Performance Study of Differential Evolution with Various Mutation Strategies Applied to Water Distribution System Optimization

Feifei Zheng[1], Angus R. Simpson[2] and Aaron Zecchin[3]

[1]School of civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, South Australia; email: fzheng@civeng.adelaide.edu.au
[2]School of civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, South Australia; email: asimpson@civeng.adelaide.edu.au
[3]School of civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, South Australia; email: azecchin@civeng.adelaide.edu.au

Abstract: Differential evolution (DE) is a relatively new optimization technique that has been employed to optimize the design of water distribution systems (WDSs). There are three important operators involved in the use of DE: mutation, crossover and selection. These operators are similar to the commonly used genetic algorithms (GAs). However, DE differs significantly from GAs in that mutation is an important operator for DE, while in contrast, crossover is an important operator for GAs. It has been found that the success of the DE algorithm in solving different mathematical optimization problems crucially depends on the mutation strategy that is used. This paper aims to investigate the relative effectiveness of five frequently used mutation strategies of DE when applied to WDS optimization. The five DE variants with different mutation strategies are applied to two well-known WDS case studies: the New York Tunnels Problem and the Hanoi Problem.

Keywords

Differential evolution, water distribution systems, mutation strategy, evolutionary algorithms

Introduction

The typical objective of optimal design for a water distribution network is to find the minimum cost network, while being able to satisfy the required water demands and head constraints at each node. Due to the nonlinear relationship between pipe discharge and head loss, and the availability of discrete pipe sizes, the optimal design of WDSs pose challenges for optimization tools. A number of deterministic optimization techniques such as linear programming and nonlinear programming have been used to tackle the WDS optimization problem (Alperovits and Shamir 1977; Fujiwara and Khang 1990). However, due to the multi-modal nature of WDS optimization problems, these deterministic approaches typically find only local optimal solutions, whereas the quality of the solution heavily depends on the initial starting point used for the optimization solver. In addition, the deterministic optimization methods such as linear and nonlinear programming are based on continuous search domains, which do not suit the discrete search space WDS optimization problem.

Evolutionary algorithms (EAs) have emerged as an alternative choice for the optimization of the design of WDSs. Additionally, EAs have been found to outperform deterministic methods as they are more likely to find the global optimal solution as they tackle the discrete search space directly. Over the last two decades, a number of EAs have been employed to optimize the design of WDSs. Murphy and Simpson (1992) were first to introduce genetic algorithms (GAs) for water network optimization; Cunha and Sousa (2001) used simulated annealing to optimize WDSs; Geem et al. (2002) developed a harmony search model for optimizing WDSs; Eusuff and Lansey (2003) proposed a shuffled frog leaping algorithm (SFLA) for network optimization; and Maier et al. (2003) applied an Ant Colony Optimization approach to optimize WDSs. More recently, Tolson et al. (2009) developed a new algorithm (the hybrid discrete dynamically dimensioned search algorithm HD-DDS) to optimize WDSs. These techniques have been successfully applied to a number of optimization problems and have been demonstrated to be more effective in finding optimal solutions compared with deterministic optimization techniques.

Differential evolution (DE), as a stochastic search method, was proposed by Storn and Price (1995). DE has been proven to be a simple but powerful evolutionary algorithm as it exhibits superior convergence properties when applied to a number of numerical case studies (Price et al. 2005). Vasan and Simonovic (2010) and Surbabu (2010) applied DE to the optimization of WDSs and concluded that the search ability of DE was at least as good as, if not better, than other EAs such as GAs and Ant Colony Optimization. Like other EAs, DE is a population-based search method that iteratively employs operators to guide the population towards the global optimum. There are three operators involved in DE: mutation, crossover and selection. The mutation operator is viewed as an important evolution mechanism for DE, (in contrast to crossover, which is considered a key operator within GAs). There are three parameters that need to be determined in the use of DE including population size (N), mutation strategy and its associated mutation weighting factor (F), and crossover rate (CR). It has been demonstrated that the performance of DE for numerical optimization problems is highly dependent on these control parameters and the mutation strategy used (Liu and Lampinen 2005). Liu and Lampinen (2005) claimed that DE, with a population size within the range of [3D, 10D] (D is the dimension of the optimization problem), F within the range of [0.5, 1.0], and CR within the range of [0.8, 1.0], generally showed good performance when applied to numerical optimization problems. Karaboga and Okdem (2004) introduced a DE with a dither mutation weighting factor (F_{dither}) and a dither crossover rate (CR_{dither}) such that the values of mutation weighting factor and crossover rate were randomized in a given range for each individual rather than specified particular values. Within this work, Karaboga and Okdem (2004) addressed that the dither DE outperformed the standard DE with a given fixed F and CR based on testing a number of numerical optimization problems. A total of five mutation strategies have been frequently used for DE application to diverse optimization fields (Price et al. 2005). However, there exists a lack of knowledge for determining which mutation strategy is preferable for different optimization problems.

The aim of this paper is to investigate the properties of the five mutation strategies applied to WDS optimization, thereby providing an effective guide for selecting an appropriate mutation strategy. In this study, for each case study, the population size (N) is set to be constant for the DE variants with different mutation strategies. A dither mutation weighting factor (F_{dither}) in the range of [0.5, 1.0] and a dither crossover rate (CR_{dither}) in the range of [0.8, 1.0] are used for DE variants with the different mutation strategies. For each DE variant, a total of 50 runs with different random number seeds are performed to enable comparison of the performance of convergence properties. These five mutation strategies and the dither mutation and crossover strategy are described in the next section.

The differential evolution algorithm

The differential evolution (DE) algorithm introduced by Storn and Price (1995) is a simple, yet powerful, EA for global optimization. The process of basic DE (Storn and Price 1995) is outlined as follows:

Initialization

An initial population is required to start the DE search. Normally, the initial population is generated by uniformly randomizing individuals within the search space as

$$x_{i,0}^{j} = x_{\min}^{j} + U(x_{\max}^{j} - x_{\min}^{j}) \quad i=1, 2, \ldots N, j=1, 2, \ldots D \tag{1}$$

where $x_{i,0}^{j}$ represents the initial value of the j^{th} parameter for the i^{th} individual in the initial population. The symbols $x_{\min}^{j}$ and $x_{\max}^{j}$ are the minimum and maximum bounds of the j^{th} parameter, U represents a uniformly distributed random variable in the range [0, 1], and N and D are the population size and dimension of the vector, respectively. The population size is not changed during the DE evolution process.

Mutation

DE is mainly defined by its mutation approach, compared with GAs, in that a mutant vector $V_{i,G}$, with respect to each individual $X_{i,G}$, is produced by adding the weighted difference (with weight F) between several random population members to a third member from the current population. Each individual $X_{i,G}$ associated with a mutant vector is denoted as target vector. A total of five frequently used mutation strategies in DE are given as:

(1) DE1-Rand1:

$$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \tag{2}$$

where $V_{i,G}$ is the mutant vector with respect to the target vector of $X_{i,G}$ at generation G. $X_{r1,G}$, $X_{r2,G}$ and $X_{r3,G}$ are three vectors randomly selected from the current population *G*. As shown in Equation (2), DE1 generates a mutant vector $V_{i,G}$ for each vector *i* by adding the weighted difference of two randomly selected vectors to a third vector. The random integers *r1*, *r2* and *r3* are different values from the population of size *N*. *F* is the weighted difference factor within the range [0, 1].

(2) DE2-Best1:

$$V_{i,G} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) \tag{3}$$

DE2 is similar to DE1 in terms of producing the mutant vector except that the third vector that is to be perturbed is the best individual of the current generation ($X_{best,G}$).

(3) DE3-Best2:

$$V_{i,G} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G}) \tag{4}$$

where $X_{best,G}$ is the best individual of the current generation G. DE3 uses two weighted differences of four randomly selected individuals and the best individual to produce the mutant vector. The random integers *r1*, *r2*, *r3* and *r4* are different values chosen from the population of size *N*.

(4) DE4-CurrentToBest2:

$$V_{i,G} = X_{r1,G} + F(X_{best,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G}) \tag{5}$$

Like DE3, DE4 also employs two weighted difference individuals, but one is the weighted difference between the best individual and a random individual. In addition, for DE4, the individual to be perturbed is a random individual rather than the best individual that used in DE3.

(5) DE5-rand2:

$$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) + F(X_{r4,G} - X_{r5,G}) \tag{6}$$

DE5 is quite similar to DE3, only differing in that the individual to be perturbed is a random individual from the population for DE5, while the individual to be perturbed is the current best individual for DE3. The random integers *r1*, *r2*, *r3*, *r4* and *r5* are different values randomly selected from the population of size *N*. The search performance of DE variants with these five mutation strategies are investigated in this study.

Crossover

After mutation, a trial vector $U_{i,G}$ is generated though selecting solution component values either from $X_{i,G}$ or $V_{i,G}$. In the basic DE version (Storn and Price 1995), uniform crossover is employed as:

$$u_{i,G}^{j}=\begin{cases} v_{i,G}^{j}, & \text{if } U_{j}\leq CR \\ x_{i,G}^{j}, & \text{otherwise} \end{cases} \tag{7}$$

where $u_{i,G}^{j}$, $v_{i,G}^{j}$, $x_{i,G}^{j}$ are the j^{th} parameter in the i^{th} trial vector, mutant vector and target vector respectively, and U_j is a uniformly distributed random number between 0 and 1. If U_j is smaller than CR ($0\leq CR \leq 1$), the value $v_{i,G}^{j}$ in the mutant vector is copied to the trial vector. Otherwise, the value $x_{i,G}^{j}$ in the target vector is copied to the trial vector.

Selection

After crossover, all the trial vectors are evaluated using the objective function $f(U_{i,G})$ and compared with their corresponding trial vectors objective function $f(X_{i,G})$. The vector with a lower objective function value (assuming a minimization problem) survives for the next generation. That is

$$X_{i,G+1}=\begin{cases} U_{i,G}, & \text{if } f(U_{i,G})\leq f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \tag{8}$$

Mutation, crossover and selection are repeated generation by generation until the stopping criteria is satisfied. The basic DE is a continuous global optimization search algorithm. Therefore, the DE is required to be modified to deal with a discrete WDS optimization problem. In this study, the continuous pipe sizes are rounded to a predetermined discrete set of pipe diameters (*i.e.* 1, 2, …, *n* where *n* is the number of discrete diameter sizes) after the mutation operator. For each vector element $x_{i,G}^{j}$, if its value is smaller or larger than the minimum or maximum allowable pipe size, the minimum or maximum allowable pipe size is given for its discrete value. If its value is between two sequentially discrete pipe diameters, the pipe diameter that is closer to the continuous value is given as its discrete value. Constraint tournament selection is used in DE to handle pressure constraints (Deb 2000).

Dither mutation and crossover factor

The dither mutation and dither crossover factor introduced by Karaboga and Okdem (2004) was randomized according to:

$$F_{dither}=F_{l}+U_{i}^{F}(F_{h}-F_{l}) \tag{9}$$

$$CR_{dither}=CR_{l}+U_{i}^{CR}(CR_{h}-CR_{l}) \tag{10}$$

where F_h and F_l are the maximum and minimum values of mutation factor as specified by the user. CR_h and CR_l are the maximum and minimum values of crossover factor as specified by the user. U_i^F and U_i^{CR} are random numbers between 0 and 1 generated for each vector *i* for determining F_{dither} and CR_{dither} values respectively. The dither mutation and crossover strategy have been employed in the research presented in this paper.

Case studies

Case Study 1: New York Tunnels Problem

The NYTP network has 21 existing tunnels and 20 nodes fed by the fixed-head reservoir as shown in Fig. 1. All the details of this network including the head constraints, pipe costs and water demands can be found in Dandy et al. 1996. The objective of this case study is to determine which of the least cost set of tunnels should be installed in parallel with the existing tunnels while satisfying the minimum head requirement at all nodes. There are 15 pipe diameters that can be selected for the NYTP. In addition, a zero tunnel size provides a total of 16 options (15 actual pipe diameters plus a zero tunnel size) for each link. Thus the total search space is 16^{21} (approximately 1.934×10^{25}).

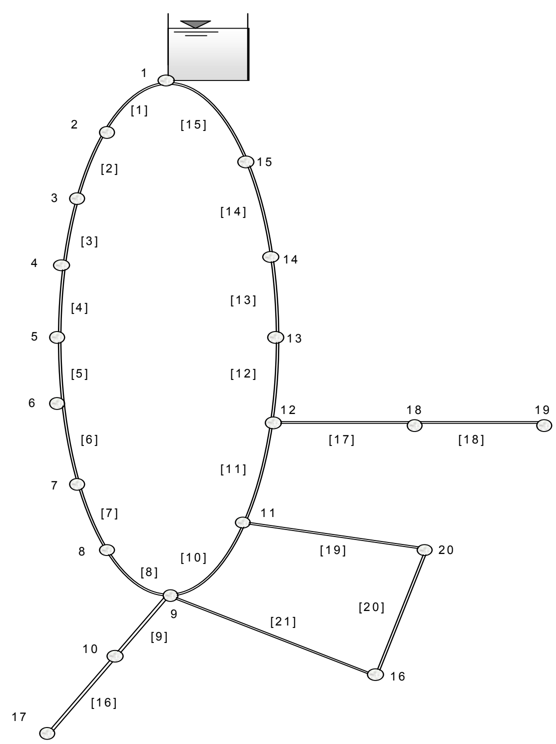

Figure 1 The layout of the New York Tunnel network

Case Study 2: Hanoi Problem

The Hanoi Problem (HP) is a network design where all new pipes are to be selected. The network is composed of 34 pipes and 32 nodes which are fed by a single reservoir with the head of 100 meters as shown in Fig. 2. The minimum head requirement of the other nodes is 30 meters. A total of six pipe diameters of {12, 16, 20, 24, 30, 40} inches is selected for each new pipe. The total search space is $6^{34} \approx 2.8651 \times 10^{26}$. The details of this network and the formulation of the cost for pipes are found in Fujiwara and Khang (1990).

DE variants with five different mutation strategies are applied to these two case studies. The population size for DE applications to these two case studies is 100, which is in the range of [3D, 10D] as D is 21 and 34 for the NYTP and HP case studies respectively. The dither mutation factor and crossover factor are given the range of [0.5, 1.0] and [0.8, 1.0] respectively, which was proposed by Liu and Lampinen (2005). The maximum number of allowable evaluations for each of these two case studies is equal to 300,000. For each DE variant with the different mutation strategies, a total of 50 runs with different random number seeds have been performed.

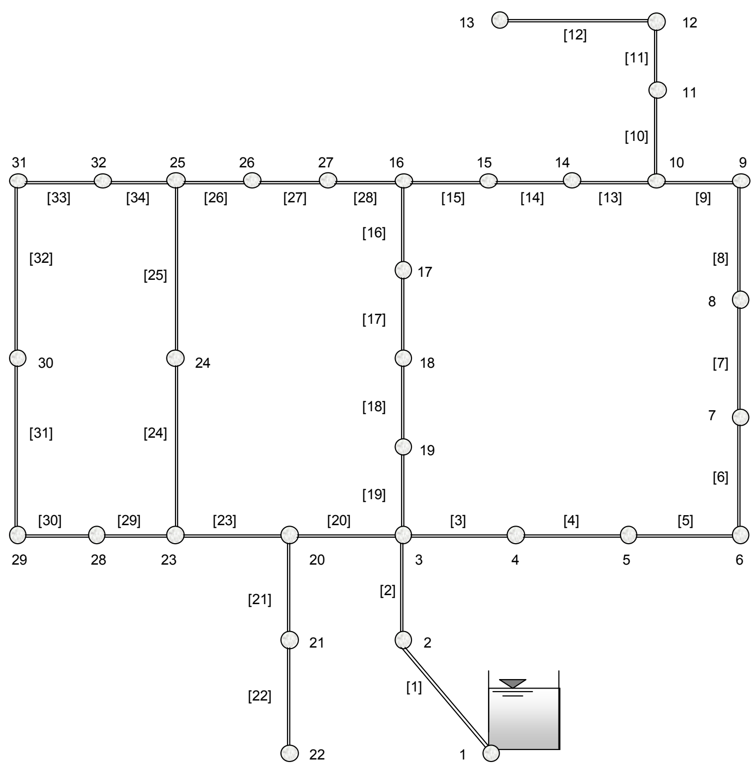

Figure 2 The layout of the Hanoi problem network

Results and discussion

In order to enable the assessment of the performance of DE with the different mutation strategies, a number of statistics have been determined for each of the DE variants applied to these two case studies. The statistics include the best solution found, the number of times the best solution was found, the average cost solution and average evaluations required to find the best solution based on 50 different runs.

New York Tunnels Problem

Table 1 gives the statistics of DE variants with the different mutation strategies applied to the NYTP case study. The current best known solution for the NYTP case study was first found by Maier et al. (2003) with a cost of $38.64 million. It is shown that this current best known solution was found by every DE variant with the five different mutation strategies in this study. As can be seen from Table 1, DE5 (Equation (6)) performed the best as it found the best known solution for the NYTP case study with a frequency of 96%, which is the highest among the five DE variants. However, DE5 required an average of 184,966 evaluations based on 50 different runs to converge, which is far more than that of other four DE variants as shown in Table 1. DE1 (Equation (2)) and DE4 (Equation (5)) found the best known solution with a frequency of 86%, which is lower than DE5, while being significantly more computationally efficient. This is evidenced by the fact that DE1 and DE4 converged at only 40,726 and 44,152 evaluations, which is 22% and 24% of that required by DE5. DE2 (Equation (3)) and DE3 (Equation (4)) performed the worst in terms of times that the best solution was found based on a total of 50 different runs, while converging the fastest as shown in Table 1. This can be explained in that it appears that the DE2 and DE3 were more likely to be trapped by local optimal solutions and hence result in premature convergence.

Table 1 also gives the statistical results of HD-DDS (Tolson et al. 2009), GA (Dandy et al. 2010) and particle swarm optimization (PSO) (Dandy et al. 2010) applied to the NYTP case study. It is observed from Table 1 that DE1, DE4 and DE5 performed at least as well as, if not better than HD-DDS, GA and PSO in terms of frequency with which the best known solution was found. In terms of efficiency, DE1 and DE4 performed similarly with HD-DDS (Tolson et al. 2009).

Table 1 Performance comparison of DE variants with different mutation strategy applied to the NYTP case study

Algorithms	Number of different runs	Best solution ($M)	Times with best solution found (percent of trials)	Average cost ($M)	Maximum allowable evaluations	Average evaluations required to find the best solutions
DE1	50	38.64	43 (86%)	38.68	300,000	40,726
DE2	50	38.64	6 (12%)	39.70	300,000	3,726
DE3	50	38.64	28 (56%)	38.76	300,000	24,462
DE4	50	38.64	43 (86%)	38.68	300,000	44,152
DE5	50	38.64	48 (96%)	38.64	300,000	184,966
HD-DDS[1]	50	38.64	43 (86%)	38.68	50,000	50,000
GA[2]	30	38.64	21 (70%)	38.78	300,000	NA
PSO[2]	30	38.64	12 (40%)	38.85	300,000	NA

[1]Tolson et al. (2009). [2] Dandy et al. (2010). NA means not available.

The Hanoi Problem

The statistical results for the HP case study are given in Table 2. The current best known solution found for the HP case study was first reported by Reca and Martínez (2006) with a cost of $6.081 million. It has been found that, DE1 (Equation (2)), DE3 (Equation (4)) and DE4 (Equation (5)) exhibited a similar performance in terms of times of finding the best known solution, while DE1 was able to converge far faster than DE3 and DE4 as shown in Table 2. In terms of the average cost solution, DE1 performed the best as it generated the lowest average cost solution with a value of $6.088 million, which deviated only 0.12% from the best known solution for the HP case study. It was found that the DE5 performed the worst for the HP case study. As can be seen from Table 2, the best solution found by DE5 (Equation (6)) was $6.109 million, which is 0.46% higher than the best known solution of the HP case study. In addition, DE5 required the most evaluations to converge as shown in Table 2, showing the slowest convergence speed.

For the HP case study, it is observed that all the five DE variants outperformed the GA (Dandy et al. 2010) and PSO (Dandy et al. 2010) in terms of best solution found and average cost solution found based on a range of trials with different random number seeds. DE1, DE3 and DE4 significantly outperformed HD-DDS (Tolson et al. 2009) in terms of frequency with which the best solution was found and the average cost solution based on 50 different trials.

Table 2 Performance comparison of DE variants with different mutation strategy applied to the HP case study

Algorithms	Number of different runs	Best solution ($M)	Times with best solution found (percent of trials)	Average cost ($M)	Maximum allowable evaluations	Average evaluations required to find the best solutions
DE1	50	6.081	43 (86%)	6.088	300,000	74,584
DE2	50	6.081	2 (4%)	6.240	300,000	6,660
DE3	50	6.081	42 (84%)	6.109	300,000	195,872
DE4	50	6.081	42 (84%)	6.100	300,000	189,432
DE5	50	6.109	0 (0%)	6.180	300,000	283,454
HD-DDS[1]	50	6.081	4 (8%)	6.252	100,000	100,000
GA[2]	30	6.167	0 (0%)	6.277	300,000	NA
PSO[2]	30	6.373	0 (0%)	6.483	300,000	NA

[1]Tolson et al. (2009). [2] Dandy et al. (2010). NA means not available.

Conclusions

Analyzing the results obtained from these two case studies, it has been found that the DE1, (which generated the mutant vector using three different random vectors from the current population) consistently performed well overall in terms of robustness and efficiency. DE4 (that produced the mutant vector using two weighted differences and one is the difference between the best individual and a random individual) exhibited similar performance with DE1 in terms of robustness, while converging slower than DE1. It is also interesting to note that the DE5 (Equation (6)) performed the best on the NYTP case study, while performing the worst on the HP case study in terms of search ability. This can be explained in that the search ability of DE5 is sensitive to the fitness landscape of the problems being optimized. DE2 (Equation (3)) was generally found to show the worst performance on these two case studies.

Based on this study, it is concluded that the DE1 and DE4 are more suitable for water distribution network optimization when compared with other three DE variants including DE2, DE3 (Equation (4)) and DE5. In addition, DE1 and DE4 outperformed other optimization techniques, such as HD-DDS (Tolson et al. 2009), GA and PSO (Dandy et al. 2010) in terms of robustness and efficiency. Thus, DE1 and DE4 can be viewed as being well–suited to water distribution system optimization.

Reference

Alperovits, E., and Shamir, U. (1977). “Design of Water Distribution Systems.” *Water Resources Research*, 13(6), 885-900.

Cunha, M. C., and Sousa, J. (2001). “Hydraulic Infrastructures Design Using Simulated Annealing.” *Journal of Infrastructure Systems*, 7(1), 32-38.

Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). “An improved genetic algorithm for pipe network optimization.” *Water Resources Research*, 32(2), 449-457.

Dandy, G. C., Wilkins, A., and Rohrlach, H. (2010). “A methodology for comparing evolutionary algorithms for optimizing water distribution systems.” *Proceedings of the 12th Water Distribution System Analysis Symposium*, Tucson, Arizona

Deb, K. (2000). “An efficient constraint handling method for genetic algorithms.” *J Comput. Methods Appl. Mech. Engrg*., 186,311-338.

Eusuff, M. M., and Lansey, K. E. (2003). “Optimisation of water distribution network design using shuffled frog leaping algorithm.” *J. Water Resources Planning and Management*, 129(3), 210-225.

Fujiwara, O., and Khang, D. B. (1990). “A two-phase decomposition method for optimal design of looped water distribution networks.” *Water Resources Res*earch, 23(6), 977-982.

Geem, Z. W., Kim, J. H., and Loganathan, G. V. (2002). “Harmony search optimization: application to pipe network design.” *Int. J. Model. Simulat*., 22(2), 125-133.

Karaboga, D., and Okdem, S. (2004). “A simple and global optimization algorithm for engineering problems: differential evolution algorithm.” *Turkish Journal of Electrical Engineering & Computing Science*, 12(1), 53-60.

Liu, J., and Lampinen, J. (2005). “A Fuzzy Adaptive Differential Evolution Algorithm.” *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 9(6), 448- 462.

Maier, H. R., et al. (2003). “Ant colony optimization for the design of water distribution systems.” *J. Water Resources Planning and Management*, 129(3), 200-209.

Murphy, L. J., and Simpson, A. R. (1992). “Pipe optimization using genetic algorithms.” *Research Report No. R93*, Department of Civil Engineering, University of Adelaide.

Price, V. K., Storn, M. R., and Lampinen, A. J. (2005). *Differential Evolution: A practical approach to global optimization*, Springer, Berlin.

Reca, J., and Martinez, J. (2006) “Genetic Algorithms foe the design of looped irrigation water distribution networks.” *Water Resources Research*, 42, W05416,doi:10.1029/2005WR004383.

Storn, R., and Price, K. (1995). “Differential Evolution-A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Space.” *Technical Report*, International Computer Science Institute, Berkeley, CA.

Suribabu, C. R. (2010). “Differential evolution algorithm for optimal design of water distribution networks.” *Journal of Hydroinformatics*, 12 (1), 66-82.

Tolson, B. A., et al. (2009). “Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization.” *Water Resources Res*earch, 45, W12416, doi:10.1029/2008WR007673.

Vasan, A., and Simonovic, S. P. (2010). “Optimization of Water Distribution Netowrk Design using Differential Evolution.” *J. Water Resource Planning and Management*, 136(2), 279-287.