Johns, M.B., E. Keedwell and D.A. Savić (2014) Adaptive locally constrained genetic algorithm for least-cost water distribution network design, Journal of Hydroinformatics, Vol. 16, No. 2, pp. 288-301, doi:10.2166/hydro.2013.218

ADAPTIVE LOCALLY CONSTRAINED GENETIC ALGORITHM FOR LEAST-COST WATER DISTRIBUTION NETWORK DESIGN

MATTHEW B. JOHNS (1), EDWARD KEEDWELL (1), DRAGAN SAVIC (1)

(1): College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK

{mbj202, E.C.Keedwell, D.Savic}@ex.ac.uk

ABSTRACT

This paper describes the development of an Adaptive Locally Constrained Genetic Algorithm (ALCO-GA) and its application to the problem of least cost water distribution network design. Genetic algorithms have been used widely for the optimisation of both theoretical and real-world non-linear optimisation problems, including water system design and maintenance problems. In this work we propose a heuristic based approach to the mutation of chromosomes with the algorithm employing an adaptive mutation operator which utilises hydraulic head information and an elementary heuristic to increase the efficiency of the algorithm's search into the feasible solution space. In almost all test instances ALCO-GA displays faster convergence and reaches the feasible solution space faster than the standard genetic algorithm. ALCO-GA also achieves high optimality when compared to solutions from the literature and often obtains better solutions than the standard genetic algorithm.

KEYWORDS | genetic algorithm, heuristic, optimisation, water distribution.

INTRODUCTION

Evolutionary algorithms (EAs) are used widely for the optimisation of both theoretical and real-world optimisation problems. These problems tend to be highly complex and incorporate one or more constraints that limit the feasible space to be searched. In this work, we propose the use of a heuristic based approach for the mutation of chromosomes and demonstrate the approach on several water distribution network design problems. Although the heuristic used is specific to the problem, the method of constraining the solutions and their children is general and applicable to other domains. The heuristic based locally constrained approach is shown to perform better than a standard evolutionary algorithm on the majority of water distribution network design problems.

Water distribution network (WDN) design is a complex non linear optimisation problem, commonly involving a large number of different network components and hydraulic constraints. Due to the inherent complexity of WDN design, a simplified formulation of the problem is commonly employed when applied to optimisation techniques. This method is commonly comprised of the allocation of a diameter to each pipe in a given network layout, with the objective of minimising cost whilst satisfying pressure constraints (Alperovits & Shamir, 1977). In this simplified version, design considerations such as water quality and network reliability are not included in the formulation of the problem. This method provides the designer with a base from which to solve the overall problem and allows the comparison of new optimisation techniques with the large amount of literature that employs this technique of problem formulation.

The optimal design of water distribution networks is considered a NP-hard problem (Yates, Templeman, & Boffey, 1984) and has been solved with a number of approaches, such as classic methods that include linear and dynamic programming (Schaake Jr. & Lai, 1969)(Kessler & Shamir, 1989)(Fujiwara & Khang, 1991)(Sherali, Subramanian, & Loganathan, 2001)(Bragalli, Ambrosio, Lee, Lodi, & Toth, 2008) and various heuristic algorithms. Due to the discrete nature of the decision space and the advent of effective hydraulic solvers, the application of global stochastic optimisation algorithms has been proven to be a good approach to the WDN design problem. These approaches, although effective can induce a large number of hydraulic evaluations which in the case of large, real world WDNs can become extremely computationally expensive. Over the last two decades a considerable amount of research has been applied to the problem of WDN design especially in the field of EAs such as Genetic Algorithms (GAs) (Dandy, Simpson, & Murphy, 1996)(D. A. Savic & Walters, 1997)(Wu & Simpson, 2001)(Kadu, Gupta, & Bhave, 2005)(Artina, Bragalli, Erbacci, Marchi, & Rivi, 2012), Simulated Annealing(Cunha & Sousa, 1999), Shuffled Complex Evolution (Liong & Atiquzzaman, 2004), Ant Colony Optimisation (Maier et al., 2003) and Harmony Search (Geem, Kim, & Loganathan, 2002). These techniques have proven to be effective on a number of benchmark WDN design problems.

Constraint Handling in EAs

In their basic form, EAs are unconstrained optimisation procedures. However, many problems have constraints imposed upon them especially in real-world optimisation problems. A common approach to dealing with constrained optimisation problems is to incorporate the constraints into the fitness function of the EA by adding a penalty function to the fitness function, where the value obtained from the penalty function represents the solution's distance from feasibility. A frequently used approach is the static penalty (Morales & Quezada, 1998), where the penalty factors remain constant throughout the evolutionary process. Another approach is the use of a dynamic penalty where the penalty function is varied over time, commonly tightening the constraints as the EA's population develops. The notion of allowing an EA to explore the search space unimpeded before increasing the focus of the search and therefore potentially improving the scope of the search has led some researchers to argue that dynamic penalties perform better than a static penalty approach. However, it has been found that deriving an effective dynamic penalty function is as difficult to achieve as producing good penalty factors for static functions (Siedlecki & Sklansky, 1989).

Another approach when handling the constraints of a problem is to employ a repair algorithm. The repair algorithm has proven a popular choice for many combinatorial optimisation problems as it is often relatively easy to 'repair' an infeasible solution through the iterative modification of individual decision variables. When a solution can be transformed from infeasible to feasible at a low computational cost, repair algorithms have proven to be effective. However, it is not always possible to repair an infeasible solution at an acceptable computational cost and in some cases the algorithm can harm the evolutionary process by introducing a strong bias in the search (Smith & Coit, 1997).

A further method is to use an indirect representation where the genes do not code for variables in the problem directly, but via a heuristic that determines the phenotype given the genotype developed by the algorithm. These approaches have been shown to work well in timetabling problems (Paechter, Rankin, Cumming, & Fogarty, 1998) but the relationship between the genotype and phenotype is more complex leading to a more multimodal fitness landscape.

Parameter Tuning

An important decision when implementing a genetic algorithm is the values for the various parameters, such as population size, tournament size, crossover rate and mutation rate. Typically these parameters have a nonlinear relationship with one another and therefore simultaneous optimisation is not feasible. Although there is a great amount of discussion in the evolutionary algorithm literature regarding parameter allocation, a conclusive approach to parameter selection for any problem has yet to be found.

(De Jong, 1975) conducted a systematic study on how search performance of a genetic algorithm was affected when varying the parameters. The study indicated that the best population size was 50-100 individuals, the best single-point crossover rate was ~ 0.6 and the best mutation rate was 0.001. Although these parameter values became widely used, it was uncertain how these setting would perform on problems other than that of De Jong's.

Other methods of parameter selection include optimising parameter settings using evolutionary algorithms (Grefenstette, 1986)(Bramlette, 1991) and the self adaptation of operator values (Davis, 1989). These approaches have shown to improve the performance of evolutionary algorithms on some problems.

Proposed Locally Constrained Approach

This paper describes the development of an Adaptive Locally Constrained Genetic Algorithm (ALCO-GA) and its application to the least cost WDN design problem. This method incorporates constraint handling based on problem feasibility directly into the chromosome construction and adapts the application of this operator based on the evolutionary progress of the algorithm. ALCO-GA is based upon a standard genetic algorithm and incorporates a modified mutation operator which directly targets elements of a network with the aim to reduce constraint violation using hydraulic head information and an elementary heuristic. In addition, an adaptive method for controlling the application of the heuristic based mutation operator is employed using convergence information. Experiments are conducted to compare a standard genetic algorithm and ALCO-GA for a number of benchmark WDN design problems including one large real-life problem. In addition, a further investigation is conducted into the effect parameter selection has on the effectiveness of both the standard genetic algorithm and ALCO-GA. The results show that the ALCO-GA approach can improve convergence rates and the number of evaluations required to find a feasible solution.

Optimal Design Problem Formulation

The optimal design of a water distribution network is presented here using the following mathematical statement. The objective function is defined as the total cost of the network with regard to pipe length and diameter

$$f(D_1, \ldots, D_n) = \sum_{i=1}^{N} c(D_i, L_i)$$

where $c(D_{i,} L_i)$ = cost of pipe i with diameter D_i and length L_i with N = number of pipes in the network. This function is to be minimised whilst satisfying the following constraints. For each junction (excluding the source) the following continuity constraint has to be satisfied

$$\sum Q_{in} - \sum Q_{out} = Q_e$$

where Q_{in} = inflow to the junction, Q_{out} = outflow from the junction and Q_e = external flow or junction demand which in this case is always positive. Head loss h_f for a specific pipe i is calculated using the following equation

$$h_f = \omega \frac{L_i}{C_i^a D_i^b} Q_i^a$$

where C_i = Hazen-Williams roughness coefficient, Q_i = flow and a, b, and ω are parameters of the equations.

The minimum head constraint for each junction in the network is as follows

$$H_j \geq H_j^{min}; j = 1, \ldots, M$$

where H_j = head at junction j, H_j^{min} = minimum head requirement at junction j and M = total number of junctions present in the network.

In the case of this formulation of the WDN design problem the optimisation is exclusively concerned with the selection of pipe diameters. Each individual problem has a set of available pipe diameters which can be selected for each decision pipe in the network. These decisions are encoded as a binary bit sub-string with a length dictated by the number of pipe sizes available. The substrings are then concatenated to form the chromosome to represent the entire solution.

ADAPTIVE LOCALLY CONSTRAINED GENETIC ALGORITHM

ALCO-GA applies a constraint based rule directly to the genotype without evaluating the effect this process has on the phenotype. The heuristic employed by ALCO-GA is developed from the constraints of a specific problem and remains constant throughout the evolutionary process. The heuristic is applied to a solution through the mutation operator; where the probability of the heuristic mutation operator being applied is directly influenced by the rate of convergence of the best solution in the population. It is the aim of this operator to guide the algorithm's search to the feasible solution space in a fast and efficient manner utilising constraint violation data from previous fitness evaluations. Unlike some other constraint handling techniques such as repair algorithms, the ALCO-GA mutation operator does not perform any additional partial or full fitness evaluations, except a single hydraulic simulation at initialisation to determine flow directions.

Standard Genetic Algorithm

ALCO-GA is essentially a standard GA (SGA) which incorporates some additional features; these include a heuristic based mutation operator and a fitness gradient monitor. The standard GA used was a steady-state GA with tournament selection with tournament size t, single-point crossover with probability c and a grey-coded binary string representation. Mutation was conducted as a random bitwise mutation with probability m.

Heuristic Based Mutation Operator

The Heuristic based Mutation Operation (HMO) is designed to allow the EA to locate feasible network designs earlier in the optimisation. It can be configured for use with any appropriate constraints, but here the application to network hydraulic performance only is considered. ALCO-GA employs a simple heuristic that is designed to primarily target pipes which are causing head constraint violations. At initialisation, the program runs a single hydraulic simulation of the WDN and logs the directional flow information of each pipe. Using this data, each junction and its immediate upstream pipe and junction are logged making it possible to identify pipes that are limiting junction head down-stream. The HMO first selects a junction through the use of a roulette wheel method which assigns wheel segment sizes using head deficit information obtained during the fitness evaluation of the solution, resulting in junctions with a high pressure head deficit having a greater probability of being selected. Once a junction is selected the heuristic searches upstream of the selected junction until a junction is found which is in pressure head excess. The pipe immediately downstream of the discovered junction is then mutated to a greater available diameter. If the pressure head at every junction satisfies the problem constraints, the HMO employs a slightly different method. The roulette wheel method is employed again; however, wheel segment size and therefore junction selection probability is now proportional to junction pressure head excess. This results in junctions with high pressure head excess having a greater probability of selection. Once selected the pipe immediately upstream of the initially selected junction is mutated to a smaller size.

It was found that if the HMO was employed exclusively (i.e., without random bitwise mutation) throughout the evolutionary process, the population would become stagnant and prematurely converge on a sub optimal solution. Therefore it was necessary to limit the amount the algorithm employs the HMO. It was observed that the HMO was very effective in the early stages of an algorithm run, finding the feasible solution space quickly. However the random bitwise mutation operator was found to be more effective in the latter generations of the algorithm. Therefore, it was necessary to employ a device to influence the usage of HMO. The Fitness Gradient Monitor (FGM) controls the probability that the HMO is applied to the current generation's child solutions, through the monitoring of the population's current best solution's fitness. The probability of the application of HMO is decreased from 1 as the fitness curve of the best solution's fitness tends to zero.

$$P_{gm} = \frac{g_c}{g_i}$$

where g_i is the initial gradient, g_c is the current gradient and P_{gm} is the probability of the HMO being employed. If HMO is not utilised then the standard bitwise mutation operator is used instead. This method ensures a smooth transition between the use of HMO and bitwise mutation operator as the algorithm's search progresses. In ALCO-GA the mutation operator precedes the crossover operator. Due to the dependency of the HMO on a solution’s pressure head information, mutation cannot be applied post crossover without the need to re-evaluate the hydraulic network of resultant solutions. This is illustrated in figure 1.

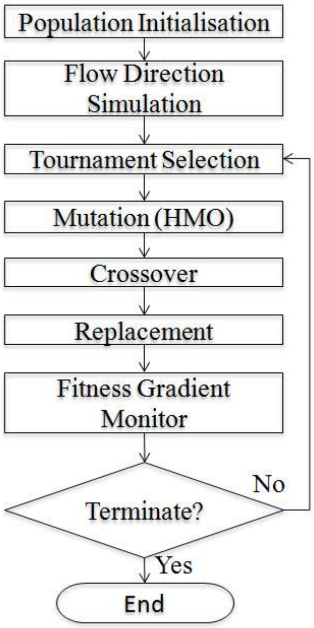

Figure 1. Flow diagram illustrating the structure of ALCO-GA.

COMPUTATIONAL RESULTS AND DISCUSSION

ALCO-GA was coded in C++ and run on an Intel Core i5 2.67GHz PC. The test problems used to evaluate the algorithm including a number of benchmark networks from the literature. The majority of the following test cases can be found at *www.ex.ac.uk/cws/benchmarks*. In all test cases both ALCO-GA and SGA are run using identical parameters. To ensure the monitoring of CPU time was accurate, each algorithm instance was locked to a single core of the CPU. Each run was terminated after 100,000 iterations (200,000 fitness evaluations).

Two Loop Network

Firstly the algorithm was tested on the Two Loop network problem (Alperovits & Shamir, 1977) a small scale benchmark network consisting of eight pipes arranged in a two loop configuration fed by a fixed 210m head. There are 14 available pipe sizes ranging from 25.4mm to 609.6mm in diameter. The parameters of SGA were reasonably tuned to the problem through a number of evaluation runs. It was found that the following parameters achieved the best results for the Two Loop network problem: population size $N = 100$, tournament size $t = 0.02N$, probability of mutation $m = 0.15$ and a penalty factor $k = 20,000$ \$/m. Both SGA and ALCO-GA were run a total of 100 times.

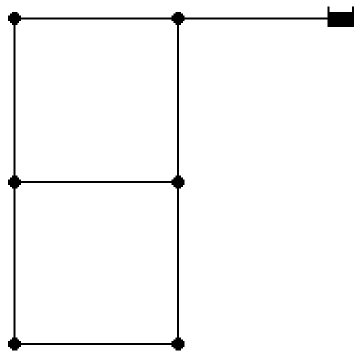

Figure 2. Two-Loop Network

The average results that meet the problem constraints from the 100 runs can be seen in table 1 for both the SGA and ALCO-GA for the Two Loop network problem. It can be observed from the results in table 1 that ALCO-GA outperforms SGA, obtaining a lower average best cost network solution and finding the best known solution ($419,000) a greater number of times. ALCO-GA also demonstrates faster convergence, achieving the best solution in significantly less fitness evaluations and CPU time.

Table 1. Computational results of both Algorithms for the Two Loop Problem.

Algorithm	*Standard GA*	*ALCO-GA*
Average Best Solution ($)	419,160	**419,110**
Number of Best Known Sols.	84	**89**
Standard Deviation ($)	366.606	**312.89**
Mean Evaluations to Best Sol.	54,953	**38,115**
Mean CPU Seconds to Best Sol.	4.7	**3**

New York Tunnels Problem

The New York Tunnels Problem (Schaake Jr. & Lai, 1969) is a parallel expansion problem consisting of 21 existing pipes and 20 junctions fed by a fixed head reservoir. The objective is to find the least cost configuration of pipes that could be installed parallel to the existing pipes to meet the head constraints of the problem. There are 16 available pipe diameters ranging from 0in to 804.0in therefore no encoding redundancy is required. The parameters of SGA were tuned to the problem as before. It was found that following parameters achieved the best results for the New York Tunnels Problem: population size $N = 100$, tournament size $t = 0.02$N, probability of mutation $m = 0.06$ and a penalty factor $k = 7{,}000{,}000$ $/ft. Both SGA and ALCO-GA were run a total of 100 times.

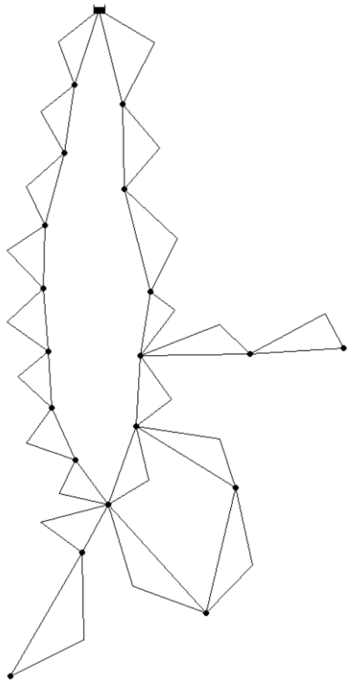

Figure 3. New York Tunnels Parallel Expansion Network

Table 2 shows the average results for the New York Tunnels (NYT) problem from the 100 runs of both SGA and ALCO-GA. ALCO-GA clearly out performs SGA, finding a better quality of solution in nearly half the fitness evaluation and time.

Table 2. Computational results of both Algorithms for the New York Tunnels Problem.

Algorithm	***Standard GA***	***ALCO-GA***
Average Best Solution ($)	39,764,100	**38,748,200**
Number of Best Known Sols.	2	**66**
Standard deviation ($)	461,116	**210,562**
Mean Evaluations to Best Sol.	151,509	**86,450**
Mean CPU Seconds to Best Sol.	30.7	**17.3**

Figure 4 shows the average best solution cost over a number of generations for both SGA and ALCO-GA for the NYT problem. The SGA initially outperforms ALCO-GA due to the feasibility of the initial population. The HMO is primarily designed to find the feasible search space; however, due to the nature of this problem a high proportion of the initial population (approximately 53%) contains feasible solutions, thus decreasing the effectiveness of the algorithm in the early generations. However ALCO-GA overtakes SGA at around 2500 generations and ultimately converges on a better average solution than that of SGA.

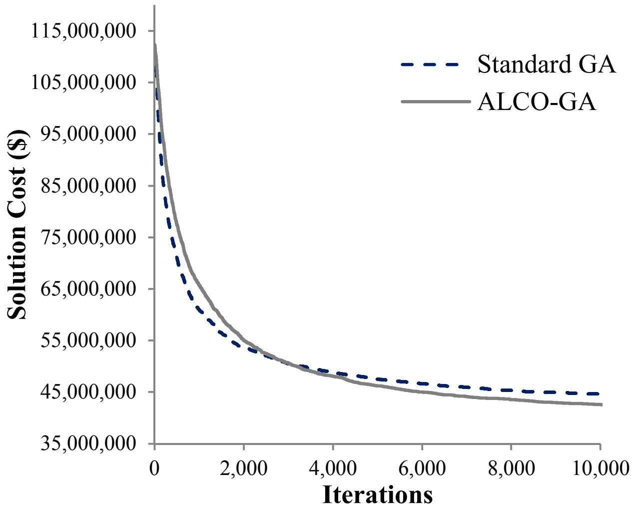

Figure 4. Average performance of best cost solutions for the New York Tunnels Problem.

Network B Problem

Network B (D. Savic, Walters, Smith, & Atkinson, 2000) is a real WDN and consists of 1277 pipes and 1106 junctions, fed by a single fixed head reservoir. 26 pipe diameters are available ranging from 50mm to 999mm. The following parameters were used by both algorithms: population size N = 100, tournament size t = 0.04N, probability of mutation m = 0.01 and a penalty factor k = 1,000,000 $/m. Due to the complexity and resultant runtime, both SGA and ALCO-GA were only run a total of 30 times each.

Figure 5. Network B

Table 3 shows the average results of the run for SGA and ALCO-GA for the Network B problem. The results show ALCO-GA reaches a higher average best solution than SGA in less solution evaluations and in less CPU time.

Table 3. Computational results of Standard GA and ALCO-GA for the Network A Problem.

Algorithm	***Standard GA***	***ALCO-GA***
Average Best Solution ($)	13,864,900	**13,820,700**
Standard deviation ($)	**293,165**	319,382
Mean Evaluations to Best Sol.	190,216	**188,667**
Mean CPU Seconds to Best Sol.	6128.7	**6096.9**

From figure 6 it is evident that in the early generations, ALCO-GA displays rapid converge compared to the slower convergence displayed by SGA. The convergence rate of ALCO-GA slows in the later generations of the runs but achieves higher optimality overall. This is a key advantage of the ALCO-GA approach, although the final networks are similar in performance, the performance gains at the start of the optimisation are high, meaning many fewer objective function evaluations are required to achieve a given optimality of solution. Note the algorithms employ an inverse cost function to generate the fitness value.

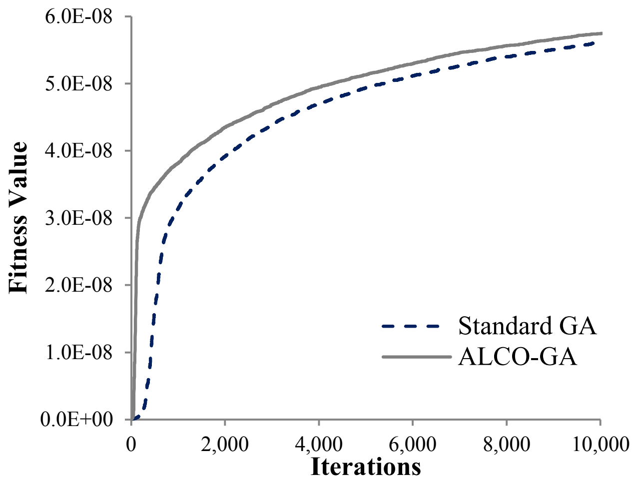

Figure 6. Average performance of best solutions for the Network B Problem.

Figure 7 shows the best feasible solution cost for ALCO-GA and SGA over 100,000 iterations for a single run. Note that ALCO-GA obtains a feasible solution over 3000 iterations (6000 hydraulic simulations) prior to SGA achieving feasibility, meaning that if the objective is to find a feasible solution, this can be achieved with many fewer objective function evaluations.. ALCO-GA also attains greater optimality throughout the run.

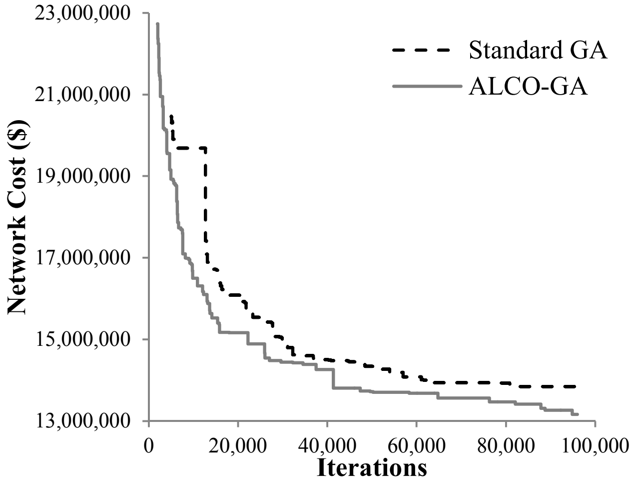

Figure 7. Performance of best solution for the Network B Problem.

Algorithmic Robustness

Identifying the optimal GA parameter set can involve a large number of algorithm runs using a number of different parameter configurations. This can be a time consuming operation and in the case of complex large real-world problems is near infeasible. The sensitivity of an algorithm to its parameter configuration is therefore an important consideration, and in this section, the performance of both the SGA and ALCO-GA are compared over a number of different parameter configurations involving the tournament size and mutation rate.

For the Two Loop and New York Tunnels problems the tournament size was varied at discrete intervals between $0.02N$ and $0.1N$. For each tournament size, the per-bit probability of mutation was varied between a value of 0.02 and 0.5. The remaining algorithm configurations remain the same as descried earlier in the paper. For each parameter configuration, both the SGA and ALCO-GA where run a total of 100 times. The following figures display the mean results for each mutation rate for every tournament size used.

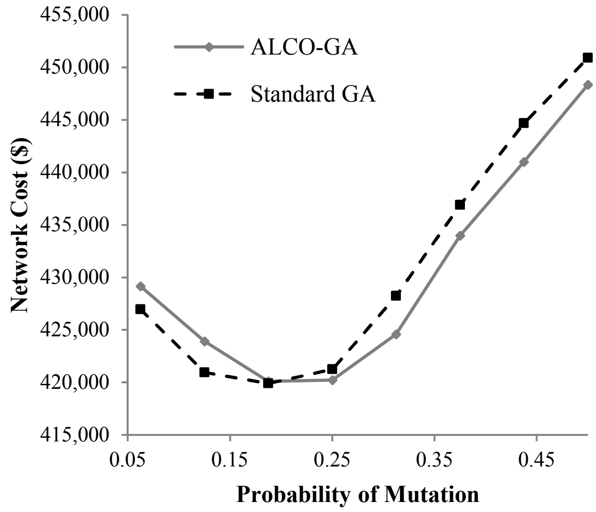

Figure 8. Mean performance of best solution for varying mutation rates for the Two Loop problem.

Figure 8 shows the performance of both the SGA and ALCO-GA with regard to mutation rate and best network cost on the Two Loop problem. The SGA out performs ALCO-GA at lower mutation rates up to 0.2 in terms of average best feasible network cost, however ALCO-GA does surpass SGA at higher mutation rates. Because the ALCO-GA heuristic is applied through the mutation operator, at low mutation rates the effectiveness of ALCO-GA is slightly diminished. However, from the results displayed in figure 9, ALCO-GA tends to find a greater number of best known solutions for the problem; outperforming the SGA at every mutation rate.

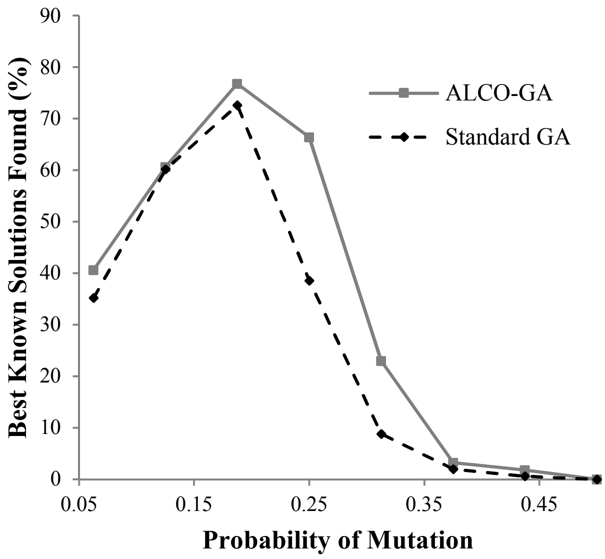

Figure 9. Average number of best known solutions found at varying mutation rates for the Two Loop problem

Figure 10 shows the comparison between SGA and ALCO-GA for the New York Tunnels problem for mutation rates between 0.02 and 0.12. Although the SGA achieves the better mean result at ~ 0.24 mutation rate, ALCO-GA goes on to achieve largely superior results at mutation rates ~ 0.05 and above.

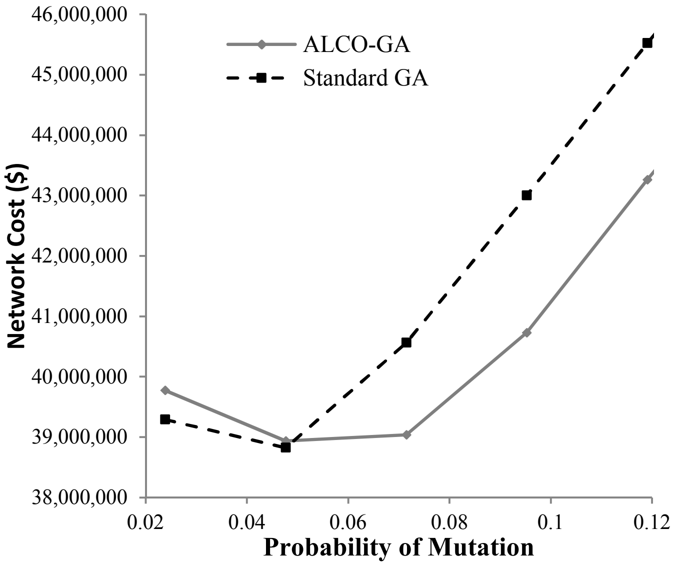

Figure 10. Mean performance of best solution for varying mutation rates for the New York Tunnels problem.

This is emulated in figure 11 which shows the performance of both algorithms in terms of average percentage of best knows solutions found from the 100 runs. ALCO-GA exhibits a greater ability to find the best known solution relative to that of the SGA, especially at mutation rates of ~ 0.05 and above.

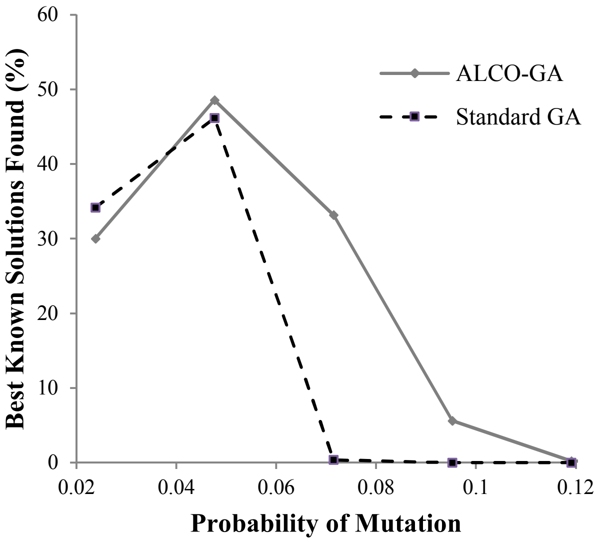

Figure 11. Average number of best known solutions found at varying mutation rates for the New York Tunnels problem

For the Two Loop and New York Tunnels benchmark problems, the performance of ALCO-GA is shown to be less sensitive to mutation rate variance than the SGA. ALCO-GA performs increasingly well compared to the SGA as the mutation rate is increased due to the resultant increased influence of the heuristic in the mutation operator.

For the Network B problem the tournament size was varied between 0.04N and 0.1N. For each tournament size, the per-bit mutation probability was varied between 0.0006 and 0.01. The remaining parameters where configured as before, apart from the run duration, which was increased to 150,000 iterations (300,000 evaluations). Due to the complexity and inherent run time, both the SGA and ALCO-GA where only run a total of 10 times per configuration.

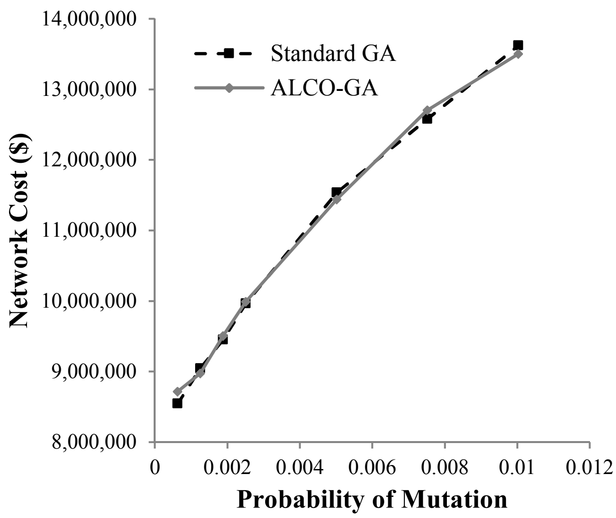

Figure 12. Mean performance of best solution for varying mutation rates for the Network B problem.

Figure 12 shows the performance of both the SGA and ALCO-GA with regard to mutation rate. It is apparent that both algorithms essentially display similar performance across the range of mutation rates, with the SGA performing best at lower mutation rates compared to ALCO-GA, which performs better at a higher mutation rate; although for this problem the difference in performance is negligible. An interesting observation is that the optimal pre-bit probability of mutation for Network B appears to be quite low compared to that of the other, smaller benchmark problems tested in this paper. However, if the probably of per-bit mutation is converted into actual mutations, the optimal number of mutations is very similar for all problems; with Two Loop, New York Tunnels and Network B having an optimal number of mutations equating to 6, 4 and 4 respectively. Therefore the results from this study suggest that regardless of problem complexity the optimal number of mutations applied to each new individual remains approximately the same.

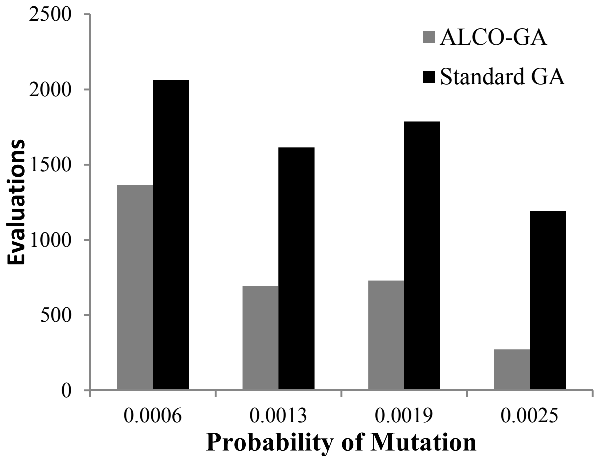

Figure 13. Average evaluations to feasible solution for varying mutation rates for Network B.

Although the resultant performance of both algorithms is similar with regard to best feasible solution for Network B, ALCO-GA achieves feasible solutions in fewer evaluations than the SGA. Figure 13 shows the average number of evaluations required for both algorithms to reach the feasible solution space for a range of per-bit mutation rates for Network B. The results indicate that the ability of ALCO-GA to find feasible solutions increases with mutation rate; this is again expected due to the increased influence the heuristic mutation operator has at higher rates of mutation.

Pumped Network – Anytown Variant

Although ALCO-GA was initially designed to solve single source gravity fed water distribution system problems, the algorithm is assessed here on a more complex type of problem involving multiple water sources and pump scheduling over an extended time period. The effect of ALCO-GA is expected to be less marked with this pumped example as the influence of pipe size is lessened due to the influence of tank placement and pumping regimes. The problem selected for this assessment was a modified version of the Anytown water distribution problem (Walski et al., 1987).

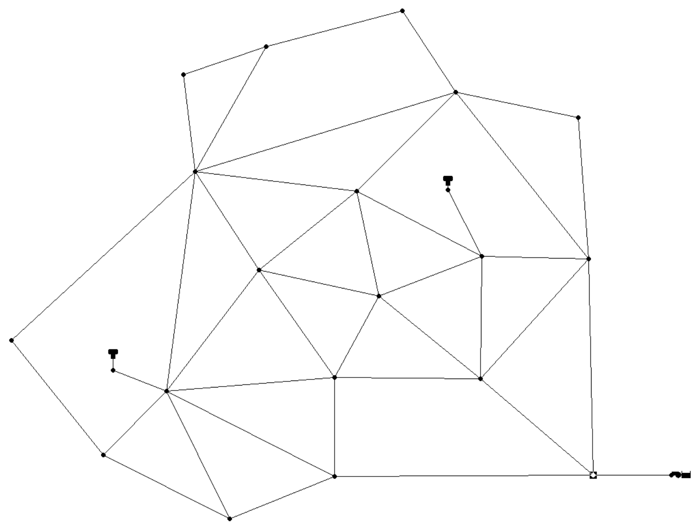

Figure 14. Anytown Network.

The Anytown water distribution problem used in this paper has been modified for ease of implementation; however the core objectives and characteristics have remained constant. The key objective of the Anytown water distribution problem is to find the most economically viable design that reinforces the existing system to meet projected demands, taking into account all capital expenditure and pumping costs. In our implementation of the problem, the options to duplicate existing pipes and clean and line existing pipes have been removed. Instead, each pipe decision in the network is treated as a new pipe. The costs of laying new pipes are given in the following table.

Table 4. New pipe cost for the Anytown Problem.

Pipe Diameter (in)	*New Pipes ($/ft)*
6	12.8
8	17.8
10	22.5
12	29.2
14	36.2
16	43.6
18	51.5
20	60.1
24	77.0
30	105.5

Each demand node is allocated an average day demand which varies throughout the day between factors of 0.6 and 1.3. A hydraulic pressure of 40 psi must be maintained at every demand node in the network throughout a 24 hour time period. In this simplification of the Anytown problem, instantaneous peak flow and fire flow network conditions are disregarded.

There is the option to place a tank at each network demand node. In this implementation of the problem there are two existing tanks in the network, which cannot be altered. All new tanks have identical parameters to the two existing tanks; 182,528 gal capacity (156250 effective) at a cost of $244,034 for each new tank.

The network is supplied by a water treatment plant and pumping station consisting of three pumps shown in the bottom right of figure 14. The water treatment plant maintains a fixed supply level of 10ft throughout the 24 hour operation cycle of the system. The number of operational pumps is defined at every 1 hour time step, contributing an additional 24 design variables into the problem. The pumping station operation costs are based upon the unit cost for energy which remains constant throughout the 24 hour cycle at $0.12/kWh. The total pumping energy cost is extrapolated over the period of 20 years.

Anytown Implementation

The implementation of ALCO-GA for the Anytown problem is similar to the gravity fed systems executed earlier in the paper, although some key changes needed to be applied to the algorithm. The representation of the network pipes is the same as before, however the chromosome has been extended to accommodate additional genes to define supplemental tank placement and pumping operations. The placement of additional tanks is represented in the chromosome by a 17 gene binary string, where each gene represents a potential tank location. If the gene value is 1 then a tank is placed at the relevant network node, if 0 then no tank is placed at the location. The 24 hour operation of the pumping station is defined be a 48 string consisting of 24 2-bit binary substrings where each substring represents a simulation time step and the binary value defines the number of pumps operating in that given time step.

In this implementation of the Anytown problem, all pipes in the network are regarded as new pipes, therefore disregarding the options for pipe duplication and the cleaning and lining of existing pipes in the original problem. The pipe cost function remains the same as in the previous implementations of the algorithm detailed in this paper. At each significant time step the hydraulic head deficit is measured at each node and the total head deficit of the network over the full 24 hour time period is used in combination with a penalty factor in the penalty function. The fitness function essentially remains unchanged, however in this implementation the cost of any additional tanks and the 20 year pumping costs are included within the network cost component of the objective function.

To identify the optimal operating envelope for the Anytown problem, the per-bit mutation probability was varied from ~0.006 – 0.09 for both algorithms. Due to the computational cost of the extended time period hydraulic simulations the algorithms were only run 20 times for each mutation rate for 100,000 iterations (200,000 solution evaluations). The tournament size remained constant at 5% of the population size throughout all runs. The results below show the mean best solutions from the 20 runs for both the standard GA and ALCO-GA for each mutation rate assessed.

Table 4. Computational Results of Standard GA and ALCO-GA for the Modified Anytown Problem

	Algorithm			
	Standard GA		***ALCO-GA***	
Per-Bit Mutation Probability	***Average Best Solution ($)***	***Standard Deviation ($)***	***Average Best Solution ($)***	***Standard Deviation ($)***
0.006	**5,438,320**	**239,725**	5,553,840	367,750
0.013	**5,209,510**	**244,092**	5,362,920	322,267
0.019	5,181,730	**241,172**	**5,159,220**	269,924
0.026	**5,197,780**	241,095	5,378,710	**237,649**
0.038	5,478,130	284,660	**5,397,220**	**230,452**
0.051	**5,733,320**	**158,546**	5,794,690	315,175
0.064	**6,659,650**	**314,648**	6,756,410	365,004
0.077	7,206,000	279,956	**6,881,940**	**236,581**
0.090	7,678,460	**277,392**	**7,513,990**	418,797

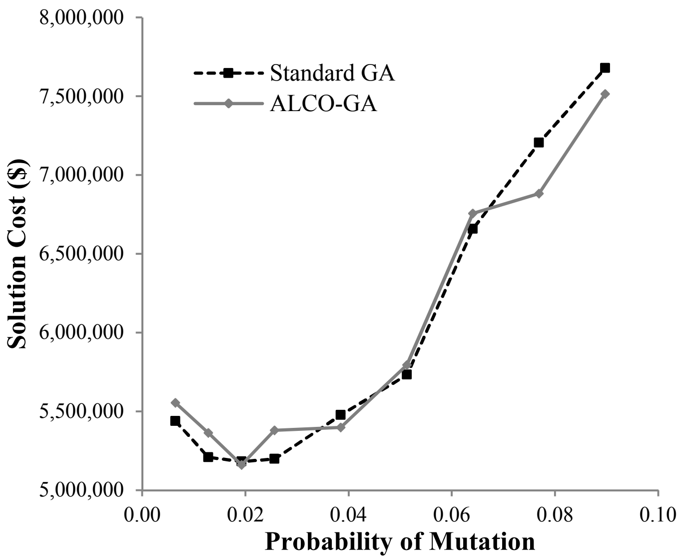

Figure 15. Mean performance of best solution for varying mutation rates for the Anytown Problem.

The performance of the standard GA and ALCO-GA are very similar at lower mutation rates although ALCO-GA does outperform the standard GA at higher mutation rates (P0.07 and above). The best performance for both algorithms is found at a gene mutation probability of 0.019, where ALCO-GA achieves a slightly improved result over that of the standard GA. Figure 16 shows the average performance of both the standard GA and ALCO-GA for the simplified Anytown Problem at a per-bit mutation probability of 0.019. The standard GA exhibits slightly faster convergence in the early stages of the search although after 18,000 iterations ALCO-GA matches the Standard GA's performance.

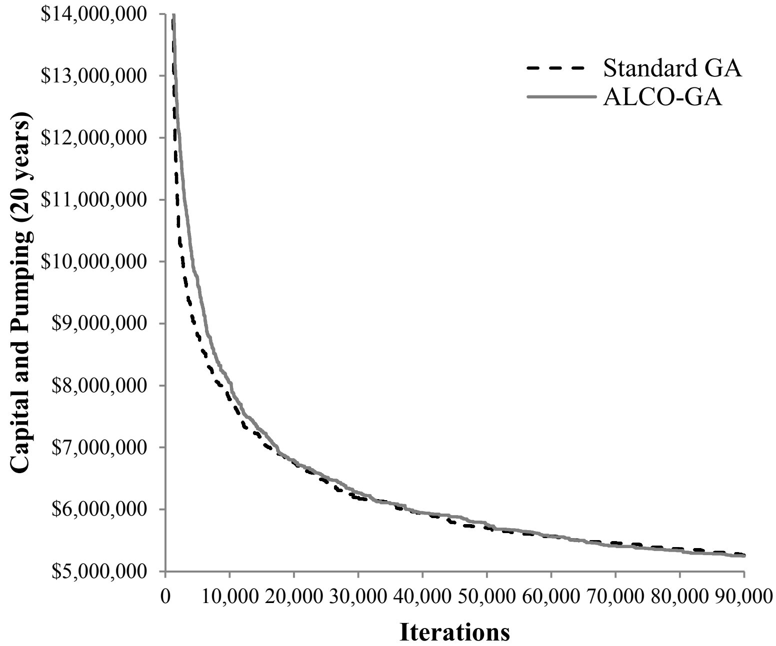

Figure 16. Average performance of best cost solutions for the Anytown Problem.

Out of the entire set of individual runs, ALCO-GA reaches a lower feasible solution to the modified Anytown problem than the standard GA (table 5).

Table 5. Best solutions achieved by the Standard GA and ALCO-GA for the Anytown Problem.

Algorithm	*Standard GA*	*ALCO-GA*
Best Solution ($)	4,857,690	**4,640,930**
Evaluations to Best Sol.	185,718	193,788

Although ALCO-GA achieves a better solution than that of the standard GA, the effect of the ALCO-GA heuristic on the search is somewhat muted due to the strong influence of the pumping operations and tank locations.

CONCLUSIONS

An Adaptive Locally Constrained Genetic Algorithm for WDN design optimisation problems has been described. The algorithm is based on a standard genetic algorithm with the addition of an adaptive heuristic based mutation operator which utilises hydraulic head information to guide the search of the algorithm into the feasible solution space. The new algorithm has been found to perform well on a range of WDNs of varying complexity compared to a standard GA; not only finding a better solution but achieving it in less solution evaluations and CPU time. It should be noted that the performance gains of ALCO-GA are largely located towards the initial stages of the search. The algorithm is able to generate feasible solutions more quickly than the standard GA and so in applications where the number of function evaluations is limited, i.e. where large real-world networks are optimised, ALCO-GA can provide valid designs much earlier in the evolutionary process. In particular, very complex networks with large objective functions will require an algorithm that is able to generate feasible solutions early in the optimisation process. ALCO-GA has demonstrated this property, and whilst the feasible solutions are reasonably far removed from being near-optimal, they do at least meet the engineering criteria. Additional experimentation has shown that the ALCO-GA is more robust to parameter settings than the standard GA meaning that if extensive parameter tuning is not feasible due to the complexity of the network, the rule-of-thumb parameters are more likely to function well with ALCO-GA than the standard GA. In short, ALCO-GA would be the algorithm of choice for an engineer wanting to optimise a large complex real-world network for the first time. Finally, the results of the parameter tuning provide some interesting results with respect to network complexity. Mutation rates are usually expressed as a percentage that refers to the probability of a variable being perturbed within a chromosome. As a percentage, this implies that larger chromosomes require more mutation than smaller ones. However this is shown not to be the case here. Ordinarily, a 5% mutation rate would see 1.6 mutations in 2 loop (32 bits), 4.2 mutations in New York Tunnels (84 bits) and 319.25 mutations in Network B (6,385 bits). However, the optimal numbers of mutations for each network are shown to be approximately 6 for Two Loop, 4 for New York Tunnels and 4 for Network B. These results suggest that a small number of mutations (e.g. < 10) are required to optimise problems regardless of the size of their chromosome and that the percentage model is not warranted for WDN problems.

ALCO-GA demonstrates the best mean performance throughout the runs of this simplified Anytown network. However, the results from the tuning show that this is to a certain extent dependent on the selection of the optimal mutation rate, and that for alternative mutation rates, there is no guarantee that ALCO-GA will improve on the performance of the standard GA. As expected the effect of the heuristic is diminished due to the reduction in influence from the pipe sizing on which ALCO-GA operates, the location of tanks and the operations of pumps will dilute the effect of ALCO-GA. We believe that Anytown represents a worst-case scenario for the algorithm as the number and length of pipes is small in comparison to the number of tanks and the effects of the operation of pumps. Larger real-world networks will typically have many more pipes outside of the trunk mains that will benefit from the ALCO-GA heuristic and results would be expected to be better on these. This experiment has demonstrated that the method can work on pumped systems and can achieve increases in performance, albeit somewhat marginal increases, over a standard GA.

FUTURE WORK

The ALCO-GA heuristic has been shown to improve performance on gravity-fed networks and to a somewhat lesser degree on the Anytown pumped network. Real world networks will include further complications above and beyond the pumped and EPS nature of Anytown, notably the inclusion of PRVs, FCVs, water treatment works and other network infrastructure. The ALCO heuristic could be incorporated into any optimisation that sizes pipes amongst other infrastructure in WDN.

However, the heuristic is currently restricted to pipe sizing and whilst pipes are among the most numerous assets represented by decision variables in a WDN, the influence of these additional infrastructure types is likely to weaken the effect of the heuristic. That said, there remains a good deal of scope in implementing ALCO-like rules within an EA for other elements of the network. Tanks for instance could be upsized if they were found to be overtopping and downsized if levels did not change sufficiently; PRV settings could be modified according to pressure at critical downstream nodes and pump settings could be incremented/decremented according to downstream head. ALCO-GA demonstrates that the principle of localised modification within a WDN can be a powerful addition to the global search of the evolutionary algorithm and future work could focus on developing a bespoke 'intelligent' operator for each asset type to achieve similar results on real-world networks.

REFERENCES

Alperovits, E., & Shamir, U. (1977). Design of optimal water distribution systems. *Water Resources Research*, *13*(6), 885. doi:10.1029/WR013i006p00885

Artina, S., Bragalli, C., Erbacci, G., Marchi, A., & Rivi, M. (2012). Contribution of parallel NSGA-II in optimal design of water distribution networks. *Journal of Hydroinformatics*, *14*(2), 310. doi:10.2166/hydro.2011.014

Bragalli, C., Ambrosio, C. D., Lee, J., Lodi, A., & Toth, P. (2008). *Water Network Design by MINLP* (Vol. 24495).

Bramlette, M. F. (1991). Initialization, Mutation and Selection Methods in Genetic Algorithms for Function Optimization. *Proceedings of the Fourth International Conference on Genetic Algorithms* (p. 100). Morgan Kaufmann Publishers.

Cunha, M. da C., & Sousa, J. (1999). Water Distribution Network Design Optimization: Simulated Annealing Approach. *Journal of Water Resources Planning and Management*, *125*(4), 215–221. Retrieved from http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(1999)125:4(215)

Dandy, G. C., Simpson, A. R., & Murphy, L. J. (1996). An Improved Genetic Algorithm for Pipe Network Optimization. *Water Resources Research*, *32*(2), 449. doi:10.1029/95WR02917

Davis, L. (1989). Adapting operator probabilities in genetic algorithms. *Third International Conference on Genetic Algorithms*. Morgan Kaufmann. Retrieved from http://www.citeulike.org/group/2892/article/1505552

De Jong, K. A. (1975). *Analysis of the behavior of a class of genetic adaptive systems*. University of Michigan. Retrieved from http://deepblue.lib.umich.edu/handle/2027.42/4507

Fujiwara, O., & Khang, D. B. (1991). Correction to "A Two-Phase Decomposition Method for Optimal Design of Looped Water Distribution Networks" by Okitsugu Fujiwara and Do Ba Khang. *Water Resources Research*, *27*(5), 985–986. doi:10.1029/91WR00368

Geem, Z., Kim, J., & Loganathan, G. (2002). Harmony search optimization: application to pipe network design. *International Journal of Simulation Modelling*, *22*(2), 125–133.

Grefenstette, J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, *16*(1), 122–128. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4075583

Kadu, M. S., Gupta, R., & Bhave, P. R. (2005). Optimal design of water distribution networks using genetic algorithm with reduction in search space. *One-Day National Conference Geotechnical and Environmentally Sustainable Development* (pp. 182–189). Nagpur, India.

Kessler, A., & Shamir, U. (1989). Analysis of the linear programming gradient method for optimal design of water supply networks. *Water Resources Research*, *25*(7), 1469. doi:10.1029/WR025i007p01469

Liong, S., & Atiquzzaman, M. (2004). Optimal design of water distribution network using shuffled complex evolution. *Journal of The Institution of Engineers, Singapore*, *44*(1), 93–107. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.7872&rep=rep1&type=pdf

Maier, H., Simpson, A., Zecchin, A., Foong, W., Phang, K., Seah, H., & Tan, C. (2003). Ant colony optimization for design of water distribution systems. *Journal of Water Resources Planning and Management*, *129*(June), 200–209. Retrieved from http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(2003)129%3A3(200)

Morales, A. K., & Quezada, C. V. (1998). A universal eclectic genetic algorithm for constrained optimization. *Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing* (pp. 518–522). Aachen, Germany.

Paechter, B., Rankin, R., Cumming, A., & Fogarty, T. (1998). Timetabling the classes of an entire university with an evolutionary algorithm. *5th International Conference on Parallel Problem Solving from Nature (PPSN V)* (pp. 865–874). Springer-Verlag Berlin.

Savic, D. A., & Walters, G. A. (1997). Genetic Algorithms for Least-Cost Design of Water Distribution Networks. *Journal of Water Resources Planning and Management*, *123*(2), 67–77. Retrieved from http://cedb.asce.org/cgi/WWWdisplay.cgi?104031

Savic, D., Walters, G., Smith, M., & Atkinson, R. M. (2000). Cost savings on large water distribution systems: design through genetic algorithm optimization. *Proc. 2000 ASCE EWRI Conference*. Minneapolis. Retrieved from http://ascelibrary.org/doi/pdf/10.1061/40517(2000)200

Schaake Jr., J. C., & Lai, D. (1969). *Linear Programming and Dynamic Programming Application to Water Distribution Network Design*. Cambridge, MA.

Sherali, H. D., Subramanian, S., & Loganathan, G. V. (2001). Effective Relaxations and Partitioning Schemes for Solving Water Distribution Network Design Problems to Global Optimality. *Journal of Global Optimization*, *19*(1), 1–26. doi:10.1023/A:1008368330827

Siedlecki, W., & Sklansky, J. (1989). Constrained genetic optimization via dynamic reward-penalty balancing and its use in pattern recognition. *Proceedings of the third international conference on genetic algorithms* (pp. 141–150). San Mateo, CA: George Mason University. Retrieved from http://dl.acm.org/citation.cfm?id=93126.93177

Smith, A., & Coit, D. (1997). Constraint handling techniques – penalty functions. In T. Baeck, D. . Fogel, & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*. Taylor & Francis. Retrieved from http://www.amazon.co.uk/Handbook-Evolutionary-Computation-Thomas-Baeck/dp/0750308958

Walski, T. M., Brill, E. D., Gessler, J., Goulter, I. C., Jeppson, R. M., Lansey, K., Lee, H.-L., et al. (1987). Battle of the network models:epilogue. *Journal of Water Resources Planning and Management*, *113*(2), 191–203.

Wu, Z., & Simpson, A. (2001). Competent genetic-evolutionary optimization of water distribution systems. *Journal of Computing in Civil Engineering*, *15*(April), 89–101. Retrieved from http://ascelibrary.org/doi/abs/10.1061/(ASCE)0887-3801(2001)15%3A2(89)

Yates, D. F., Templeman, A. B., & Boffey, T. B. (1984). The computational complexity of the problem of determining least capital cost designs for water supply networks. *Engineering Optimization*, *7*(2), 143–155. doi:10.1080/03052158408960635