ELSEVIER

Urban Water 1 (1999) 23–38

UrbanWater

www.elsevier.com/locate/urbwat

Improved design of "Anytown" distribution network using structured messy genetic algorithms

Godfrey A. Walters [a,*], Driss Halhal [b], Dragan Savic [a], Driss Ouazar [c]

[a] *School of Engineering and Computer Science, University of Exeter, Exeter, Devon EX4 4QF, UK*
[b] *Water and Electricity Distribution Co.(RAID), 5 Rue Okba Ibn Naffiy, BP 286, Tangier, Morocco*
[c] *Hydraulic Systems Analysis Laboratory, Mohammadia School of Engineers (EMI), BP 765, Agdal, Rabat, Morocco*

Received 12 January 1999; received in revised form 14 July 1999; accepted 20 July 1999

Abstract

The recently introduced Structured Messy Genetic Algorithm model for optimising water distribution network rehabilitation is expanded to include not only pipe rehabilitation decisions but also pumping installations and storage tanks as variables. The formulation of the model is detailed with two approaches presented for handling the design of storage within the system. The application of the model to the benchmark "Anytown" problem is used as an example of its capabilities, with cheaper designs being produced than any previously published. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Water distribution; Genetic algorithms; Multi-objective; Optimisation; Rehabilitation; Networks; Systems; Design

1. Introduction

Water distribution networks use interconnected elements such as pipes, pumps and tanks (service reservoirs) to convey treated water from one or more sources to customers spread over a wide area. The capital invested in water distribution networks represents a large proportion of a water company's assets. Depending on the system, the annual energy costs may also be considerable.

Usually demand for water grows over the years, due to population expansion, industrial and commercial growth and increased per capita consumption. Along with this, the physical condition of the elements continually worsens, with, for example, pipes leaking, breaking and corroding. Such deterioration leads to loss of water, increased head losses, increased pumping costs, water quality problems and increased emergency maintenance bills. Consumer complaints also steadily rise.

Periodically, therefore, networks require rehabilitation in the form of reinforcement, expansion and reconditioning to maintain or improve the level of service required by society. However, the rehabilitation of an existing water distribution system is a complex task if it is to be completed in the most effective and economic manner. It necessitates a systematic and thorough approach, backed up by skillful engineering judgement, and significant capital resources. The examination and evaluation of design alternatives is an area in which optimisation models can play an important role, particularly when finance is limited and the problems are large.

This paper presents a model for solving water distribution network design and upgrading problems in an optimal fashion, including the addition of new pipes, reconditioning and replacement of existing pipes, siting and sizing of new tanks and provision of new pumps. The method is based on the Structured Messy Genetic Algorithm (SMGA), introduced by the authors (Halhal, Walters, Ouazar & Savic, 1997) for gravity flow systems, and extended here to include storage and pumping in the decision process.

The problem is posed and solved as a multi-objective one, with minimum cost and maximum benefit the twin objectives. The reasons for this are twofold. First, when planning network improvements, there is usually a requirement to investigate the trade-off to be made between the cost of the project and the benefits predicted. A multi-objective approach allows a range of schemes to be produced, including both high cost, high benefit schemes as well as low cost, low benefit options, each

* Corresponding author. Fax: +44-1392-217965.
E-mail address: g.a.walters@exeter.ac.uk (G.A. Walters)

1462-0758/99/$ - see front matter
PII: S1462-0758(99)00005-9

scheme delivering the maximum benefit for the cost involved. The second reason is one of computational efficiency. SMGA is an evolutionary design process within which a diverse range of "non-dominated" solutions is developed and maintained, using the techniques of Pareto optimality ranking and fitness sharing. As demonstrated in the earlier paper, (Halhal et al., 1997), SMGA works very efficiently for multi-objective optimisation (MOO) due to the incremental development of complex solutions from simple but cost-effective partial solutions to the problem.

The alternative approach, often adopted, is to pose the problem more narrowly as minimising the cost of a system subject to a set of performance constraints. Satisfying the performance constraints is then equivalent to achieving a specific level of benefit. Conventionally, this formulation can be handled by incorporating the constraints into a penalty function, which is added to the cost to form a single objective. A weighting factor must be assigned to the penalty function large enough to prevent the final solution from being infeasible and small enough not to prevent adequate exploration of the search space. An adaptive weighting procedure is often used with a conventional GA and other search techniques to give increasing penalties as the search converges on a solution. Use of MOO avoids the requirement for the penalty function and its associated weighting factor, as cost and benefit (system performance) are independently assessed. The SMGA MOO methodology allows a more flexible approach, which is also suitable for handling other forms of multi-objective criteria.

As an example, the method is used to examine the "Anytown" water distribution system, which was set up by Walski et al. (1987) as a realistic benchmark on which to compare and test network optimisation software, and has features and problems typical of those found in many real systems. The "Anytown" problem was originally tackled by the participants at the "Battle of the Network Models" workshop, and has since been examined by Murphy, Dandy and Simpson (1994). All participants in the original workshop used optimisation models to size the piping system while manually choosing the location and size of tanks. The feasibility of solutions was checked using a simulation of the system operation over 24 h. The differences between the various methods used lie essentially in the pipe optimisation models, which were based on linear programming, partial enumeration or non-linear programming techniques. No attempt was made to optimise the provision of tanks or pumps, except by use of expert judgement and experience.

Murphy et al. (1994) later obtained a better solution to the problem using a Standard Genetic Algorithm (SGA), which was able to handle the tanks and pumps as additional design variables. To obtain a solution, network deficiencies were incorporated into the objective function (cost) as penalties, using a set of weightings, one for each type of constraint violation (e.g. pressure deficiency, unbalanced tank flows). Selection of the most effective weightings required a number of trial runs.

The performance of the SMGA developed in the present paper is shown to be significantly better than that of Murphy et al.'s SGA, even though the problem is posed in almost identical terms. It is further argued that greater advantages would become apparent for larger, more complex problems.

2. The approach

The basic SMGA MOO procedure adopted is similar to that used for the gravity distribution system problem tackled by Halhal et al. (1997), as this has already shown its efficiency in tackling large and complex systems. However, the extension of the optimisation to include the sizing and operation of pumps and tanks adds a great deal of complexity to the problem, and requires significant modifications to the model.

The presence of pumps requires that both the design and the operation of the network should be considered in the optimisation. The method developed allows the selection of both the pump type (specification and capacity) and the number of pumps to be installed for each new or upgraded pumping station. It also allows the pumps that are operational to be defined for each period of the day in which different loading conditions are used.

Storage tanks are included in a system to act as buffers, filling at low demand periods of the day and emptying at peak demand periods, thereby reducing both pumping and pipeline costs. They also provide the required reserve for fire fighting and emergency conditions. Including tank storage in the optimisation requires simulation of the filling and emptying of the tanks through the daily cycle of demands to ensure feasible operation. Accurate simulation of the system's response to the variations in demand over a day using, say, hourly time steps, is at present too time consuming for use in an optimal design program which requires evaluation of a very large number of trial designs. Hence a coarse simulation splitting the day into four representative periods was used for modelling the system performance during the optimisation. A full simulation using 24 h time steps on the maximum-demand day was then used to check the feasibility of the final designs, as specified in the original "Anytown" problem.

It is assumed that the hydraulic design of storage within a network can be adequately defined by specifying the number and location of all tanks, and, for each tank, the volume, and the maximum and minimum operating levels. Hence it is these variables that are determined in the optimisation process. Furthermore, for an optimal solution, all tanks should empty and fill over their operational ranges during the specified maximum-demand day, leaving the specified emergency volumes untouched. Two different approaches were explored for achieving fully balanced tank designs during the optimisation process.

In the first approach, which is similar to that adopted by Murphy et al. (1994), the locations and volumes of new tanks are specified as design variables. The maximum and minimum operating levels are treated as dependent variables, whose values are determined by the pressures required to provide the network inflows and outflows necessary for filling and emptying the tanks. The procedure works as follows:

- Within the SMGA iteration process, trial values of tank locations and volumes (along with pump and pipe variables) are produced.
- For each of the four periods of the day, the total flow for the system into or out of storage is calculated according to whether the demand for that period is less than or greater than the average for the day.
- The flows into and out of each tank (the "demand balancing" flows) are calculated for each period by allocating the system inflows and outflows to the tanks in proportion to their trial volumes. This constrains all tanks to empty and fill in a similar manner irrespective of their volumes.
- The tanks are then treated as nodes with known demands (tank demand nodes).
- The heads at the tank demand nodes are then determined by steady state simulation of the network for each of the four demand periods.
- For each tank demand node, the maximum and minimum head identified over the four time periods are interpreted as the maximum and minimum operating levels for the corresponding tank. As the trial tank volumes are known, this determines the required plan areas of the tanks.
- Initially, there will be a mismatch between the tank heads calculated as above for each of the four time periods and the corresponding levels in the tanks that result from volume changes due to the inflows and outflows. To ensure that fully compatible designs evolve in the GA, the tank levels are recalculated over the sequence of four time periods, using the tank dimensions, an initial level and the inflows and outflows previously determined. The magnitude of the mismatch is then used as a penalty, which tends to zero as fully compatible designs evolve. This penalty is referred to as the tank operating level difference (TLD).

In the second method, location, volume, maximum operating level and minimum operating level are treated as independent design variables for the tanks, thereby defining completely the design of the storage. However, storage will initially not match the requirements of the network. To prevent reservoirs overfilling or dropping below safe operating levels, it is necessary to devise a penalty to drive the GA towards fully feasible, balanced solutions. The procedure now works as follows:

- Within the SMGA iteration, trial values of tank designs (along with pump and pipeline variables) are produced.
- For each tank, the "demand balancing flows" for each of the four time periods are derived as in the first method.
- For each tank, the volume changes and hence the water levels are determined over the sequence of four time periods.
- The flows in the system are then determined for each time period by steady state hydraulic analyses with fixed water level tanks.
- The flows into or out of each tank are compared to the demand balancing flows previously specified. The accumulated difference between the two sets of flows, for the four periods of the day, is used as a penalty in the evaluation of the solution, and will tend to zero as fully balanced designs evolve. This penalty is referred to as the tank flow difference (TFD).

With both approaches, the outcome is a penalty term (TLD or TFD), that gives an approximate measure of the lack of hydraulic balance within the system under maximum design conditions.

The response of the system to specified peak and emergency flow conditions must now also be considered. A steady state hydraulic network solver is used to determine the head at all the system nodes for all specified "snapshot" design conditions, and the accumulated sum of the nodal pressure shortfalls is used in the evaluation of the benefit objective function. This value is referred to as the nodal pressure shortfall (NPS). A similar term based on pressure excesses could also be derived, but was not incorporated into this work.

The optimisation process so far described will tend to produce tanks with the minimum volume required for balancing demand fluctuations in the system, and may lead to designs that have insufficient emergency storage capacity. To force the process to incorporate adequate capacity, the volume of water that should be stored in the system is calculated, allowing for fire and other emergency use, and is then compared with the volume available. The difference is referred to as the storage capacity difference (SCD), and is used as the final measure in the evaluation of the benefit objective.

3. Problem formulation

The problem is to find the optimal design and mode of operation of a pumped water distribution system with storage. The problem is substantially more difficult than for systems without pumps or tanks. It is presented here as a MOO process with the two objectives, to maximise benefit and to minimise cost. The design variables are mainly discrete (pipe diameters, pump numbers, pump status, tank locations), but tank volumes and levels are continuous. The problem can be analytically formulated as follows:

$$\text{Maximise } f(i) = \text{Benefit(i)}$$

and

$$\text{Minimise } F(i) = \text{Cost}(i),$$

where Benefit(i) is the benefit resulting from a solution i and Cost(i) its cost.

The benefit from solution i is determined here as the difference between the predicted deficiencies of the initial, unimproved system and those of solution i. Although expressed here in terms of remedying low pressures and storage shortfalls, other measures of benefit such as network reliability or operational flexibility could be incorporated if required. For convenience the penalty term that quantifies the storage imbalance (TLD or TFD) is incorporated into the benefit formulation as a deficiency, it being a measure of the mismatch between storage allocation and hydraulic requirements. Hence:

$$\begin{aligned}\text{Benefit}(i) \\ &= w_{\text{NPS}}(\text{NPS}_0 - \text{NPS}_i) + w_{\text{SCD}}(\text{SCD}_0 - \text{SCD}_i) \\ &\quad + [w_{\text{TLD}}(\text{TLD}_0 - \text{TLD}_i) \text{ or } w_{\text{TFD}}(\text{TFD}_0 - \text{TFD}_i)],\end{aligned}$$

where the index 0 corresponds to the initial system while the index i corresponds to solution i, and w_{NPS}, w_{SCD}, w_{TLD}, w_{TFD} are dimensional weightings to convert the various deficiency measures to a common basis. The remaining variables in the formula have been explained earlier in the text. The benefit will reach a maximum value when all deficiencies in the system have been met by a proposed design. (In terms of single objective optimisation, all performance constraints are then satisfied.)

The cost of the solution includes the capital costs of pipes, pumps and tanks as well as the present value (PV) of the energy consumed during a specified period.

The twin objectives of maximising benefit and minimising cost will lead to a set of non-dominated solutions spread over a wide range of cost and benefit, as described later.

4. Structured messy genetic algorithms and multi-objective optimisation

4.1. Genetic algorithms

Genetic Algorithms (GAs) are stochastic search procedures based on the evolutionary mechanisms of natural selection and genetics (Holland, 1975). They use the rapid iterative processing ability of computers to simulate the methods by which species adapt themselves to suit their environments. In other words, GAs mimic the very effective optimisation model that has evolved naturally for dealing with large, highly complex systems. Good descriptions of GAs are given by Goldberg (1989) and Michalewicz (1992). A brief summary is presented here for completeness.

A population of random trial solutions to the problem are created, each trial solution being defined by the values of its design variables, which are encoded as a data string (chromosome), usually using a binary coding. Every solution (individual) is evaluated using the objective function, and its "fitness" is determined relative to the entire population. This value determines the probability that an individual will contribute offspring to the next generation of solutions, the offspring being created by a "breeding" process.

The breeding process uses three simple operators: selection, crossover and mutation. Selection produces a pool of individuals with, in effect, multiple copies of fitter members from the old population appearing alongside small numbers of less fit members. From this pool, random selections of "parents" are made, thus biasing the selection of parents towards the fitter members of the old population. Once two parents have been selected, their data codes are rearranged by the crossover process, which breaks the data strings at a random point and crosses the tail ends of the strings over. Two new individuals (offspring) are thus formed, each having characteristics inherited from its parents. The third genetic operator, mutation, changes, with a low probability, randomly selected digits of the offspring's data string. The breeding process continues until a complete new generation has been formed, at which point the old generation is discarded and the process starts a new iteration.

4.2. Structured messy genetic algorithms

The Structured Messy Genetic Algorithm (SMGA) (Halhal et al., 1997) is a type of GA with flexible coding and variable string length. Basically, it uses a progressive evolutionary process, which starts with a population of very short string length members. The string lengths are then increased in subsequent generations alongside improvements to the fitness of the population members, until a predetermined maximum string length is reached. In other words, small partial solutions to a problem are evaluated initially, and are progressively increased in complexity until they become solutions to the full problem. The natural analogy is with the evolution of complex life-forms from single cell organisms.

SMGA adopts two main processes: concatenation and a simple GA. The algorithm starts with either a complete or partial enumeration of single variable decisions, the best of which are retained in an initial population of single digit strings from which building blocks are taken for the creation of subsequent populations of longer strings. For example, in the context of water network rehabilitation, a single digit string would represent the rehabilitation decision for an individual pipe in the network. The concatenation process consists of adding on to the strings of current population members, elements from the initial population, thereby forming a new population with longer strings. The new population then undergoes a GA process for a number of generations until meeting a termination criterion, whereupon another new population of longer strings is generated by the concatenation process.

The cycle continues until no improvement can be achieved after two concatenation steps, or the population strings reach their predetermined maximum length. The flowchart of Fig. 1 outlines the SMGA algorithm.

Many different GA formulations exist, which makes performance comparisons quite difficult. However, in comparison to the conventional SGA, as used, for example, by Murphy et al. (1994), SMGA has two main advantages:

(i) The optimum solution for a rehabilitation or network strengthening problem may be to select only a very small number of pipes or other elements for upgrading, renovation or strengthening, from the hundreds or thousands of elements that may need to be considered in a system. While searching through all the combinations of possibilities, SMGA encodes for a potential solution only the relatively small number of decision variables that are active in that solution. Only the relevant element numbers and the rehabilitation decisions for those elements are stored in a string. In contrast, most conventional SGAs would encode the decisions for all variables, even though most of the decisions would

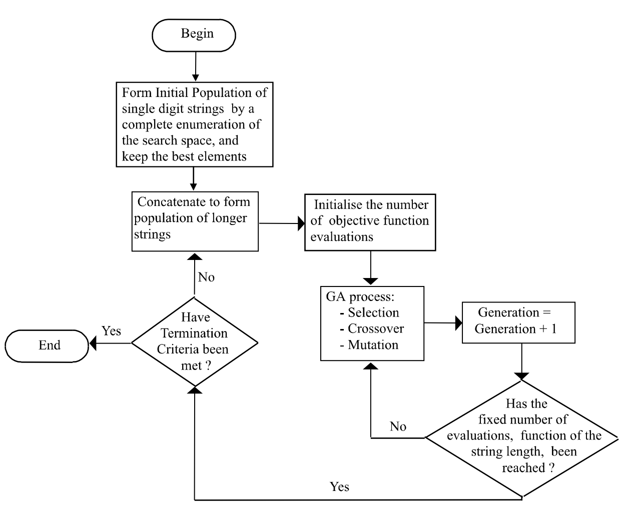

Fig. 1. Flowchart representation of SMGA.

correspond to "do nothing" choices. SMGA explores the search space with strings of small maximum length, thus taking full advantage of the problem structure, and handling easily large and complex water distribution systems using less computing time and memory space than a conventional SGA.

(ii) The search space is reduced compared to an SGA approach that considers all combinations of arcs as candidate solutions. This speeds up the process of finding the optimal or near optimal solutions, thus reducing the CPU time consumption. In fact, when only p arcs among q are considered, and each takes n alternative solutions, the search space contains a number of solutions equal to

$$n^p \frac{q!}{p!(q-p)!},$$

while it contains the much larger n^q possible solutions when the total number q of arcs is considered as in SGA.

4.3. Structured messy genetic algorithm in multi-objective optimisation

4.3.1. Multi-objective optimisation

It has long been recognised that most water system design is truly multi-objective (DeNeufville, Schaake & Stafford, 1971). In particular, water distribution network design involves several, usually competing, objectives such as minimising cost and maximising benefit. The simultaneous optimisation of these objectives is often handled by formulating them into a single criterion using weighting factors, the selection of which is subjective and often requires many adjustments and runs of the optimiser before a satisfactory combination of objectives is found.

In MOO, the different problem objectives are evaluated independently. Rather than producing a unique optimal solution, MOO aims to produce a family of "non-dominated" solutions, known as a Pareto optimal set. For any member of the Pareto optimal set, the solution is optimal in the sense that no improvement can be achieved in one criterion without causing the degradation of at least one of the remaining criteria. For example, in the case of minimising cost and maximising benefit, the Pareto optimal set would be a set of solutions spread, ideally evenly, over a range of feasible costs, each solution having the greatest possible benefit for its particular cost.

GAs are known to be well suited to MOO since the population of individuals they handle can search for multiple solutions in parallel, taking advantage of any possible similarities between them. Some additional and modified GA procedures are used to handle the MOO problem, among them:

Pareto optimality ranking: Introduced by Goldberg (1989), Pareto optimality ranking is a method of assigning fitness values to a population of solutions in such a way that equal probability of selection is given to all non-dominated individuals. The method consists of finding and then assigning rank 1 to the non-dominated individuals in the population, and then removing them from further contention. Rank 2 is assigned to the next set of non-dominated individuals, which are then removed and so on until all population members have been assigned a rank. A single numerical value of fitness is then assigned to all individuals of rank 1, with progressively lower fitness values assigned to the lower ranks. For the present work, the fitness of an individual is taken as numerically equal to the inverse of the rank of that individual.

However, Pareto optimal ranking alone does not guarantee a uniform spread of the Pareto set, and the population may converge to a single solution in the non-dominated set, a phenomenon known as genetic drift. To prevent such a problem, another technique known as fitness sharing is used.

Fitness sharing: Introduced by Goldberg (1989), fitness sharing prevents genetic drift and promotes an adequate spread of the Pareto optimal set within the search space. For the present application (minimising cost and maximising benefit), fitness sharing is implemented by first arranging the current population into sub-populations on the basis of cost. Sub-populations are formed by dividing the cost range into several intervals, and grouping all the individuals having a cost within the interval limits into the same sub-population or class. The shared fitness of an individual is then calculated as the fitness of the individual divided by the number of individuals in its class, thereby penalising solutions which are clustered together, and encouraging a wide spread of individuals over the range of objectives.

4.3.2. Structured messy genetic algorithm and multi-objective optimisation

SMGA builds up the complexity of solutions as the evolution proceeds, and so it needs to generate and maintain a large range of "building blocks" of different cost and benefit to form useful solutions with increasingly longer strings throughout the process. It is particularly important in the early stages to avoid one objective dominating the formation of potential solution elements. For example, a population consisting only of very low cost (and low benefit) elements at the start of the SMGA would not allow high benefit (and high cost) elements to be incorporated later in the process when larger, more complex solutions are developed. The multi-objective approach described above is therefore ideal for the SMGA, as it maintains throughout the evolution sub-populations of good individuals spread over a range of costs, thereby providing efficient building blocks throughout the process.

In its turn, SMGA provides an effective technique for generating and developing a wide range of good solutions for the MOO, generating early in the process populations of individuals of diverse benefit and cost. So the link between SMGA and MOO is mutually beneficial and forms an efficient model capable of handling complex water distribution systems, as was shown in its successful application to the design and rehabilitation of piping systems (Halhal et al., 1997). The method is extended in the present paper for application to pumped water distribution problems with storage.

5. Structured messy genetic algorithm parameters

5.1. Coding

The encoding of the design variables onto numerical strings is the principal parameter of SMGA that distinguishes it from an SGA. Unlike the SGA, the string length changes during the process, with strings only accommodating a limited number of the decision variables. The selection of variables encoded is part of the optimisation process and will generally change from string to string, and from one population to the next, so special coding is necessary to identify the variables in a string as well as their values. A data string therefore carries both a reference to the decision variable and the decision for that variable. In the present formulation, the string is actually split into two separate sub-strings. One sub-string contains the decision variable references, while the other indicates the decision numbers for the corresponding variables. So, for example, a design could be represented by the variable sub-string [4 1 5 3 9] in conjunction with the decision sub-string [2 8 3 4 4], meaning that the variable with reference number 4 takes a value corresponding to decision number 2, and so on. This representation is flexible and allows a genetic operator to be applied to one sub-string independently of the other.

The coding for pipe rehabilitation decisions is straightforward, the decision variable for a pipe taking one of several available rehabilitation alternatives according to the pipe's type and location in the network. The coding for tanks and pumping stations is more complicated, with the decision number incorporating the values of several variables.

5.2. Coding for tanks

In the network model, tanks are connected to a node by a short pipe known as a riser, of known length but of variable diameter. As tank and riser cannot exist separately, the coding for tanks includes that for the riser diameter. The tank codification is quite different for the two approaches used to model tank storage.

Approach 1: In this approach the decision number for a tank is an integer containing two digits, the first representing the tank volume decision, while the second corresponds to the riser diameter decision. For example, in the following code, variable number 26 is a tank, and has 58 as the corresponding decision integer.

variable sub-string[......26 34 16......]

decision sub-string[......58 43 28......]

This means that there is a tank at location reference 26, with a volume represented by decision number 5 and with a riser whose diameter is given by decision number 8.

Approach 2: In the second approach, the decision integer for a tank defines its volume, elevation and operating levels as well as the diameter of its riser. It is in fact composed of 5 digits gathered into one integer to define all these characteristics. For example, in the following code, tank reference number 26 has decision integer 45836.

variable sub-string[......26 34......]

decision sub-string[......45836 65972......]

The first digit, 4, in the decision sub-string indicates the size of the reservoir. The second and third together (58) define the maximum depth of the reservoir. It is calculated as the percentage of the maximum possible depth fixed according to the pump characteristics and the ground topography. The fourth number (3) defines the bottom water level for the reservoir, while the last number corresponds to the code for the riser diameter.

5.3. Coding for pumping stations

The decision code for pumping stations consists of five digits. One represents the number of new pumps to be installed in parallel to existing ones, while the others define the number of pumps operating during each of the four loading periods considered during the day. Fixed speed pumps with a pre-specified performance rating and curve identical to the existing pumps are assumed for this work.

5.4. Mutation

The mutation operator lets randomly selected digits of newly created strings change in value with a small probability. All design variables are therefore subject to low probability random changes. It prevents premature convergence of the optimisation process and keeps the population diversity. As the string length increases during the evolution, and with different coding types present in the same decision string, some alterations to the mutation operator are made for its use within SMGA. Thus the mutation rate is made variable during the evolution, and is a function of the string length as well as the nature of the decision variable.

5.5. String repair

The various GA operators such as cross-over and mutation are applied to the decision variable sub-string as well as to the decision sub-string. The new decision variable sub-string formed is checked to eliminate any duplicated integers, these being replaced randomly by others as necessary.

6. The "Anytown" example

Fig. 2, represents the water distribution system for the hypothetical "Anytown" system (Walski et al., 1987). The problem is to reinforce the system in the most economic way to meet projected demands, taking into account pumping costs as well as capital expenditure. The town is formed around an old centre situated to the south east of pipe 28, where excavations are more difficult to undertake and consequently are more expensive. There is a surrounding residential area, with some existing industries near node 160 and a projected new industrial park to be developed to the north. Options include duplication (in a range of possible diameters) of any pipe in the system, addition of new pipes, addition and/or extension of pumping stations and provision of new reservoir storage at any location.

Full details are presented in the original publication (Walski et al., 1987) but are summarised here for convenience. Water is pumped into the system from a water treatment works by means of three identical pumps connected in parallel. Table 1 gives five points on the characteristic curve for the pumps, together with the corresponding wire-to-water efficiencies. The water level in the water treatment works is maintained at a fixed level of 10 ft. (Imperial units are retained throughout this example for easy comparison with the original work.)

The average daily water use at each node for years 1985 and 2005 as well as the elevation of the nodes and the bottoms of the tanks are given in Table 2. The two existing tanks are full at water level 255 ft, but they are operated with water levels between elevations 225 and 250 ft. The volume of water below the level 225 is retained for emergency needs. The variation in water use throughout the day is given in Table 3.

A minimum pressure of at least 40 psi must be provided at all nodes at instantaneous peak flow, which is 1.8 times the average day flow. The system is also subject to fire flows under which it must supply water at a minimum pressure of at least 20 psi. The critical fire flows are:

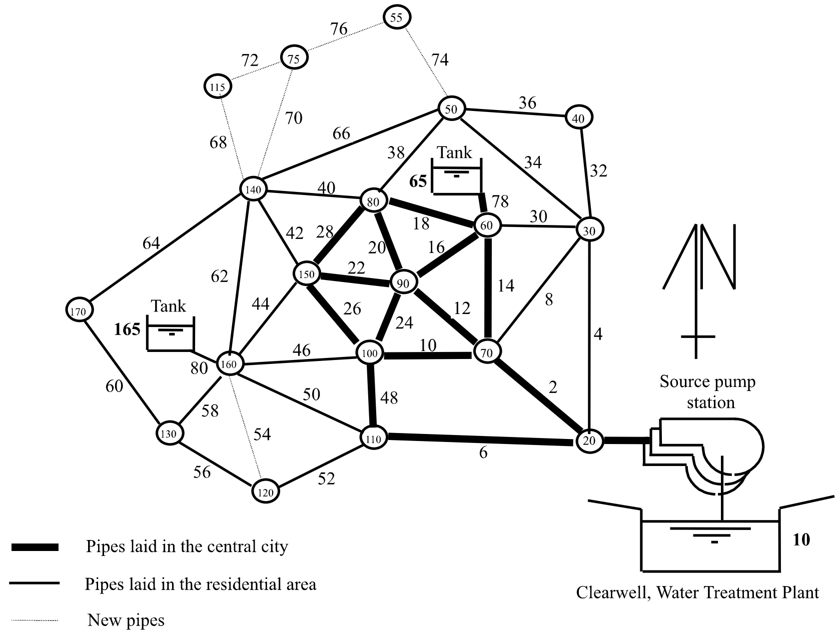

Fig. 2. The "Anytown" network.

Table 1
Pump characteristics

Discharge (gpm)	Pump head (ft)	Efficiency (%) (wire to water)
0	300	0
2000	292	50
4000	270	65
6000	230	55
8000	181	40

1. 2500 gpm at node 90,
2. 1500 gpm at nodes 55, 75 and 115,
3. 1000 gpm at nodes 120 and 160.

The fire flow duration is two hours, and must be met while also supplying peak day flows, which are 1.3 times average flow. All the pressure requirements must be met with one pump out of service and the tank water levels at their low level during a normal day.

The unit costs (in \$/ft) for pipe laying, pipe cleaning and lining are given in Table 4. The capital costs for pumping stations are based on the rated discharge (Q_R) and head (H_R) of the pumping station. For new equipment the cost is estimated by

$$C = 500Q_R^{0.7}H_R^{0.4}.$$

For upgrading an existing pumping station, the cost is estimated by

$$C = 350Q_R^{0.7}H_R^{0.4},$$

where C is the capital cost in dollars; Q_R the rated discharge (gpm), and H_R the rated head (ft).

Pump station operating costs are based on a unit cost for energy, constant throughout the 24 h day, equal to \$0.12/kWh. The present worth of energy costs are based on an interest rate of 12% and an amortisation period of 20 years.

Table 5 gives the pipe characteristics, with the Hazen–Williams C factors as projected values for the year 2005. Pipes laid in the central city are represented in thick solid lines in Fig. 2.

Tank costs are considered as a function of volume and are given in Table 6. Intermediate tank sizes are considered in the proposed methods and the corresponding costs are determined linearly according to the standard size and cost given in Table 6.

The problem was tackled by the two approaches to storage modelling previously described. All nodes are considered as potential sites for new tanks, except those which are already connected directly to existing tanks (nodes 60 and 160). All pipes were considered for duplication, and all new pipe routes treated as optional. No new pumping stations were considered, but possible additional pumps were allowed as options in the existing pumping station, and the on/off status of all pumps during the four time periods of the day were considered as decision variables. The SMGA program was run three times for each approach, stopping after 50 000 solution evaluations in each run. The results are summarised on the graph of Fig. 3, which shows all non-dominated solutions accumulated from the six runs.

The two approaches to storage design perform differently and give final results that are significantly different from each other in detail. Solutions yielded by

Table 2
Nodal data

Node	Average daily demand of year 1985 (gpm)	Expected average daily demand in year 2005 (gpm)	Elevation (ft)
10	Treatment works	Treatment works	10
20	500	500	20
30	200	200	50
40	200	200	50
50	200	600	50
55	–	600	80
60	500	500	50
65	Tank	Tank	215
70	500	500	50
75	–	600	80
80	500	500	50
90	1000	1000	50
100	500	500	50
110	500	500	50
115	–	600	80
120	200	400	120
130	200	400	120
140	200	400	80
150	200	400	120
160	800	1000	120
165	Tank	Tank	215
170	200	400	120

the second method were generally seen to have smaller storage capacities and be slightly more expensive. The first approach is about 30% more CPU time consuming than the second one, but its solutions were found to be more robust. In particular, as the elevation and the water levels of the reservoirs are determined analytically, once the solution satisfies the pressure constraints, all tanks will usually fill up and drain completely. However, in the second approach, tank, pipeline and pumping characteristics are all generated together by the GA process. Their values are controlled indirectly by constraint penalties. As the evolution proceeds, the process tends first to satisfy the head constraints at low cost by locating small tanks at critical nodes. However, the levels are initially too high for the pumps to fill the tanks, and are subsequently altered during the process to satisfy the flow constraints into and out of the tanks. However, in some runs the evolution ends before fully meeting these flow constraints, and the non-dominated solutions are, in these cases, non-feasible in terms of balanced operation, even though they may meet the pressure requirements.

Table 3
Daily water use pattern

Demand period	Time of the day	Average day demand factor
1	00:00–03:00	0.7
2	03:00–06:00	0.6
3	06:00–09:00	1.2
4	09:00–12:00	1.3
5	12:00–15:00	1.2
6	15:00–18:00	1.1
7	18:00–21:00	1.0
8	21:00–24:00	0.9

In its original form, the "Anytown" problem is not multi-objective, as the requirement is to find the least cost solution that satisfies all the constraints. However, using MOO, SMGA was able to identify a range of fully feasible solutions with various characteristics and having total costs ranging from \$10.9 million to more than \$12 million. All these solutions meet the pressure constraints and have balanced operation of reservoirs when modelled using a full simulation based on 24 one-hour time steps, this being a design requirement for the original problem. They all have a total storage capacity of more than 1.5 million gallons, and typically up to 2 new reservoirs and no new pumps. Two pumps operate during the whole day, with a third one operating for a period of 6 h in some cases during peak hours but most often during a low consumption period to help fill the tanks.

Feasible solutions of various storage capacities, meeting all the constraints under the different loading conditions differ from each other not only in the number and size of new tanks and pipeline decisions, but also in the way the tanks and the pumps operate during the day. Some of the solutions found give highly efficient use of storage, with reservoirs emptying and filling almost steadily throughout the day. Other solutions, although meeting all the specified constraints, give intermittent emptying and filling in what would appear to be a less

Table 4
Pipe rehabilitation alternative costs

Pipe diameter (in)	New pipes ($/ft)	Duplicating existing pipes ($/ft)		Clean and line existing pipes ($/ft)	
		City	Residential	City	Residential
6	12.8	26.2	14.2	17.0	12.0
8	17.8	27.8	19.8	17.0	12.0
10	22.5	34.1	25.1	17.0	12.0
12	29.2	41.4	32.4	17.0	13.0
14	36.2	50.2	40.2	18.2	14.2
16	43.6	58.5	48.5	19.8	15.5
18	51.5	66.2	57.2	21.6	17.1
20	60.1	76.8	66.8	23.5	20.2
24	77.0	109.2	85.5	–	–
30	105.5	142.5	116.1	–	–

Table 5
Pipe characteristics

Pipe ID	Length (ft)	Diameter (in)	Location	HW coefficient
2	12 000	16	City	70
4	12 000	12	Residential	120
6	12 000	12	City	70
8	9000	12	Residential	70
10	6000	12	City	70
12	6000	10	City	70
14	6000	12	City	70
16	6000	10	City	70
18	6000	12	City	70
20	6000	10	City	70
22	6000	10	City	70
24	6000	10	City	70
26	6000	12	City	70
28	6000	10	City	70
30	6000	10	Residential	120
32	6000	10	Residential	120
34	9000	10	Residential	120
36	6000	10	Residential	120
38	6000	10	Residential	120
40	6000	10	Residential	120
42	6000	8	Residential	120
44	6000	8	Residential	120
46	6000	8	Residential	120
48	6000	8	City	70
50	6000	10	Residential	120
52	6000	8	Residential	120
54	9000	New	Residential	130
56	6000	8	Residential	120
58	6000	10	Residential	120
60	6000	8	Residential	120
62	6000	8	Residential	120
64	12 000	8	Residential	120
66	12 000	8	Residential	120
68	6000	New	Residential	130
70	6000	New	Residential	130
72	6000	New	Residential	130
74	6000	New	Residential	130
76	6000	New	Residential	130
78	100	12	City	120
80	100	12	Residential	120
82	100	30	Residential	130

Table 6
Tank costs

Tank volume (gal)	Cost ($)
50 000	115 000
100 000	145 000
250 000	325 000
500 000	425 000
1 000 000	600 000

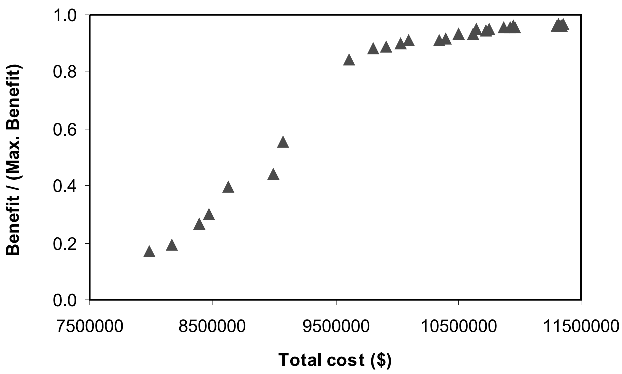

Fig. 3. Best solutions of differing costs.

efficient fashion. This mode of operation is noticeable in some of the previously published solutions. For example, in the design by Murphy et al. (1994), tank 185 fills up in 4 h, remains full for 8 h, then releases its entire contents in about 3 h, which is far from efficient operationally.

One of the benefits of the present multi-objective approach is the generation of a range of solutions, from which different designs can be selected. Two such solutions are presented here: the cheapest feasible solution at $10.914 million and the most operationally satisfactory solution (preferred by the authors) at $11.044 million. Both are cheaper solutions than any previously reported, as shown in Tables 7 and 8.

6.1. Solution 1

The cheapest solution has a total cost of $10.91 million, with $4.14 million on new and improved pipelines, $0.86 million on new tanks and $5.91 million as the present value of future energy consumption. The piping improvements, tank locations and capacities are shown in the schematic of Fig. 4. Tank details are given in Table 9. No new pumps are provided, and only two of the three existing pumps operate and do so for the whole day with an energy consumption of 18 080 kWh. Two new tanks are included. The pressure requirements are met at all the specified design loadings, as shown in Table 10. Figs. 5 and 6 summarise the operation of reservoir balancing in the system, the results being obtained from a simulation over 24 one-hour time steps. It can be seen that three of the four tanks remain at a constant low level for most of the day, with the two existing tanks filling and emptying very rapidly in the early morning. The elevation of tank 55 is also questionable. Although it meets the specifications and constraints on the problem as originally defined, it is clear that on days when the demand is significantly less than the maximum, the tank will remain full, which is undesirable from a water quality standpoint.

6.2. Solution 2

The preferred solution costs $11.04 million, consisting of $4.26 million on pipes, $0.74 million on new storage and $6.04 million on future energy costs. The solution is shown in schematic form in Fig. 7 and has just one new tank of 1.4 million gallons storage capacity. Tank details are given in Table 11. No new pumps are needed to reinforce the pumping station. The three existing pumps operate for the 6 h of the low demand period to fill up the tanks, while only two are required for the remaining time, giving an energy consumption of 18 460 kWh for the day.

Table 7
Cost comparison with previously published solutions ($\$ \times 10^6$)

Description	Gessler (Walski et al., 1987)	Lee et al. (Walski et al., 1987)	Morgan & Goulter (Walski et al., 1987)	Ormsbee (Walski et al., 1987)	Murphy et al., 1994)	Walters et al. (present study)	
						Solution 1	Solution 2
Pipes	4.5	4.6	3.3	5.7	4.5	4.1	4.3
Pump equipment	0.7	1.8	2.7	0.0	0.0	0.0	0.0
Energy	6.6	6.0	6.3	6.3	6.0	5.9	6.0
Tanks	0.5	0.5	0.7	1.8	0.9	0.9	0.7
Total	12.3	12.9	13.0	13.8	11.4	10.9	11.0

Table 8
Comparison of designs with previously published solutions

	Length (ft) of cleaned or new pipes by diameter (in) shown below										New tanks		Pumps
	6	8	10	12	14	16	18	20	24	30	Node	Size[a]	
Gessler	12 000	9000	6000	12 000	30 000	6000	6000	–	18 000		150	0.8	1
Lee et al.[b]	27 000	–	42 000	9000	18 000	12 000	9000	12 000			140	0.7	3
Morgan & Goulter	15 000	24 000	24 000	18 000	–	–	6000	12 000		–	75	1.0	1
											170	0.1	
Ormbsee	12 000	24 000	30 000	6000	–	6000	24 000	33 000		–	80	Total = 3.0	
											150		–
											160		
Murphy et al.	21 000	6000	24 100	12 100	18 000	6100	12 000	12 100		–	70	0.75	–
						12 000[c]					140	0.3	
Walters et al. (solution 1)	15 000	6100	12 100	18 000	6000	6000	200	12 000	12 000	100	55	0.70	–
		6000[c]				12000[c]					110	0.35	
Walters et al. (solution2)	15 000	100	18 000	30 000	6000	12 000[c]	–	6200	18 000	–	140	1.4	–

[a] Volume in MG.
[b] Solution with two diameters listed for each pipe link.
[c] Cleaned.

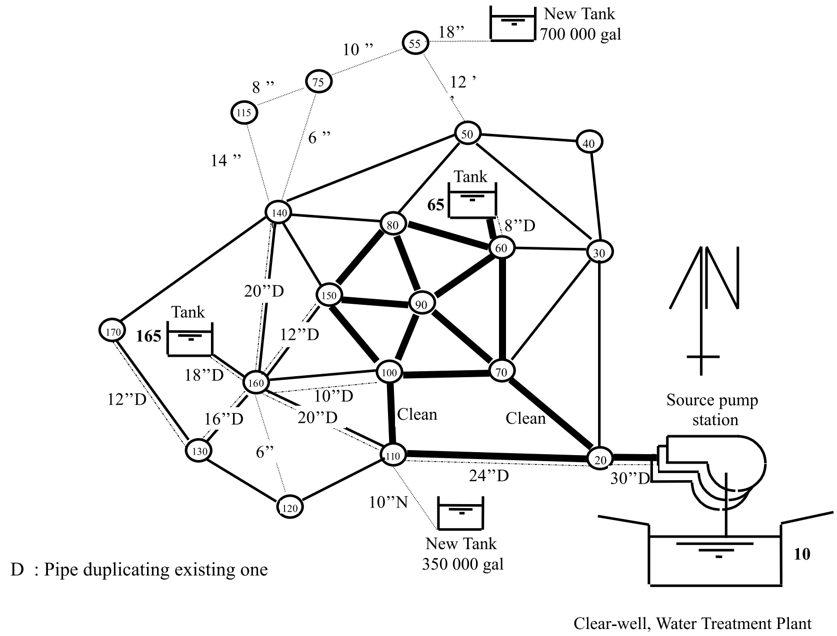

Fig. 4. Layout of the cheapest solution.

Table 9
Tank characteristics

Location node	Capacity (gal)			Operating water level (ft)		Bottom of the tank
	Effective	Emergency	Total	High	Low	
Existing tank 65	156 250	62 500	250 000	250	225	215
Existing tank 165	156 250	62 500	250 000	250	225	215
New tank 55	300 000	400 000	700 000	204	189	169
New tank 110	250 000	100 000	350 000	251	241	237

Table 10
Minimum pressures

Loading patterns	Min. pressure (psi)	
	Value	Node
Instantaneous peak flow	40.59	120
Fire flow at node 90	41.73	150
Fire flows at 55, 75 and 115	37.59	75
Fire flows at 120 and 160	36.36	120

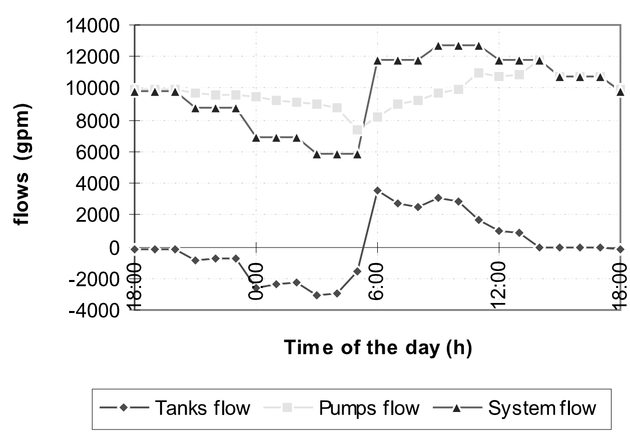

Fig. 5. Different flow origins.

The solution satisfies the pressure constraints for the different loading patterns with tanks assumed to be at their low operating levels. Table 12 gives the most critical node and the value of the minimum pressure at that node for the various loading conditions. In addition, fire flow pressure constraints were checked at the beginning and the end of the fire duration (2 h), i.e. with tanks starting at their low operating levels and finishing at their bottom levels. Table 13 shows the minimum pressures and the different sources of flow in the system, at the start and end of different fire events.

A simulation of the solution was performed over 24 one-hour time steps for the maximum day event, and showed that the tanks use all their available operational

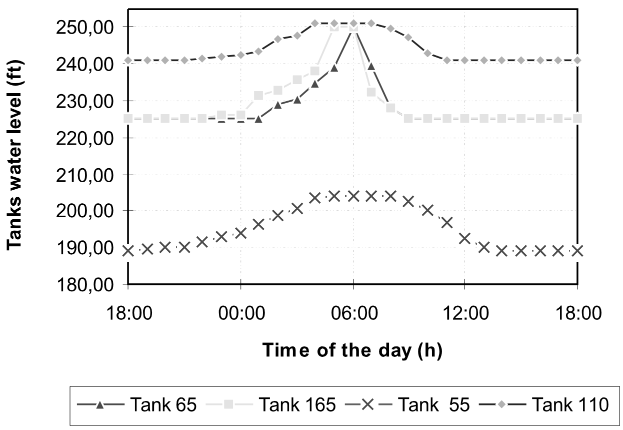

Fig. 6. Tank operating levels over a cycle of 24 h.

volumes during the cycle, as shown in Fig. 8. Existing tank 65 fills up in 6 h and drains in more than 5.5 h, while tank 165 fills up in 12 h and drains during 12 h. New tank 145, which has an effective storage capacity exceeding 6 times that of each of the existing tanks, has a very efficient operating cycle. It remains at its lowest operational level between 18.00 and 21.00 hours, during which period demand equals the average demand, fills until 06.00 hours in the morning, then releases its contents smoothly during the whole peak demand period. The pumps are either operating at the average daily demand (for about 14 h during the day) or close to it. Fig. 9 shows the pump and the tank flows during the day.

7. Conclusion

The previously published SMGA approach to water distribution network rehabilitation has been extended to include pumping and storage. Two approaches to modelling the storage elements were tested in a multi-objective formulation. The two approaches differ in the

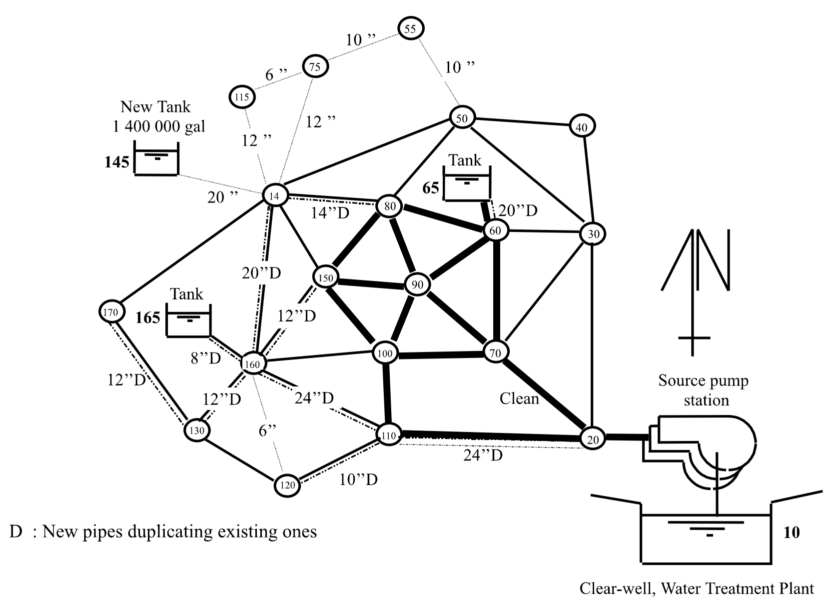

Fig. 7. Layout of the preferred solution.

Table 11
Tank characteristics

Location node	Capacity (gal)			Operating water level (ft)		Bottom of the tank
	Effective	Emergency	Total	High	Low	
65	156 250	62 500	250 000	250	225	215
165	156 250	62 500	250 000	250	225	215
New tank 145	950 000	450 000	1 400 000	240	220	210

Table 12
Minimum pressures

Loading pattern	Minimum pressure (psi)	
	Value	Node
Instantaneous peak flow	40.10	170
Fire flow at node 90	41.51	150
Fire flows at 55, 75 and 115	37.42	55
Fire flows at 120 and 160	41.94	170

way they determine the operating levels for new tanks (reservoirs). The first method determines levels analytically during the network analysis routine, whereas the second method specifies levels as independent decision variables.

The two methods were applied to the "Anytown" benchmark network, and yielded comparable results but with noticeable advantages to the first method, which, although slightly more time consuming, yielded more robust solutions. Both methods require several hours to run on a high performance PC.

The multi-objective approach combines well with the SMGA. The "Anytown" problem was originally presented with the single objective of minimising cost, but was successfully reformulated here in terms of minimising cost and maximising benefit, the benefit being treated as the reduction in predicted hydraulic deficiencies.

Two fully feasible solutions were selected and presented. They are 4–5% cheaper than any previously published solutions to the "Anytown" problem. The authors' preferred solution has storage tanks operating highly efficiently, and has a total storage capacity exceeding those present in most of the previously published solutions.

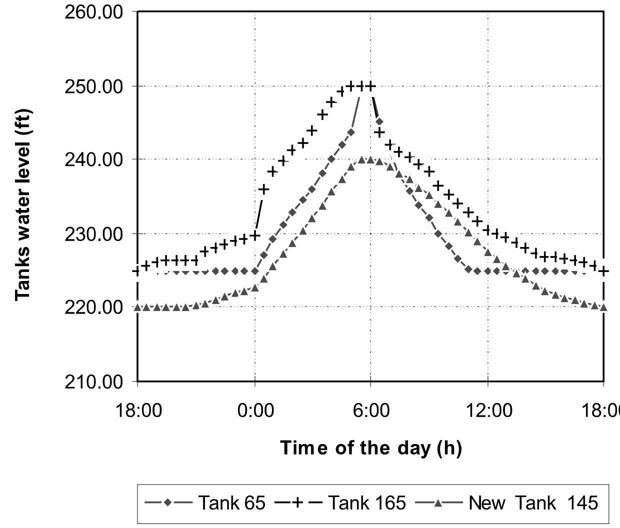

Fig. 8. Tank operating levels over a cycle of 24 h.

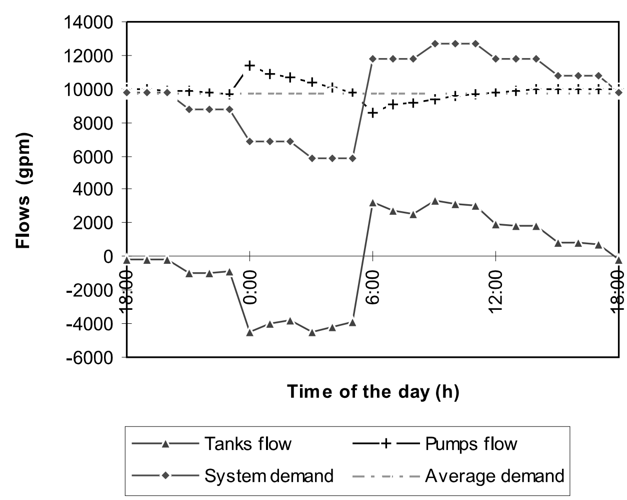

Fig. 9. Pump and tank flows during the day.

Table 13
Fire flows and pressures

Water level	Tanks at their low operating level					Water at the bottom of the tanks				
Fire flow patterns	Pressure (psi)	Source of flows (gpm)				pressure (psi)	Source of flows (gpm)			
		pumps	Tank 65	Tank 165	Tank 145		pumps	Tank 65	Tank 165	Tank 145
Fire flow at 90	41.51	10 137	1428	798	1577	37.25	10 680	1288	431	1541
Fire flow at 55, 75 and 115	35.42	10 108	1211	556	3025	31.17	10 647	1101	168	2984
Fire flow at 120 and 160	41.94	10 119	977	595	1229	37.72	10 660	860	213	1187

References

DeNeufville, R., Schaake, J., & Stafford, J. (1971). Systems analysis of water distribution networks. *Journal of Sanitary Engineering Division, ASCE*, 97 (SA6), 825–842.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimisation, and machine learning*. Reading, MA: Addison-Wesley.

Halhal, D., Walters, G. A., Ouazar, D., & Savic, D. A. (1997). Water network rehabilitation with structured messy genetic algorithm. *Journal of Water Resources Planning and Management ASCE*, *123* (3), 137–146.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.

Michalewicz, Z. (1992). *Genetic algorithms + data structure = evolution programs*. Berling: Springer.

Murphy, L.J., Dandy, G.C., & Simpson, A.R. (1994). Optimum design and operation of pumped water distribution systems. *Proceedings of the Conference on Hydraulics in Civil Engineering.* Brisbane, Australia: Institution of Engineers, Australia.

Walski, T. M., Brill, E. D., Gessler, J., Goulter, I. C., Jeppson, R. M., Lansey, K., Han-Lin Lee, , Liebman, J. C., Mays, L., Morgan, D. R., & Ormsbee, L. (1987). Battle of the network models: epilogue. *Journal of Water Resources Planning and Management, ASCE*, *113* (2), 191–203.