalty factor is too low, many infeasible solutions will dominate the genetic algorithm population. If the penalty factor is too high, good solutions that have just failed will be eliminated permanently from the GA search process. A new approach called a self-adaptive boundary search strategy is developed in this paper for the selection of penalty factors within the framework of genetic algorithm optimization. The strategy co-evolves and self-adapts the penalty factor such that the GA search is guided to the boundary of feasible and infeasible solution spaces. In this paper the boundary search strategy is applied to a case study of the optimization of a water distribution system to demonstrate its efficiency and effectiveness.

Previous optimization models

The development of simulation modeling software has provided a sound basis for the optimization of water distribution systems. Different models have been developed in the literature for the optimization of water distribution systems. The optimization problem is posed as minimizing the cost of water distribution systems or maximizing the benefits by searching for a set of variables (e.g. pipe diameters) subject to system constraints (e.g. the minimum required pressures). Various optimization techniques have been applied to solving the problem. Optimization models may be classified depending on whether or not a hydraulic network solver is incorporated within the formulation.
An optimization model that does not incorporate a hydraulic network solver may be formulated into a two-stage (or an outer-inner) optimization procedure (Alperovits and Shamir 1977). The outer problem solves for the optimum flow status for a given network and is a nonlinear optimization problem. The inner problem determines the optimum solution of the network variables for a given flow distribution and is a linear optimization problem. An iterative process of solving the outer and inner problems is required to improve the solution. This formulation has been adopted and improved by a number of researchers (Quindry et al. 1981; Fujiwara et al. 1987; Fujiwara & Khang 1990, Eiger et al. 1994 and Loganathan et al. 1995).
In contrast, the optimization models that incorporate a hydraulic network solver are formulated by taking advantage of well developed and widely used hydraulic simulation models. A hydraulic network solver is integrated within the optimization technique and used to simulate each alternative design of water network systems during the optimization process. The simulation results are passed to optimization routines to check the feasibility of the solution. Thus this approach is able to optimize any of system components that can be modeled in a simulation model. In addition, multiple demand loadings and different types of constraints can be handled within the formulation. Su et al. (1987) integrated a generalized reduced gradient (GRG) technique with the simulation model KYPIPE. Lansey and Mays (1989) extended the methodology of Su et al (1987) for optimizing pipe sizes and also pumps and tanks under multiple demands. Duan et al. (1990) developed an optimization model that also considered system reliability. Kim and Mays (1994) proposed a mixed integer non-linear programming formulation for optimal rehabilitation of water distribution systems.
More recently, genetic algorithms (GA) and simulated annealing have been integrated with hydraulic network solvers for the optimization of water distribution systems (Simpson et al. 1994, Dandy et al. 1996, Savic & Walters 1997, Hahal et al. 1997, Lippai et al. 1999, Cunha & Sousa 1999). It has been shown that the GA is robust at searching for optimal combinations of pipe diameters and rehabilitation actions for water distribution systems. The GA technique has many advantages over traditional mathematical optimization approaches. One disadvantage is that the GA technique requires the large number of hydraulic simulation evaluations.

Constrained evolutionary optimization

Genetic algorithms or evolutionary optimization techniques are global search methods for solving complex objective and constraint functions that may be both non-differentiable and discontinuous. A constrained non-linear optimization problem is often solved by a penalty method within evolutionary algorithms.

Penalty Methods

A penalty method is designed to penalize infeasible solutions and to force the search towards the feasible solution region. The penalty cost for an infeasible solution is calculated based on constraint violations or the distance away from the feasible region. A weighted sum of a network cost plus penalty cost is computed. The weight on the penalty term is usually referred as to a penalty factor. Different penalty methods have been developed for genetic and evolutionary optimization (Michalewicz and Schoenauer 1996). A common situation for a constrained optimization is that some constraints are active at the optimal solution. An optimal or near optimal solution lies at the active constraint boundary between the feasible and infeasible regions within a solution space. It is difficult, however, to predict active constraints for a constrained non-linear optimization problem. To improve the efficacy of constrained non-linear optimization, the search for the optimal solutions should be restricted to the boundary of a solution space.

Boundary Search Methods

For a constrained optimization problem given as (Zhu et al. 1984):

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

minimizing $obj(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i x_1^{\beta_{i1}} x_2^{\beta_{i2}} \; ? \; x_n^{\beta_{in}}$

subject to $g_j(\boldsymbol{x}) \leq 0, j = 1, ..., J.$

where α_i, β_{ij} are constant coefficients in an objective function and $\beta_{ij} \geq 0$. It has been proven (Zhu et al. 1984) that the optimal solution is at the boundary of a constraint set if there is an optimal solution. A boundary search method (Zhu et al. 1984), based on the complex method of Box (1965), was proposed for the optimization problems in the case of the optimal solution at the boundary. Wu and Wang (1992) extended the method for solving structural optimization problems in the formulation of fuzzy logic that quantified the subjective uncertainties of active constraints.
Evolutionary optimization methods have recognized the need for searching the boundary between feasible and infeasible regions within the solution space. Two evolutionary algorithms (Schoenauer and Michalewicz 1996) have been designed for searching the solution boundary of two continuous numerical functions. The algorithms start with initializing feasible solutions and then generate the new population of individuals by using specific crossover and mutation operators that keep the offspring on the boundary surface of constraints. The results obtained showed that the algorithms designed specifically for these two numerical functions are very effective and efficient at reaching the global optima. Schoenauer and Michalewicz (1996) also considered the design of generic evolutionary operators for searching the boundary, however, it is not possible to construct such operators using an analytical approach for highly non-linear and implicit constraints. This situation is often the case in real-world engineering optimization problems.

Self-adaptive boundary optimization

A new evolutionary optimization strategy is introduced in this paper. The method is referred to as self-adaptive boundary optimization. Although it has been realized that evolutionary computation techniques have the potential for incorporating genetic operators that search the boundary of the feasible and infeasible regions, a method has not yet been developed to be generally effective for constrained non-linear optimization. A penalty method is often applied for evolutionary algorithms to solve constrained non-linear optimization problems. Each of the penalty methods, however, involves one (or more than one) penalty factor that is employed to construct a weighted sum of the constraint violations and an objective function. The augmented term of penalty cost distorts the landscape of the original objective function. It is the penalty factor that defines the degree of the distortion, and consequently has a major influence on the performance of a search algorithm. Different values of a penalty factor may lead a search process to different solution regions and thus different solutions. Therefore the penalty factor needs to be tuned to achieve the best performance for a constrained non-linear optimization problem. Tuning the penalty factor may require a large number of trial and error runs, thus it may be a time consuming process. It also produces a dilemma, in that, too small a penalty factor leads the search towards an infeasible region, but too big a penalty factor restricts the search inside the feasible region and prevents any acceleration of the search process from using the infeasible region. As a result the search algorithm may fail to reach the optimal solution. The self-adaptive boundary search method as developed in this paper is to adapt and co-evolve the penalty factor within a genetic algorithm such that the GA population is adjusted (or forced) to search the boundary of the feasible and infeasible regions. Thus it is believed that the performance of a GA search can be improved for constrained non-linear optimization problems. The self-adaptive search method also relieves the need for tuning the penalty factor for solving a constrained non-linear optimization problem.

Co-evolutionary Penalty Factor

The co-evolutionary concept was originally introduced by Bäck et al. (1991) within evolutionary strategy to evolve a mutation probability. A similar approach is employed in this paper to co-evolve a penalty factor to preserve the population around the boundary between feasible and infeasible solution spaces. Along with the original decision variables that are being optimized, a penalty factor is treated as an additional decision variable and optimized within a genetic and evolutionary algorithm.
The implementation of a co-evolutionary penalty factor in a genetic-based search algorithm is now considered. For any given constrained optimization problem, each string of a genetic algorithm population encodes a solution of original decision variables and also a decision variable for the penalty factor that is to be applied to the solution when a constraint is violated. As shown in Figure 1, the sub-string representing a penalty factor as a decision variable is included as part of a problem solution string. The penalty factor sub-string is mapped onto a prescribed penalty factor in a range of [γ^{nin}, γ^{max}]. The mapping is given by:

$$\gamma_n = \gamma^{min} + \frac{\gamma^{max} - \gamma^{min}}{b^{rr} - 1} \left[\sum_{dex=1}^{rr} a_{n,dex}^{r} b^{dex-1} \right] \qquad (1)$$

where γ_n is the penalty for string n; γ^{min} is the lower bound of the penalty factor; γ^{max} is the upper bound of the penalty factor; $a_{n,dex}^{r}$ is the dex-th bit of the sub-string encoding the penalty factor for string n; rr is the number of the bits encoding the penalty factor; b is 2 for a binary string or 10 for a real number string. The mapped penalty factor is then used for computing the penalty cost

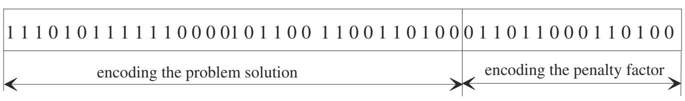


Fig. 1 String Representation of Co-evolution of Penalty Factor

of the solution when constraints are found being violated. The fitness of a string is evaluated by using the sum of the network cost and a penalty cost. Thus the fitness is contributed to not only by the original decision variable string corresponding to the design pipes in the network, but also by the penalty factor substring. The strings that have encoded penalty factors and produce better solutions will survive longer, consequently, the preferred penalty factor will spread through the population and be evolved over generations. In this way, a penalty factor is allowed to coevolve as the GA search proceeds.

The co-evolutionary penalty factor concept has been applied to the optimization of a water distribution system to investigate its effectiveness. A detailed description is given in a later section. The initial results of the application indicated that the genetic algorithm search favored the lower bound of the penalty factor γ^{min}. After a few generations, the lower bound of the penalty factor dominated the population. However, a small value of γ^{min} leads the population far away from the feasible region. In contrast, a large value of γ^{min} forces the population to converge to local optima within the feasible region. Ideally, the lower and higher bounds of a penalty factor are adapted in such a way that the search is maintained around a constraint boundary, and also that a genetic algorithm is allowed to approach the optimal or near optimal solutions from both feasible and infeasible regions.

Self-adaptive Penalty Factor

A heuristic rule has been developed for adjusting the lower and higher bounds of a penalty factor to adapt the GA population towards and around the boundary according to the GA population feasibility. The population feasibility is defined as the ratio of the number of feasible solutions to a population size (the sum of feasible and infeasible solutions in one generation). This serves as a feasibility measure of the GA population referred to as φ. For instance, if the ratio in a population is 0.6, it means that 60% of the population are inside the feasible region boundary, and that the other 40% are outside the boundary in the infeasible region. The lower the ratio is, the more the infeasible solutions are in a population. The penalty factor bounds are adapted to maintain the population feasibility measure within a certain range, which preserves the GA population around the boundary during the genetic algorithm optimization. The rule that has been developed for adapting the penalty factor bounds γ^{min} and γ^{max} is as follows.

Increase the penalty factor bounds if the population feasibility φ *is less than the minimum feasibility ratio* φ_{min};

Or

Decrease the penalty factor bounds if the population feasibility φ *is greater than the maximum feasibility ratio* φ_{max}.

The population feasibility is checked every T generations. More precisely, the heuristic rule can be given as:

A Self-Adaptive Penalty Rule:

If $(N_t - N_{t-1} \geq T$ and $\varphi_t < \varphi_{min})$ ***then***

$$\gamma_t^{min} = (1.0+\alpha)\gamma_{t-1}^{max};$$

$$\gamma_t^{max} = (1.0+\alpha)\gamma_{t-1}^{max};$$

If $(N_t - N_{t-1} \geq T$ and $\varphi_t > \varphi_{max})$ ***then***

$$\gamma_t^{min} = (1.0-\alpha)\gamma_{t-1}^{min};$$

$$\gamma_t^{max} = (1.0-\alpha)\gamma_{t-1}^{min};$$

where N_{t-1} and N_t are the numbers of generations at penalty adaptation step (t - 1) and t respectively; T is 20 ~ 40, the number of generations, over which a penalty range is adapted; φ_t is the ratio of feasible solutions to the total number of solutions between adaptation step (t - 1) and t; φ_{min} is 0.3 ~ 0.2, the minimum feasibility ratio; φ_{max} is 0.6 ~ 0.8, the maximum feasibility ratio; γ_{t-1}^{min} and γ_{t-1}^{max} are the lower and upper bounds of penalty factors at adaptation step (t - 1); γ_t^{min} and γ_t^{max} are the lower and upper bounds of penalty factors at adaptation step t; α *is* a coefficient with $0 < \alpha < 1.0$, typically it is approximately 0.2. The initial upper and lower bounds of a penalty factor can be specified with quite a large range since they are self-adapted along the optimization process.

The self-adaptive boundary search strategy has been employed in conjunction with the fast messy genetic algorithm (fmGA) (Goldberg, Deb, Kargupta and Harik 1993) and applied to the optimization of water distribution systems. Detailed descriptions of a messy genetic algorithm are given in Goldberg et al. (1989, 1990). Its applications to the optimization of water distribution systems can be found in Wu and Simpson (1996, 1997). The main features of the fast messy genetic algorithm are given below.

Features of fast messy genetic algorithm

A messy genetic algorithm (mGA) (Goldberg, Korb and Deb 1989, 1990) was carefully designed to solve problems more efficiently and effectively than standard genetic algorithms. A standard GA uses a fixed-length string representation, simple recombination operators and appropriate GA control parameters. It has demonstrated a capability in solving non-linear optimization problems, however, it requires an exponential number of function evaluations to achieve a reliable solution to highly dimensional problems or problems with bounded difficulty. This has been overcome, in the messy GA, by introducing paired-value strings of variable length and by splitting the artificial evolution process into two phases namely identifying *building blocks* or partial strings contained in good solutions to a problem and exchanging them effectively. The key components of the fast messy GA are described as follows.

Messy GA Representation

A messy GA represents a gene bit by a pair of bit tag and bit value, noted as (*bit tag*, *bit value*), in a string of variable length. A bit tag is the sequential order of a gene bit in a full-length string. For binary string representation, each bit takes a value of either 0 or 1. Variable-length strings may be either under-specified when some bits are missing or over-specified when multiple values are given for one bit. The variation of a string length permits the messy GA to uncover building blocks and also to exchange them effectively.

Building Block Filtering

A building block is a short partial string, defined as a similarity template of gene bit values that contribute above-average fitness to a string. Identifying building blocks in messy GA was originally achieved by completely enumerating all the combinations of a desired length of strings. The disadvantage of the complete enumeration is that an exponentially large number of function evaluations are required (Goldberg et al. 1989). It is computationally costly even for a small problem. This bottleneck of the original messy GA has been overcome by introducing gene filtering in the fast messy GA (Goldberg et al. 1993) to replace the complete enumeration procedure. During the gene filtering stage, a string is selected, half of the current genes in the string are randomly deleted, which reduces the string length to just half of the original string. The shortened strings are evaluated and then the same procedure of string selection and gene deletion is applied until the string length is equal to a desired length.

Messy GA Reproduction

Exchanging building blocks is the so-called juxtapositional phase in a messy GA. During this phase, a messy GA starts with a selection and follows by a string reproduction. It is similar to the action of a standard GA, but uses different genetic operators including *thresholding selection*, *cut* and *splice* (Goldberg et al. 1990). *Thresholding selection* is a tournament selection except that a string is not allowed to compete with another string unless it has, at least, a threshold number of bits from the same bit locations. *Cut* acts to cut a string into two strings while *splice* concatenates two strings to form one individual. Both *cut* and *splice* operations are designed to effectively exchange the building blocks to reproduce a GA population of solution strings.
The fast messy GA combines two evolutionary phases of building block filtering and justapositional operations into one artificial evolution process. It iterates over a number of outer loops that are referred to as eras (*era* = 1, 2, 3, …, *n*). In each era, the fast messy GA starts with an initialization and continues with a building block filtering and the juxtapositional operations. The outer iterations are repeated until the optimal or near optimal solutions are found in a population.

An application to water distribution systems

The self-adaptive boundary search strategy has been implemented within a fast messy genetic algorithm (fmGA), and integrated with a hydraulic network solver EPANET (Rossman 1994) for the optimization of water distribution systems. The optimization is undertaken to search for the least cost solution of pipe sizes ($\vec{D}$) subject to the minimum allowable pressure requirements at demand nodes. A penalty function is introduced for calculating the penalty cost of infeasible solutions. A penalty cost is defined as a function of the maximum pressure deficit and given as:

$$\boldsymbol{penalty}(\vec{D}) = \gamma\left\{\max_{i=1}^{I}\left[\max_{j=1}^{J}\left\{0, \boldsymbol{H}_{ji}^{\min} - \boldsymbol{H}_{ji}\right\}\right]\right\} \quad (2)$$

where $\vec{D}$ is the vector of pipe diameters of one solution; H_{ji} and $\boldsymbol{H}_{ji}^{\min}$ are the pressure and the minimum allowable pressure at node *j* under demand loading case *i*, respectively; *I* is the number of demand loading cases; *J* is the number of demand nodes and γ is the penalty factor that is to be adapted and co-evolved during the GA optimization. The New York City Water Tunnels problem has been chosen for this application. Although this problem may not represent all the characteristics (i.e. pumps, tanks, valve selection) of the optimization of water distribution systems, it has been thoroughly studied in the literature and provides an excellent example to investigate the performance of the self-adaptive boundary GA search strategy.

Table 1 Pipe Data for the Existing Tunnels of New York City Water Supply System (as of 1969)

Pipe No.	Start Node	End Node	Length ft	(m)	Existing Diameters inch	(mm)
[1]	1	2	11600	(3536)	180	(4570)
[2]	2	3	19800	(6035)	180	(4570)
[3]	3	4	7300	(2225)	180	(4570)
[4]	4	5	8300	(2530)	180	(4570)
[5]	5	6	8600	(2621)	180	(4570)
[6]	6	7	19100	(5822)	180	(4570)
[7]	7	8	9600	(2926)	132	(3350)
[8]	8	9	12500	(3810)	132	(3350)
[9]	9	10	9600	(2926)	180	(4570)
[10]	11	9	11200	(3414)	204	(5180)
[11]	12	11	14500	(4420)	204	(5180)
[12]	13	12	12200	(3719)	204	(5180)
[13]	14	13	24100	(7346)	204	(5180)
[14]	15	14	21100	(6431)	204	(5180)
[15]	1	15	15500	(4724)	204	(5180)
[16]	10	17	26400	(8047)	72	(1830)
[17]	12	18	31200	(9510)	72	(1830)
[18]	18	19	24000	(7315)	60	(1520)
[19]	11	20	14400	(4389)	60	(1520)
[20]	20	16	38400	(11704)	60	(1520)
[21]	9	16	26400	(8047)	72	(1830)

All pipes including existing and new pipes are assumed to have a Hazen Williams C = 100.

New York City Water Tunnels Problem

The problem was first posed in 1969 by Schaake and Lai to determine the optimal size of the pipes to be added to the system as shown in Figure 2. A set of existing tunnels needs to have pipes added, referred to as duplication pipes, in parallel to the existing pipes to meet increased demand conditions. Details of the existing pipe system are given in Table 1. Pipe costs for the new duplicate pipes were given by a fitted cost function $c_f = 1.1D^{1.24}$, where D is the diameter of pipe. The available pipe sizes as listed in Table 2 are discretized from the cost function. The minimum allowable hydraulic grades are given in Table 3 for all nodes. This problem has been used as a case study by many researchers (Schaake and Lai 1969; Quindry et al. 1981; Gessler 1982; Bhave 1985; Morgan and Goulter 1985; Dandy et al. 1996; Savic & Walters 1997 and Lippai et al. 1999).

To apply the boundary GA search strategy, four binary bits were

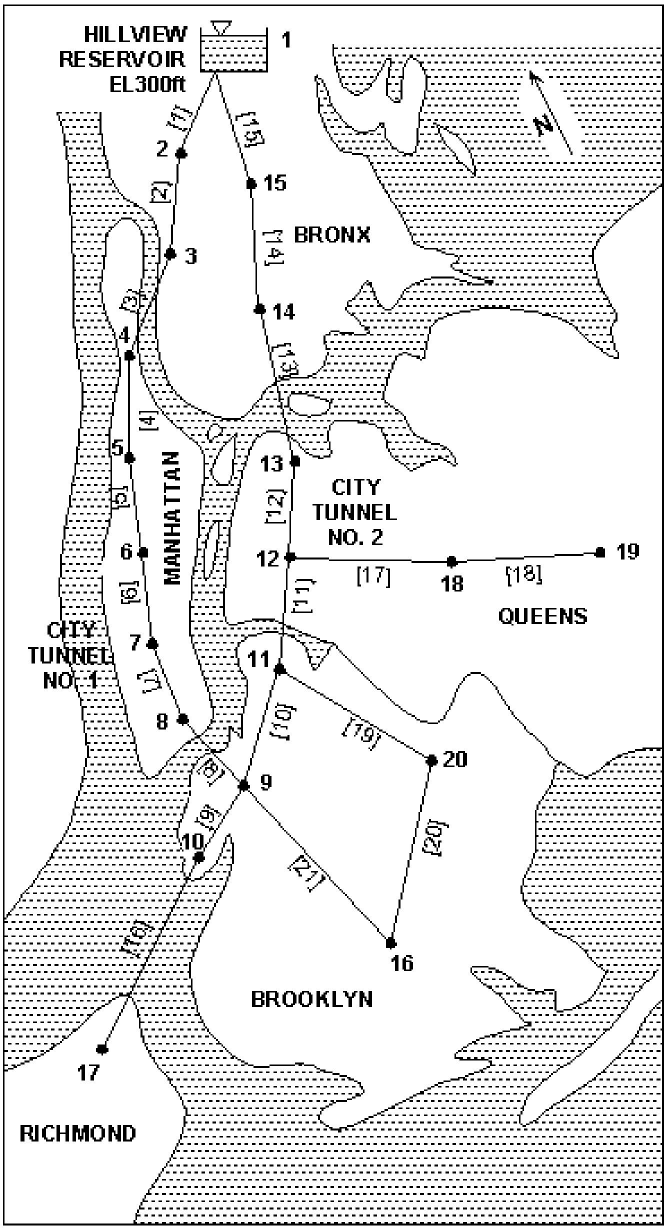


Fig. 2 New York City Water Supply Tunnels (as of 1969)

Table 2 Available New Pipe Sizes and Costs for New York City Tunnel Expansion (as of 1969)

Diameter inch	(mm)	Pipe Cost $/ft	($/m)*
36	(910)	93.5	(306.8)
48	(1220)	134.0	(439.6)
60	(1520)	176.0	(577.4)
72	(1830)	221.0	(725.1)
84	(2130)	267.0	(876.0)
96	(2440)	316.0	(1,036.8)
108	(2740)	365.0	(1,197.5)
120	(3050)	417.0	(1,368.1)
132	(3350)	469.0	(1,538.7)
144	(3660)	522.0	(1,712.6)
156	(3960)	577.0	(1,893.0)
168	(4270)	632.0	(2,073.5)
180	(4570)	689.0	(2,260.5)
192	(4880)	746.0	(2,447.5)
204	(5180)	804.0	(2,637.8)

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

Table 3 Node Data for New York Water Supply Tunnel Problem (as of 1969)

Node	Demand ft^3/s	(m^3/s)	Minimum HGL ft	(m)
1	Reservoir		300	(91.4)
2	92.4	(2.62)	255	(77.7)
3	92.4	(2.62)	255	(77.7)
4	88.2	(2.5)	255	(77.7)
5	88.2	(2.5)	255	(77.7)
6	88.2	(2.5)	255	(77.7)
7	88.2	(2.5)	255	(77.7)
8	88.2	(2.5)	255	(77.7)
9	170.0	(4.81)	255	(77.7)
10	1.0	(0.03)	255	(77.7)
11	170.0	(4.81)	255	(77.7)
12	117.1	(3.32)	255	(77.7)
13	117.1	(3.32)	255	(77.7)
14	92.4	(2.62)	255	(77.7)
15	92.4	(2.62)	255	(77.7)
16	170.0	(4.81)	255	(77.7)
17	57.5	(1.63)	273	(83.2)
18	117.1	(3.32)	255	(77.7)
19	117.1	(3.32)	255	(77.7)
20	170.0	(4.81)	255	(77.7)

used to code the possible sizes for each of the 21 duplicate pipes, and an additional four bits were used to code the penalty factor. Thus a total of 88 binary bits were used for the application of the self-adaptive boundary genetic algorithm. To investigate the performance of the self-adaptive boundary search strategy, a maximum era size of 10 and a population size of 200 were used in the fast messy GA optimization. The Hazen-Williams equation used for solving this problem is given as (Dandy et al. 1996):

$$\boldsymbol{h}_{\mathrm{f}} = 4.729\boldsymbol{L}\left(\frac{\boldsymbol{Q}}{\boldsymbol{C}}\right)^{1.852}\boldsymbol{D}^{-4.8704} \quad (3)$$

where L is the length of a pipe; Q is the flow rate of a pipe, C designates the Hazen-Williams coefficient and D is the internal diameter of the pipe.

Results Comparison

The New York city water tunnels problem for the case study presented in this paper has been solved by using the fast messy GA both with and without the self-adaptive boundary search. For the fmGA without the self-adaptive boundary search, four different penalty factors of $2.0, $5.0, $7.0 and $11.0 million per ft ($6.6, $16.4, $23.0 and $36.1 million per meter) of hydraulic grade deficit have been used. The penalty factor is treated as a constant over one GA run. The self-adaptive boundary search has been investigated by using three initial penalty factor ranges including [1.0, 10.0], [1.0, 50.0] and [0.2, 10.0] million per ft ([3.3, 32.8], [3.3, 164.0] and [0.7, 32.8] million per meter) of HGL deficit. The fast messy genetic algorithm optimization identified the same critical nodes (nodes 16, 17 and 19) as many other studies in literature. The results obtained by both approaches with and without the self-adaptive penalty approach are compared in Table 4. All the other nodal hydraulic grades are greater than the minimum required hydraulic grades. The pipe sizes for the optimal or near optimal solutions are given in Table 5.

Table 4 Hydraulic Grades and Pressure Head Excesses at Critical Nodes for the New York Tunnel Problem by Genetic Algorithm Optimization with Constant and Self-adaptive Penalty

Node	Minimum Required HGL	Actual HGL (ft and (m)) and Pressure Head Excesses (ft and (m)) for fmGA with Constant Penalty Factors million/ft (million per meter)				Actual HGL (ft and (m)) and Pressure Head Excesses (ft and (m)) for fmGA with Self-Adaptive Penalty Factors million/ft (million per meter)		
ID	ft (m)	2.0 (6.6)	5.0 (16.4)	7.0 (23.0)	11.0 (36.1)	[1.0, 10.0] ([3.3, 32.8])	[1.0, 50.0] ([3.3, 164.0])	[0.2, 10.0] ([0.7, 32.8])
16	260.00	259.15	259.84	261.29	260.40	260.53	260.53	260.53
	(79.25)	(78.99)	(79.20)	(79.64)	(79.37)	(79.41)	(79.41)	(79.41)
	Excess =	-0.85*	-0.16	1.28	0.39	0.52	0.52	0.52
		(-0.26)	(-0.05)	(0.39)	(0.12)	(0.16)	(0.16)	(0.16)
17	272.80	271.46	272.64	272.80	272.83	272.87	272.87	272.87
	(83.15)	(82.74)	(83.10)	(83.15)	(83.16)	(83.17)	(83.17)	(83.17)
	Excess =	-1.35	-0.16	0.00	0.07	0.07	0.07	0.07
		(-0.41)	(-0.05)	(0.00)	(0.02)	(0.02)	(0.02)	(0.02)
19	254.99	254.23	254.82	255.48	254.99	255.71	255.71	255.71
	(77.72)	(77.49)	(77.67)	(77.87)	(77.72)	(77.94)	(77.94)	(77.94)
	Excess =	-0.75	-0.16	0.46	0.00	0.72	0.72	0.72
		(-0.23)	(-0.05)	(0.14)	(0.00)	(0.22)	(0.22)	(0.22)
Cost ($million)		32.33	37.62	39.42	39.69	38.80	38.80	38.80
Achieved at Evaluation No.		15,500	44,500	21,200	18,800	22,500	22,500	51,400

*A negative value of the pressure head excess means a constraint violation.

Table 5 Comparison of Optimal Parallel Pipe Sizes for the New York City Tunnel Problem by Fast Messy Genetic Algorithm with and without Self-adaptive Boundary Search

Pipe ID	Dandy et al. (1996)		Self-adaptive Boundary fmGA		fmGA with Constant Penalty Factors million per ft (million per meter)							
					2.0	**(6.6)**	**5.0**	**(16.4)**	**7.0**	**(23.0)**	**11.0**	**(36.1)**
	Diameter		Diameter		Diameter		Diameter		Diameter		Diameter	
	in	(mm)	in	(mm)	in	(mm)	in	(mm)	in	(mm)	in	(mm)
1	0	(0)	0	(0)	0	(0)	0	(0)	0	(0)	72	(1830)
7	0	(0)	0	(0)	0	(0)	120	(3050)	108	(2740)	108	(2740)
15	120	(3050)	120	(3050)	0	(0)	0	(0)	0	(0)	0	(0)
16	84	(2130)	84	(2130)	84	(2130)	96	(2440)	108	(2740)	96	(2440)
17	96	(2440)	96	(2440)	96	(2440)	96	(2440)	108	(2740)	96	(2440)
18	84	(2130)	84	(2130)	84	(2130)	84	(2130)	72	(1830)	84	(2130)
19	72	(1830)	72	(1830)	72	(1830)	72	(1830)	60	(1520)	72	(1830)
21	72	(1830)	72	(1830)	72	(1830)	72	(1830)	84	(2130)	72	(1830)
Cost ($ million)	38.80		38.80		32.33*		37.62*		39.42		39.69	
Average Evaluations	125,000		30,000		15,500		44,500		21,200		18,800	

*An infeasible solution.

Table 6 Comparison of Hydraulic Grades of Optimal Solutions for the New York City Water Tunnels Problem

Node ID	Minimum Grades		Lippai et al. (1999)				fmGA Boundary Search			
			HGL		Excess		HGL		Excess	
	ft	(m)	ft	(m)	ft	(m)	ft	(m)	ft	(m)
2	255.0	(77.7)	294.2	(89.7)	39.2	(12.0)	294.6	(89.8)	39.6	(12.1)
3	255.0	(77.7)	286.2	(87.2)	31.2	(9.5)	287.2	(87.5)	32.2	(9.8)
4	255.0	(77.7)	283.8	(86.5)	28.8	(8.8)	285.0	(86.9)	30.1	(9.2)
5	255.0	(77.7)	281.8	(85.9)	26.8	(8.2)	283.2	(86.3)	28.2	(8.6)
6	255.0	(77.7)	280.2	(85.4)	25.1	(7.7)	281.7	(85.9)	26.7	(8.2)
7	255.0	(77.7)	277.6	(84.6)	22.6	(6.9)	279.5	(85.2)	24.5	(7.5)
8	255.0	(77.7)	276.4	(84.2)	21.4	(6.5)	276.4	(84.2)	21.4	(6.5)
9	255.0	(77.7)	273.6	(83.4)	18.5	(5.7)	274.2	(83.6)	19.2	(5.9)
10	255.0	(77.7)	273.5	(83.4)	18.5	(5.6)	274.2	(83.6)	19.2	(5.8)
11	255.0	(77.7)	273.7	(83.4)	18.6	(5.7)	274.4	(83.6)	19.3	(5.9)
12	255.0	(77.7)	274.9	(83.8)	19.9	(6.1)	275.8	(84.1)	20.8	(6.3)
13	255.0	(77.7)	277.9	(84.7)	22.9	(7.0)	279.0	(85.0)	24.0	(7.3)
14	255.0	(77.7)	285.5	(87.0)	30.4	(9.3)	287.0	(87.5)	32.0	(9.8)
15	255.0	(77.7)	293.3	(89.4)	38.3	(11.7)	295.3	(90.0)	40.3	(12.3)
16	260.0	(79.3)	259.8	(79.2)	**-0.2***	(-0.1)	260.5	(79.4)	0.5	(0.2)
17	272.8	(83.2)	272.6	(83.1)	**-0.2**	(-0.1)	272.8	(83.2)	0.0	(0.0)
18	255.0	(77.7)	260.9	(79.5)	5.9	(1.8)	261.8	(79.8)	6.8	(2.1)
19	255.0	(77.7)	254.8	(77.7)	**-0.2**	(-0.1)	255.7	(77.9)	0.7	(0.2)
20	255.0	(77.7)	260.5	(79.4)	5.5	(1.7)	261.2	(79.6)	6.1	(1.9)

*A negative value of the pressure head excess means a constraint violation.

As shown in Tables 4 and 5, the results obtained by the fast messy GA without the self-adaptive boundary search indicate that a constant penalty factor of about $7.0 million/ft ($23.0 million/m) is the best value. For the larger penalty factor of $11.0 million/ft (36.1 million/m), the genetic algorithm search appears to be trapped at a local optimal solution, that is not the least cost solution. Conversely, for smaller penalty factors such as $2,000,000/ft ($6.6 million/m), the optimal solution was found to be infeasible as shown in Table 4. Thus the value of a penalty factor has a great impact on the efficacy of the solution technique. A reliable solution cannot be guaranteed for the optimization of water distribution systems by using GA optimization with a constant penalty factor.

On the contrary, the self-adaptive boundary search genetic algorithm has been shown consistently effective and efficient at reaching the least cost solution as shown in Table 4. Initially, a penalty factor range of [1.0, 10.0] million/ft ([3.3, 32.8] million/m) was used to test the boundary GA optimization strategy. The optimal solution has been achieved after 22,508 evaluations. The other two runs, one with a scaled-up upper bound of the penalty factor range [1.0, 50.0] million/ft ([3.3, 164.0] million/m) and the other with the scaled-down lower bound of the penalty factor range [0.2, 10.0] million/ft ([0.7, 32.8] million/m), have also been tested. The results show that the same optimal solution of $38.80 million has been achieved for all three different penalty factor ranges. The optimal solution is not sensitive to the initial values of the penalty factor ranges. Thus the self-adaptive boundary search provides a more effective approach in contrast to the constant penalty factor.

The optimal solution of $38.8 million has been found by the self-adaptive boundary search after an average of 30,000 evaluations. The same optimal solution was obtained by Dandy et al. (1996) after approximately 125,000 evaluations. More recently, Lippai et al (1999) integrated Evolver 4.0 (*Evolver* 1998), a commercial genetic algorithm optimizer, with EPANET and solved the New York city water tunnels problem with a cost of $37.8 million after an average of 50,000 evaluations. The optimal solution of $37.8 million, however, has been found to be infeasible as shown in Table 6 when using the Hazen-Williams equation Eq.(3). The self-adaptive boundary search approach is shown to be more efficient than any of previously reported genetic based methods for the optimization of water distribution systems.

Convergence Behavior for a Constant Penalty Factor

For the fast messy GA optimization with a constant penalty factor, the GA improves the search process very quickly in the early stages of the optimization and then slowly converges to a final solution. The feasibility and optimality of the final solution, however, depend on the penalty factor value being used for the GA run. Figure 3 shows convergence rates of the GA optimization for three different constant penalty factor values for solving the New York city water tunnels problem. Figure 4 shows the patterns of the population feasibility traced into the GA optimization process using different penalty factors.

As shown in Figure 4(a), a large penalty factor distorts the objective function in such a way that the GA favors solutions within the feasible region. Initially, there is a variation of population feasibility due to the random initialization and gene filtering. After about 12,000 evaluations, the ratio of feasible solutions to population size, is greater than 0.8 as the GA optimization proceeds. The population feasibility is consistently high. Thus the GA appears to be trapped within a feasible region, which leads to a non-optimal solution. With small penalty factors such as 2.0 and 5.0 million/ft (6.6 and 16.4 million/m), the GA favors solutions within the infeasible region. As shown in Figure 4(b), the population

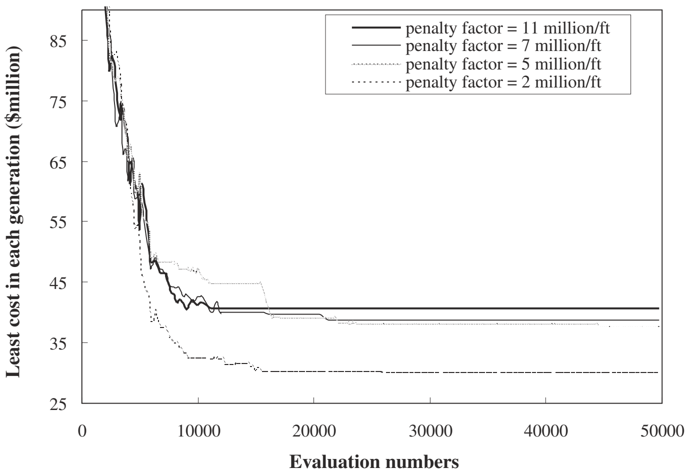
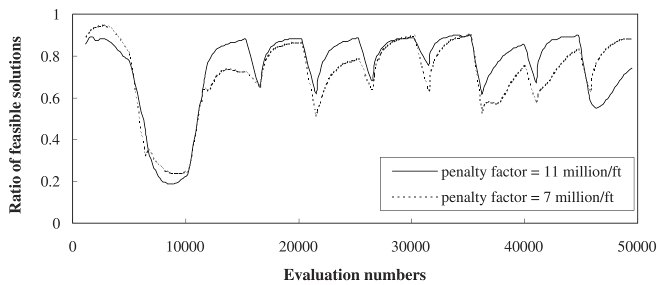
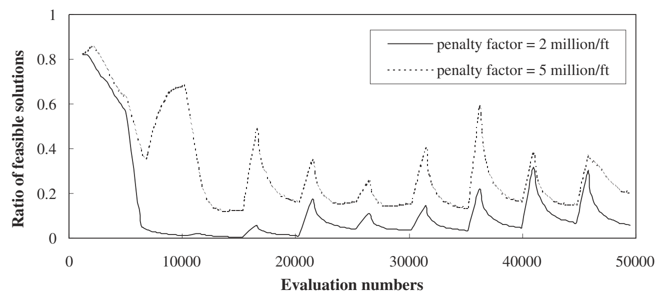


Fig. 3 Convergence Behavior of Genetic Algorithm Optimization of New York Tunnel Problem by Using Constant Penalty Factors



(a) Feasibility of GA population with penalty factor of 7 and 11 million/ft (23.0 and 36.1 million/m)



(b) Feasibility of GA population with penalty factor of 2 and 5 million/ft (6.6 and 16.4 million/m

Fig. 4 Population Feasibility Trace of Genetic Algorithm Optimization of New York City Tunnel Problem by Using Constant Penalty Factors feasibility is consistently low and the ratio of feasible solutions to population size is less than 0.2 for some of the optimization process. The GA consequently reaches an infeasible solution as shown in Figure 3 and Table 4. Thus a constant penalty factor approach requires a well-tuned penalty factor to achieve the least cost solution for this particular case study.

Convergence Behavior of the Self-Adaptive Boundary Search

The self-adaptive boundary optimization has been effective and efficient at searching for the optimal solution for the New York city water tunnels problem. Two different convergence patterns have emerged as shown in Figure 5, 6 and 7. Figure 5 compares the convergence for each of the three initial penalty factor ranges, in terms of the least cost solution in each generation versus the cumulative number of evaluations. When the initial upper limit of the penalty factor is \$10.0 million/ft (\$32.8 million/m) the GA overshoots resulting in network costs that are far below the optimal network of \$38.80 million. Adjustment of the penalty factor range then occurs bringing the GA back towards the minimum cost network from below. In contrast, when the initial upper limit of penalty factor is \$50.0 million/ft (\$164.0 million/m) the GA converges on the optimal solution from above. Figure 6 compares the average penalty factor in each generation versus the total number of generations while Figure 7 illustrates the cumulative proportion of feasible solutions that have occurred since the last penalty factor was set or adjusted.

The self-adaptive boundary search approach co-evolves the penalty factors along with the solutions, and also self-adapts the range of the penalty factors after every 20 generations (or 5,000 evaluations). The self-adaptation of the penalty factor ranges was controlled with the minimum feasibility ratio φ_{min} is 30% and the maximum feasibility ratio φ_{max} is 80%. For the initial penalty factor range [1.0, 10.0] million/ft ([3.3, 32.8] million/m), the GA search started with a highly feasible initial population as shown in Figure 7(a). The feasibility ratio was about 0.9 and the average penalty factor was about 5.0 million/ft (\$16.4 million/m) as shown in Figure 6. After 20 generations (or 5,000 evaluations) the

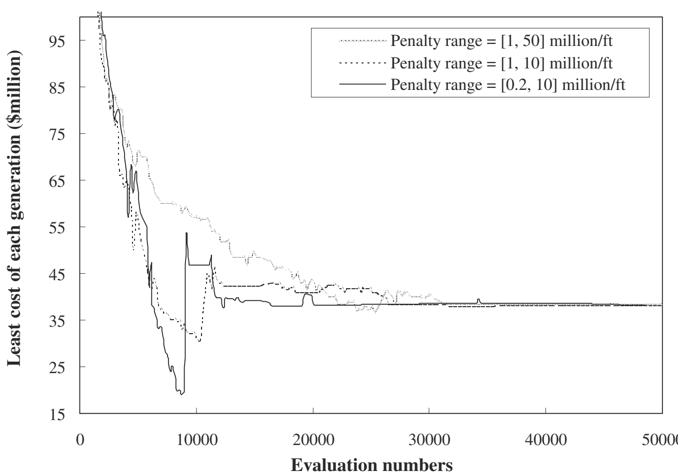


Fig. 5 Convergence Behavior of Genetic Algorithm Optimization of New York Tunnel Problem by Using the Self-adaptive Boundary Search Strategy

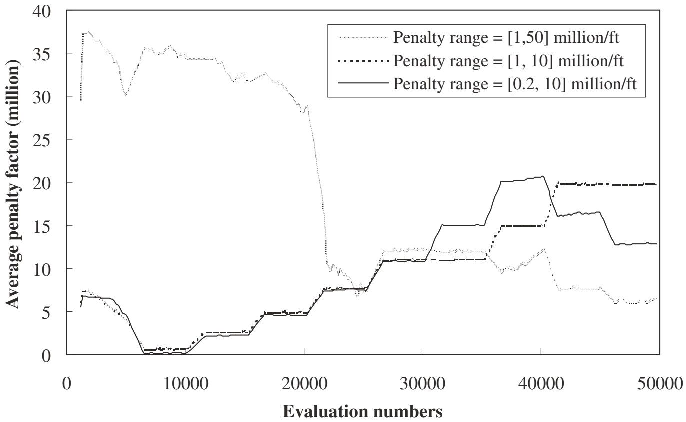
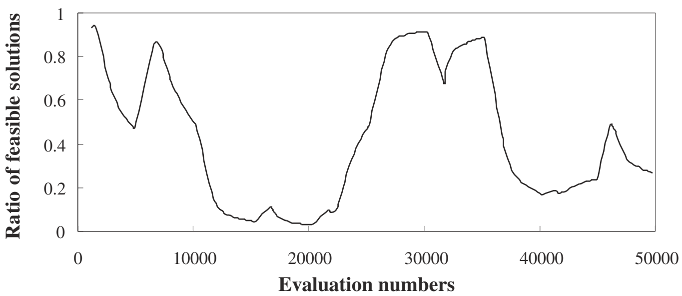
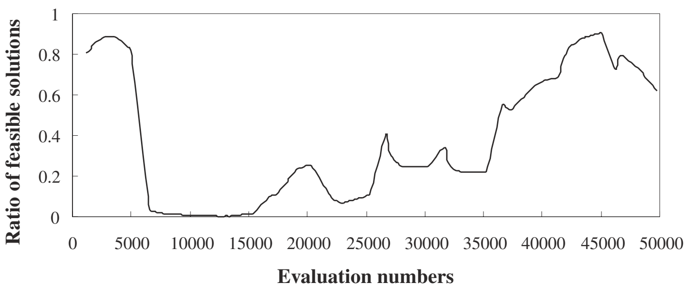


Fig. 6 Average Penalty Factor of Self-adaptive Genetic Algorithm Boundary Search for Optimization of New York Tunnel Problem

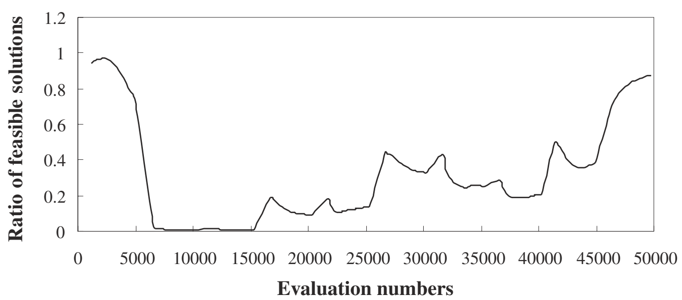
upper and lower bounds of the penalty factor were reduced by 20% of the lower bound of 1.0 million/ft (3.3 million/m) based on the heuristic rule introduced earlier. Thus the average value of



(a) penalty factor range = [1.0, 10.0] million/ft ([3.3, 32.8] million/m)



(b) penalty factor range = [0.2, 10.0] million/ft ([0.7, 32.8] million/m)



(c) penalty factor range = [1.0, 50.0] million/ft ([3.3, 164.0] million/m)

Fig. 7 Population Feasibility Trace of Self-adaptive Genetic Algorithm Boundary Search for Optimization of New York Tunnel Problem penalty factors decreased as shown in Figure 6, however, the penalty factor was reduced too much to enable the GA to penalize and trade off infeasible solutions. Consequently, the GA selected infeasible solutions and the population feasibility became very low as shown in Figure 7(a). The cost of the best solution, at this stage, was dramatically reduced as shown in Figure 5, but the solution was infeasible. The penalty factor range was then adjusted by the self-adaptive penalty rule. The lower and upper bounds of penalty factors were now increased by 20% of the upper bound of the previous penalty factor range. The result was an increase in population feasibility that enabled the GA to disregard infeasible solutions. Thus the GA search was then adapted across the constraint boundary towards the feasible solution region. As the penalty factor was adapted according to population feasibility, the GA search reached the optimal solution.

For the initial penalty factor range [0.2, 10.0] million/ft ([0.7, 32.8] million/m), the GA search followed the same pattern as the GA run using the initial penalty range [1.0, 10.0] million/ft ([3.3, 32.8] million/m). By using the self-adaptive penalty rule, the GA has been adapted towards the boundary of hydraulic pressure constraints. The same optimal solution was found in this case of the scaled-down (for the lower limit) penalty factor range.

A different convergence pattern was observed for the self-adaptive boundary search when the upper limit of the initial range of penalty factors is scaled-up. For the scaled-up penalty factor range of [1.0, 50.0] million/ft ([3.3, 164.0] million/m), the GA search started with an average penalty factor of about 30.0 million/ft (98.4 million/m) as shown in Figure 6. Due to a high initial population feasibility, the penalty factor bounds were reduced until the feasibility ratio was less than the minimum feasibility ratio. Then the population feasibility increased as the penalty factor was adapted to be slightly greater and greater as shown in Figure 6. The best solution (see Figure 5) is improved while the penalty factor is increased according to the self-adaptive penalty rule. The optimal solution of $38.80 million has also been found for the case of the scaled-up penalty factor range.

The convergence patterns observed for the New York city water tunnels system indicate that the self-adaptive boundary search strategy is able to adapt a GA to search the boundary of the solution space. Different initial ranges of a penalty factor have been adapted by following the self-adaptive penalty rule. A population is either maintained around or adapted towards a constraint boundary to reach the optimal solution. The optimal solution does not seem to be sensitive to the initial range of a penalty factor. The self-adaptive boundary search strategy has been effective not only at adjusting the penalty factor and consequently adapting the search process, but also improved the efficiency and effectiveness of the GA optimization.

Conclusions

A self-adaptive boundary search technique has been developed within a framework of genetic algorithms for solving constrained non-linear optimization problems. For constrained non-linear optimization problems, the optimal solution is found at the boundary between the feasible and infeasible regions of a solution space. Conventional genetic or evolutionary algorithms using a constant penalty factor are not effective at searching the boundary of a solution space in terms of being able to consistently find low cost feasible solutions. A constant penalty factor results in either high or low population feasibility depending on its magnitude. To achieve the optimal solution at the boundary, the penalty factor needs to be carefully tuned for a specific case study. The approach of the self-adaptive boundary search as presented in this paper has successfully preserved a GA population around or adapted a GA population towards the boundary between feasible and infeasible regions of a solution space. The adaptation of a GA population is achieved by co-evolving a penalty factor and automatically adjusting the penalty factor range. The penalty factor for one solution is coded as a sub-string that is included within a solution string. A string that encodes a penalty factor and produces better offspring survives longer. The preferred penalty factor spreads through the population and co-evolves over the generations. Meanwhile, the penalty factor range is adjusted according to population feasibility. A heuristic rule has been developed for adapting the penalty factor range in such a way that a population of water network designs is forced towards the boundary of feasible and infeasible regions. Thus the population can be either maintained around or adapted towards the constraint boundary.

In this paper, a self-adaptive boundary search has been implemented within the fast messy GA and applied to the optimization of water distribution systems. The New York city water tunnels problem has been used as the case study. The results obtained by using the boundary GA search strategy have been compared with the conventional GA optimization using constant penalty factors. It has been shown that the solution by the conventional approach is sensitive to the value of a penalty factor. Too large a penalty factor value enables the GA to select feasible solutions, but the search is trapped at a local optimum within a feasible region, which is not the least cost solution. Too small a penalty factor value forces the GA to select the solutions outside the feasible region that eventually leads the search to an infeasible solution. The boundary GA search strategy has been shown to be effective at adapting the feasibility of GA population within a wide range of penalty factors. It automatically adjusts penalty factor ranges and co-evolves penalty factors during the GA optimization procedure. Thus it guarantees a reliable solution and improves the efficacy of the GA optimization of water distribution systems. Meanwhile, it would be a fruitful research to investigate the performance of the boundary search strategy for more complex water distribution systems. The approach should also have broader applications beyond the optimization of water distribution systems to other types of constrained non-linear optimization problems.

Acknowledgements

This work was carried out during the first author's Ph.D research at University of Adelaide. The writers wish to thank University of Adelaide for the support of the research project.

References

ALPEROVITS, E. & SHAMIR, U. (1977). 'Design of water distribution systems.' *Water Resources Research.* Vol. 13, No. 6, 885-900.

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.

BHAVE, P.R. (1985). 'Optimal expansion of water distribution systems,' *J. Environmental Engineering,* ASCE, 111(2), 177-197.

BOX, M. J. (1965). 'A new method of constrained optimization and a comparison with other methods.' *J. Computer,* Vol. 8, pp. 42.

CUNHA, M. C. and SOUSA, J. (1999). 'Water Distribution System Design Optimization: Simulated Annealing Approach,' *Journal of Water Resources Planning and Management*, Vol. 125, No. 4, 215-221.

DANDY, G. C., SIMPSON, A. R. & MURPHY, L. J. (1996). 'An improved genetic algorithm for pipe network optimization,' *Water Resources Research,* Vol. 32, No. 2, 449-458.

DUAN, N., MAYS, L. W. & 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. & BEN-TAL, A. (1994). 'Optimal design of water distribution networks.' *Water Resources Research,* 30(9), 2637-2646.

Evolver with bestFit and @Risk, the genetic algorithm solver ___ Version 4.0: Guid to Evolver, Evolver's developer's kit. (1998). Palisade Corp., Newfield, N. Y.

FUJIWARA, O., JENCHAIMAHAKOON & EDIRISINGHE N. C. P., (1987). 'A modified linear programming gradient method for optimal design of looped water distribution networks', *Water Resources Research*, 23(6), 977-982.

FUJIWARA, O. & 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.

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 of Illinois at Urbana-Champaign, Urbana, IL 61801, USA.

HALHAL, D, WALTERS, G. A., OUAZAR, D. & 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.

KIM, J. H. & 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. & MAYS, L.W. (1989). 'Optimization Model for Water Distribution System Design'. *J. Hydr. Engrg.*, ASCE, Vol. 115, No. 10, 1401-1418.

LIPPAI, I, HEANEY, J. P. & LAGUNA M. (1999). 'Robust Water System Design with Commercial Intelligent Search Optimizers'. *J. of Computing in Civil Engrg*, ASCE, Vol. 13, No. 3, 135-143.

LOGANATHAN, G. V. GREENE, J.J. & 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.

MICHALEWICZ, Z. & SCHOENAUER, M. (1996). 'Evolutionary algorithms for constrained parameter optimization problems.' *Evolutionary Computation,* 4(1), 1-32, MIT Press.

MORGAN, D. R., & GOULTER, I. C. (1985). 'Optimal urban water distribution system.' *Water Resources Research,* 21(5), 642-652.

MURPHY, L.J. and SIMPSON, A. R. (1992). 'Pipeline optimization using genetic algorithms,' *Research Report No. R 93,* Dept. of Civil and Envir. Eng., The University of Adelaide, Australia.

QUINDRY, G. E., BRILL, E.D., & LIEBMAN, J.C. (1981). 'Optimization of looped water distribution systems,' *J. Environmental Engineering Division,* ASCE, 107(EE4), 665-679.

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.

SAVIC, D.A. & 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. & 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.

SCHOENAUER, M., & MICHALEWICZ, Z. (1996). 'Evolutionary computation at the edge of feasibility.' In W. Ebeling & H.-M. Voigt (Eds.), *Proceeding of the Fourth Conference on Parallel Problem Solving from Nature,* Berlin: Springer-Verlag.

SIMPSON, A. R., DANDY, G. C. & 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.

SU, Y. C., MAYS, L.W., DUAN, N., & LANSEY, K.E. (1987). 'Reliability-based optimization model for water distribution system.' *J. Hydr. Engrg.,* ASCE, 114(12), 1539-1556.

WU, Z. Y. & SIMPSON, A. R. (1996). 'Messy genetic algorithms for optimal design of water distribution systems.' *Research Report,* No. 140, Department of Civil & Environmental Engineering, University of Adelaide.

Wu, Z. Y. & 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.

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.

Zhu, B. F., Li, Z. M. and Zhang, B. C. (1984). *Structural optimal design: theory and applications.* Hydro-electrical Press, Beijing, China, (in Chinese).