Zheng, Feifei; Zecchin, Aaron Carlo; Simpson, Angus Ross
Self-adaptive differential evolution algorithm applied to water distribution system optimization
Journal of Computing in Civil Engineering, 2013; 27(2):148-158

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

A self-adaptive differential evolution algorithm applied to water distribution system optimization

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

Abstract: Differential evolution (DE) is a relatively new technique that has recently been used to optimize the design for water distribution systems (WDSs). Several parameters need to be determined in the use of DE, including: population size, *N*; mutation weighting factor, *F*; crossover rate, *CR* and a particular mutation strategy. It has been demonstrated that the search behavior of DE is especially sensitive to the *F* and *CR* values. These parameters need to be fine-tuned for different optimization problems as they are generally problem-dependent. A self-adaptive differential evolution (SADE) algorithm is proposed to optimize the design of WDSs. Three new contributions are included in the proposed SADE algorithm: (i) instead of pre-specification, the control parameters of *F* and *CR* are encoded into the chromosome of the SADE algorithm and hence are adapted by means of evolution; (ii) *F* and *CR* values of the SADE algorithm apply at the individual level rather than the generational level normally used by the traditional DE algorithm; and (iii) a new convergence criterion is proposed for the SADE algorithm as the termination condition, thereby avoiding pre-specifying a fixed number of generations or computational budget to terminate the evolution. Four WDS case studies have been used to demonstrate the effectiveness of the proposed SADE algorithm. The results obtained show that the proposed algorithm exhibits good performance in terms of solution quality and efficiency.

[1]PhD student, School of Civil, Environmental and Mining Engineering, University of Adelaide, Adelaide, South Australia, 5005, Australia. fzheng@civeng.adelaide.edu.au
[2]Lecturer, School of Civil, Environmental and Mining Engineering, University of Adelaide, Adelaide, South Australia, 5005, Australia azecchin@civeng.adelaide.edu.au
[3]Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide, Adelaide, South Australia, 5005, Australia. asimpson@civeng.adelaide.edu.au

The advantage of the proposed SADE algorithm is that it reduces the effort required to fine-tune algorithm parameter values.

CE Database subject headings: optimization; water distribution systems; differential evolution.

Author Keywords: optimization; differential evolution; water distribution systems.

INTRODUCTION

Water distribution systems (WDSs) are one of the most expensive public infrastructure works as they require a high level of capital investment for construction and a continuing investment for maintenance. Research into the optimal design of WDSs is motivated, therefore, by the possibility of substantial cost savings. The optimal design of a WDS involves indentifying the lowest cost pipe network that is able to provide the required demand and head pressure for each individual supply node. The design of WDSs poses challenges for optimization tools for two main reasons: (i) the nonlinear relationships between pipe discharges and head losses introduce complex nonlinear constraints into the optimization problem, and (ii) the discrete pipe diameters lead to a combinatorial optimization problem.

Historically, a number of traditional optimization techniques have been applied to water network optimal design, such as linear programming (Alperovits and Shamir 1977; Quindry et al. 1981; Fujiwara et al. 1987) and non-linear programming (Lansey and Mays 1989; Fujiwara and Khang 1990). However, due to the multi-modal nature of the fitness landscape for the optimization of water distribution system problem, these methods are more likely to converge on local optimal solutions, where the final solutions are highly

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

sensitive to the initial starting point (Eiger et al. 1994). In addition, the final solutions may include continuous pipe sizes or split pipes, which is a significant practical limitation.

Evolutionary algorithms (EAs) have been popular alternatives for optimizing WDS designs as they are able to handle a discrete search space directly, and are less likely to be trapped by local optimal solutions. The search strategy of EAs differs from the traditional optimization techniques, such as linear programming or non-linear programming, in that they explore broadly across the search space using a population-based stochastic evolution algorithm, where no gradient information is required.

Over the last two decades, a number of EAs have been employed to optimize the design of WDSs, such as genetic algorithms (Murphy and Simpson 1992; Simpson et al. 1994; Dandy et al. 1996; Savic and Walters 1997); simulated annealing (Cunha and Sousa 2001); harmony search (Geem et al. 2002); shuffled frog leaping algorithm (Eusuff and Lansey 2003); Ant Colony Optimization (Maier et al. 2003); particle swarm optimization (Suribabu and Neelakantan 2006); cross entropy (Perelman and Ostfeld, 2007); and scatter search (Lin et al. 2007). These techniques have been successfully applied to a number of WDS optimization problems and have been demonstrated to be more effective in finding optimal solutions compared to traditional optimization techniques. It has been noticed that the performance of all these EAs, in terms of robustness and efficiency, are significantly affected by the algorithm parameter settings, which need to be adjusted for different optimization problems. It has been reported by Tolson et al. (2009) that the number of parameters that need to be fine-tuned for different optimization problems for these EAs varies from 3 to 8. These do not include a termination criterion parameter that also needs to be pre-specified to end the EA run (i.e.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

normally the maximum number of allowable evaluations or generations). The appropriate parameters of EAs are varied for different optimization problems and normally are adjusted by trial and error. Thus, it is extremely computationally expensive to determine the proper parameter values for a newly given WDS case study.

Differential evolution (DE), proposed by Storn and Price (1995), has recently been used to optimize WDSs (Suribabu 2010; Dandy et al. 2010). There are three important operators involved in the application of the DE algorithm: a mutation operator, a crossover operator and a selection operator. These operators are similar to a genetic algorithm (GA), but DE algorithms differ significantly from a GA in the mutation process, in that the mutant solution is generated by adding the weighted difference between two random population members to a third member.

A total of four parameters need to be pre-determined in the use of DE, including: population size, N; mutation weighting factor, F; crossover rate, CR; and a particular mutation strategy. It has been demonstrated that the performance of DE is governed by these parameters (especially the F and CR) based on a number of numerical optimization case studies (Storn and Price 1995; Vesterstrom and Thomsen 2004). In terms of optimizing WDSs, Suribabu (2010) and Vasan and Simonovic (2010) concluded that the performance of DE algorithms was at least as good as, if not better, than other EAs such as GAs and Ant Colony Optimization. While Dandy et al. (2010) has stated that GAs give better results overall than DE algorithms in terms of solution quality and efficiency. The contradiction of results reported by Suribabu (2010) and Dandy et al. (2010) can be explained by the fact that the different parameter values including N, F and CR are used in these DE applications. In addition, Suribabu (2010) investigated the effectiveness of

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

the DE using a number of different *F* and *CR* combinations (*N* is constant) applied to WDS optimization problems. His results show that the performance of the DE algorithm applied to the WDS optimization is highly dependent on the parameter values selected. As these control parameters are problem dependent, using the DE algorithm effectively is time consuming since appropriate parameter values have to be established for each new WDS case study.

Investigations have been undertaken to avoid pre-specifying parameter values in EAs. Bäck et al. (1991) initially introduced a self-adaptive algorithm to dynamically adjust the mutation probability in the evolution strategy. Eiben et al. (1999) gave a systematic analysis of a self-adaptation strategy for the parameters of EAs. Wu and Simpson (2002) and Wu and Walski (2005) proposed a self-adaptive penalty approach GA for pipeline optimization. The penalty multiplier was encoded onto each member of the population, thereby allowing the penalty multiplier to evolve over the course of the GA optimization. Thus, there was no need to pre-specify a penalty multiplier before performing the GA run. Gibbs et al. (2010) provided an estimate of population size for GA applications based on the genetic drift. Tolson et al (2009) developed a hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for WDS optimization and proposed the HD-DDS as a parameter-setting-free algorithm. Geem and Sim (2010) proposed a parameter-setting-free harmony search algorithm to optimize the design of WDSs.

Brest at al. (2006) proposed a self-adaptive strategy to evolve the *F* and *CR* values of the DE algorithm, which is called jDE. In the jDE algorithm, the *F* and *CR* values were adjusted by introducing two new parameters τ_1 and τ_2. They concluded that the self-adaptive DE algorithm performed better than the traditional DE algorithm in terms of

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

convergence speed and final solution quality based on testing a number of numerical benchmark optimization problems.

In this paper, a new self-adaptive differential evolution (SADE) algorithm is proposed. A total of thee novel aspects are involved in the proposed SADE algorithm, which are (i) control parameters of F and CR are encoded into the chromosome of the SADE algorithm rather than pre-specification and hence are adapted by means of evolution; (ii) F and CR values of the SADE algorithm apply at the individual level, which differs to the traditional DE algorithm that F and CR values applied at the generational level; and (iii) a new convergence criterion is proposed for the SADE algorithm as the termination condition in order to avoid pre-specifying a fixed number of generations or evaluations to terminate the evolution.

The F and CR are encoded into the solution string and hence are subject to evolution in the proposed SADE algorithm. Each individual in the initial population is assigned with randomly generated F and CR values within a given range. The better values of F and CR that produce fitter offspring are directly passed onto the next generation. If the F and CR values are unable to yield better offspring, these two values are randomly regenerated within the given range for the next generation. This newly proposed SADE differs with the jDE algorithm (Brest et al. 2006). For the jDE algorithm used in Brest et al. (2006), the F and CR values survive to the next generation with a particular probability τ_1 and τ_2 ($0 < \tau_1, \tau_2 < 1$) respectively. With a probability of $1-\tau_1$ and $1-\tau_2$, the F and CR values are randomly re-initialized to new values within the given range for the next generation respectively. The τ_1 and τ_2 values need to be pre-specified and tuned for different optimization problems and hence two new parameters were introduced in the jDE

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

algorithm proposed by Brest at al. (2006). The self-adaptive strategy proposed in this paper allows the *F* and *CR* values that are able to yield fitter offspring are more likely to survive longer over generations during the running of the algorithm, which in turn, generates further better offspring. The details of the proposed SADE algorithm are presented in this paper.

The *F* and *CR* values in traditional DE algorithms (Storn and Price 1995) and the DE algorithms applied to the WDS optimization (Suribabu 2010; Dandy et al. 2010; Zheng et al. 2011) are typically applied at the generation level during optimization. This implies all the individuals are therefore subject to identical mutation weighting and crossover strength. As with Brest et al. (2006), the *F* and *CR* values in the proposed SADE algorithm are applied at the individual level and hence different individuals within a population may have different mutation weightings and crossover rates applied. This approach was motivated by the fact that different individuals in a generation will be at varying distances from the optimal solutions and therefore require different mutation and crossover strength. For the individuals at greater distances from the optimal solutions, a relatively large *F* and *CR* is probably appropriate, while in contrast, for the individuals at relatively short distances from the optimal solutions, a relatively smaller *F* and *CR* may be suitable. Thus, the search performance of the proposed SADE algorithm is expected to improve as different individuals are associated with different *F* and *CR* values by means of evolution.

For EAs, the convergence condition is usually a fixed number of generations reached (limit of computational budget) or a predefined small value reached between two consecutive generations in terms of objective function values (Deb 2001). In the case of

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

WDS optimization problems, the maximum number of allowable evaluations or generations is normally used as the termination condition (Savic and Walters 1997; Tolson et al. 2009; Suribabu 2010; Dandy et al. 2010). However, the appropriate number of allowable evaluations or generations is optimization problem-dependent and hence generally determined by trial and error. Moreover, the evolution time to reach the same final solutions of EAs applied to the same optimization problem with different starting points is also different. This unavoidably results in computational waste when the budget is greater than required or computational insufficiency when the budget is smaller than required. In addition to the self-adaptive strategy, a new convergence criterion is proposed in this paper for the SADE algorithm to eliminate the need to preset the computational budget and thereby avoid computational excess or insufficiency. The details of the proposed convergence criterion are given in the next section.

SELF-ADAPTIVE DIFFERENTIAL EVOLUTION

Figure 1 illustrates the flowchart of the proposed SADE algorithm to be discussed in the following sections.

Initialization

The SADE algorithm is a population based stochastic search technique. Thus, an initial population is required to start the DE algorithm search. Normally, each initial population $X_{i,0}$={ $x_{i,0}^1$, $x_{i,0}^2$,……… $x_{i,0}^D$ } is generated by uniformly randomizing individuals within the search space. In addition, initial values of the mutation factor *F* and crossover rate *CR* are randomly generated within a given range for each initial individual real-valued string. The initialization rule is given by:

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

$$x_{i,0}^{j} = x_{\min}^{j} + Rand_1(x_{\max}^{j} - x_{\min}^{j}) \quad i=1, 2, \ldots.N, j=1, 2, \ldots.D$$

$$F_{i,0} = F_l + Rand_2(F_u - F_l) \quad (1)$$

$$CR_{i,0} = CR_l + Rand_3(CR_u - CR_l)$$

where $x_{i,0}^{j}$ represents the initial value of the j^{th} parameter in the i^{th} individual at the initial population; $x_{\min}^{j}$ and $x_{\max}^{j}$ are the minimum and maximum bounds of the j^{th} parameter; $F_{i,0}$ and $CR_{i,0}$ are the initial values for the i^{th} individual; F_l and F_u are the minimum and maximum lower and upper bounds of the mutation weighting factor; CR_l and CR_u are the minimum and maximum lower and upper bounds of the crossover rate; $Rand_1$, $Rand_2$ and $Rand_3$ represent three independently uniformly distributed random variables in the range [0, 1]; N and D are population size and dimension of the vector (number of decision variables) respectively. The population size N is not changed during the SADE evolution process.

In the proposed SADE algorithm, the F and CR values are appended to the actual solution strings as shown in Figure 2. G is the generation number and G=0 is the initial generation. These F and CR values will evolve along with their corresponding actual solutions.

Mutation

Before the mutation operator is applied, each vector $X_{i,G}$ in the current population is treated as the target vector. Corresponding to each target vector, a mutant vector $V_{i,G}$ = $\{ v_{i,G}^{1}, v_{i,G}^{2}, \ldots\ldots\ldots v_{i,G}^{D} \}$ is generated by adding the weighted difference between two random vectors to a third vector (the base vector) from the current population (D is the number of decision variables). The $F_{i,G}$ value of each target vector $X_{i,G}$ is used to generate the mutant vector, which is given by:

$$V_{i,G} = X_{a,G} + F_{i,G}(X_{b,G} - X_{c,G}) \tag{2}$$

where $X_{a,G}$, $X_{b,G}$, $X_{c,G}$ are three vectors randomly selected from the current population ($a \neq b \neq c$). These three indices are randomly generated for each mutant vector $V_{i,G}$. A total of N mutant vectors, one for each target vector in the population, are produced using Equation (2).

Crossover

A trial vector $U_{i,G}$ ={ $u^1_{i,G}$, $u^2_{i,G}$,……… $u^D_{i,G}$ } is produced by selecting solution component values from either mutant vector ($V_{i,G}$) or its corresponding target vector ($X_{i,G}$) using a crossover process that is similar to uniform crossover. Thus, each component within the trial vector $U_{i,G}$ becomes:

$$u^j_{i,G} = \begin{cases} v^j_{i,G}, \text{ if } Rand_2 \leq CR_{i,G} \\ x^j_{i,G}, \text{ otherwise} \end{cases} \tag{3}$$

where $u^j_{i,G}$, $v^j_{i,G}$, $x^j_{i,G}$ are the j^{th} parameters in the i^{th} trial vector, mutant vector and target vector respectively. If $Rand_2$ is smaller than $CR_{i,G}$ ($0 \leq CR_{i,G} \leq 1$), the value $v^j_{i,G}$ in the mutant vector is copied to the trial vector. Otherwise, the value $x^j_{i,G}$ in the target vector is copied to the trial vector. A total of N mutant vectors $V_{i,G}$ and their corresponding target vectors $X_{i,G}$ are crossed over to generate N trial vectors using Equation (3).

Selection

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

After crossover, the objective function $f(U_{i,G})$ for each trial vector is evaluated. Then each trial vector $U_{i,G}$ is compared with the corresponding target vector $X_{i,G}$ in terms of objective function values. The vector with a smaller objective function value (given that a minimization problem is being considered) survives into the next generation ($X_{i,G+1}$). That is

$$X_{i,G+1} = \begin{cases} U_{i,G}, \text{ if } f(U_{i,G}) \le f(X_{i,G}) \\ X_{i,G}, \text{ otherwise} \end{cases} \quad (4)$$

Thus, N solutions are selected utilizing Equation (4) to form the next generation.

The F and CR values in this proposed SADE algorithm are subject to the selection operator. If a combination of $F_{i,G}$ and $CR_{i,G}$ is able to generate a better solution $U_{i,G}$ compared to $X_{i,G}$, these two values are given to $X_{i,G+1}$ and survive to the next generation; in contrast, if $F_{i,G}$ and $CR_{i,G}$ generate a worse solution $U_{i,G}$ than $X_{i,G}$, then new randomly generated F and CR values are given to $X_{i,G+1}$. The F and CR selections for the next generation are given by:

$$\begin{aligned} F_{i,G+1} &= \begin{cases} F_{i,G}, \text{ if } f(U_{i,G}) \le f(X_{i,G}) \\ F_l + Rand_3(F_u - F_l), \text{if } f(U_{i,G}) > f(X_{i,G}) \end{cases} \\ CR_{i,G+1} &= \begin{cases} CR_{i,G}, \text{ if } f(U_{i,G}) \le f(X_{i,G}) \\ CR_l + Rand_4(CR_u - CR_l), \text{if } f(U_{i,G}) > f(X_{i,G}) \end{cases} \end{aligned} \quad (5)$$

where $Rand_3$ and $Rand_4$ are independently generated random numbers in the range of [0, 1].

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

As can be seen from Equation (1) to (5), the *F* and *CR* values are applied at the individual level and adjusted by means of evolution in the proposed SADE algorithm. It should be noted that neither the population size (*N*) nor mutation strategy have been included in the self-adaptation of the proposed SADE algorithm. For the population size (*N*), a sensitivity study has been undertaken to investigate its impact on the proposed SADE's performance in terms of WDS optimization. For the mutation strategy, it has been demonstrated that the mutation strategy given in Equation (2) is most effective among a number of various mutation strategies introduced by Storn and Price (1995) (Zheng et al. 2011). Thus, the mutation strategy given in Equation (2) is used for the proposed SADE algorithm.

Convergence criterion

In the proposed SADE algorithm, the coefficient of variation ($C_{v,G}$) of the objective function values for the current DE population of solutions is used as the convergence criterion. The coefficient of variation is a concept commonly used in hydrology (Haan 1977) and is defined as

$$C_{v,G} = \frac{s_G}{\left|\overline{OBJ_G}\right|} = \sum_{i=1}^{N}\left(\frac{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(OBJ_{i,G} - \overline{OBJ_G})^2}}{\left|\overline{OBJ_G}\right|}\right) \tag{6}$$

where $C_{v,G}$ is the coefficient of variation of the objective function value based on all individuals at generation *G*; s_G is the standard deviation for the *N* (population size) objective function values at population *G*. $\overline{OBJ_G}$ is the average objective function value

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

at generation of G; The $C_{v,G}$ value reflects the convergence property of the SADE algorithm that has been run as when s_G approaches zero then all individuals of the population are similar in objective function values. The coefficient of variation is used to effectively non-dimensionalize the standard deviation with respect to the mean so that values are comparable across different case studies. This is an important advantage of the proposed new convergence criterion.

If $C_{v,G}$<Tol (where Tol is an appropriately small value, say 10^{-6}), it indicates that all the individuals in the current population at generation G have already located final solutions (usually they will all be identical) and no further improvement can be made. If $C_{v,G}$>Tol, it is likely that not all individuals have converged on the same final solution and that better solutions may be able to be found as the SADE algorithm continues to explore the search space.

This proposed convergence criterion is new and motivated by the fact that all individuals in the DE tend to converge at the same final solution (Price et al. 2005). This convergence criterion significantly differs to the method of using the objective function values between two consecutive generations to terminate the EA evolution (Deb 2001). In the proposed convergence criterion approach, the search of SADE is terminated when all the individuals in the DE locate the same or extremely close final solutions, rather than using the differences of objective function values between two consecutive generations.

Self-adaptive differential evolution applied to the WDS optimization

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

The basic SADE algorithm is a continuous global optimization search algorithm. Therefore, the algorithm must be modified to solve the discrete WDS optimization problem. In this study, the decision variables included in the proposed SADE are the integers that represent the set of discrete pipe diameters. However, real continuous values are created in the mutation process in the proposed SADE algorithm. In the proposed method, these real values are truncated to the nearest integer number and hence mapped to the corresponding pipe diameters for the hydraulic analysis.

A network solver is used to compute the hydraulic balance in the proposed SADE method. For each individual, the network solver is called to perform the hydraulic simulation based on the pipe diameters decoded from integer string of this individual. As such, the head at each node of the WDS that is being optimized is obtained for each individual of the SADE, which, in turn, is used to assess the feasibility of each individual solution (a minimum allowable head requirement at each node usually needs to be satisfied when designing a WDS).

Constraint tournament selection is used in the proposed SADE to handle the constraints and determine the individuals that survive into the next generation (Deb 2000). The constraint tournament algorithm when comparing two solutions (one is the trial vector solution and the other is the target vector solution in the proposed SADE) is given as follows:

1 The feasible solution is selected when compared with an infeasible solution;

2 The solution with a smaller value of objective function value (if cost is being minimized) is preferred between two feasible solutions;

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

3 The solution with less constraint violation is preferred between two infeasible solutions.

With this method, the comparison between the solutions in a tournament never happens in terms of both objective function and penalty function. In the first case, the solution with no head violation is preferred to the one with a head violation and does not take the value of objective function into account. In the second case, the two solutions are compared based on the objective values and the one with a smaller value is selected as both solutions satisfy the constraints. In the last case, the solution with less head violation is selected and the value of the objective function is not considered. Thus, unlike traditional tournament selection, there is no need to specify a penalty multiplier in this proposed method.

CASE STUDIES

The SADE algorithm was developed in C++ and combined with the EPANET2 network solver (Rossman 2000). Four WDS case studies have been used to investigate the effectiveness of the proposed algorithm. These include the New York Tunnels Problem (NYTP) (Dandy et al. 1996), the Hanoi Problem (HP) (Fujiwara and Khang 1990), the Double New York Tunnels Problem (NYTP2) (Zecchin et al. 2005) and the Balerma network (BN) (Reca and Martínez 2006). The number of decision variables and the search space size for each case study is given in Table 1.

The ranges for the *F* and *CR* are generally between 0 and 1 (Storn and Price 1995). The recommended range for *F* is [0.5, 1.0] and for *CR* is [0.8, 1.0] (Price et al. 2005; Liu and Lampinen 2005) based on testing on numerical optimization problems. In order to

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

demonstrate the effectiveness of the self-adaptive algorithm, relatively larger ranges for the *F* and CR values were used in the proposed SADE algorithm. Both *F* and *CR* values in the range of [0.1, 0.9] were utilized for each case study. For the SADE algorithm applied to the WDS optimization, convergence is taken to have occurred when $C_{v,G} < \text{Tol}$. For the computer runs presented in this research the Tol value was set to be 10^{-6}.

CONVERGENCE CRITERION ANALYSIS

The $C_{v,G}$ values at each generation for three SADE algorithm runs with different starting random number seeds applied to the NYTP case study is illustrated in Figure 3.

When the SADE algorithm is run, as can be seen from Figure 3, the value of $C_{v,G}$ overall reduces as the number of generations increases. This shows that individuals in the SADE algorithm tend to be converging by means of evolution. The current best solution for the NYTP case study was first reported by Maier et al. (2003) with a cost of $38.64 million. This best solution was initially found by SADE-2 run when $C_{v,G}=0.023$ at generation 152 (at 4,557 evaluations). Then all the individuals converged at this current best solution at generation 179 ($C_{v,G} < \text{Tol}$). The SADE-1 run first arrived at the current best solution when $C_{v,G}=0.004$ at generation 216 (at 6,478 evaluations) and finally converged at $C_{v,G} < \text{Tol}$ at generation 244. The SADE-3 run initially reached an optimal solution with a cost of $39.06 million when $C_{v,G}=0.034$ at generation 154 (at 4,618 evaluations) and finally converged at this solution at generation 196. The SADE-3 was unable to reach the current best solution by the time the search was terminated at $C_{v,G} < \text{Tol}$.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

From Figure 3, it can be seen that the SADE algorithm runs with different starting random number seeds consistently converged at $C_{v,G}$ <Tol, although they require a different computational overhead. The search process varies for SADE runs starting with different random number seeds and hence each run may require different computational overheads to reach the same final solution. This is reflected by the fact that SADE-1 required 244 generations for all individuals converge to the solution with a cost of $38.64 million, while SADE-2 required 152 generations for all individuals to finally locate this solution. In this case, if a fixed computational budget is used to terminate the evolutions of EA runs, it is impossible to avoid the computational excess or insufficiency since each EA run with different starting random number seed requires different computational overhead. The proposed convergence criterion is able to overcome this disadvantage as convergence occurs based on the evolution feedback for each SADE run rather than specifying a fixed computational budget in advance. This allows SADE runs starting with different random number seeds to terminate their exploration at different numbers of generations purely based on the convergence criterion being satisfied.

It is also difficult to guarantee that each EA run with various starting random number seeds will find the same final solution. For the three different SADE runs given in Figure 3, SADE-1 and SADE-2 found the current best known solution ($38.64 million) for the NYTP case study, while the best solution found by SADE-3 was $39.06 million. The proposed convergence approach is able to indicate that no further improvement on the solution quality can be expected for the SADE-3 run although it has not arrived the current best known solution. This is because that all the individuals for the SADE-3 have converged at the identical final solution with a cost of $39.06 million for $C_{v,G}$ < Tol. Thus,

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

providing a larger computational budget for the SADE-3 run for this particular random number seed would make no difference. Starting another SADE run with other starting random number seeds should be carried out if better solutions are required.

The convergence properties of the SADE algorithm in terms of $C_{v,G}$ applied to the other three case studies produced results similar to those exhibited by the NYTP case study and are therefore not given. From this study, it can be concluded that the proposed termination criterion with $C_{v,G}$ <Tol (see Equation (6)) for WDS optimization successfully avoids computation excess and insufficiency.

POPULATUION SIZE STUDY

Table 2 gives the results of the proposed SADE applied to the four case studies with different population sizes. Multiple SADE runs with different random number seeds were performed for each case study in order to enable a reliable comparison.

The current best known solutions for the NYTP, HP and NYTP2 case studies were first reported by Maier et al. (2003), Reca and Martínez (2006) and Zecchin et al. (2005) with costs of $38.64 million, $6.081 million and $77.28 million respectively. These current best known solutions were also found by the proposed SADE with different population sizes. The best solution found by the proposed SADE for BN case study was €1.983 million.

As shown in Table 2, in terms of percent with the best solution found and the average cost solution based on R runs with different starting random number seeds, the SADE algorithm with a larger population size performed better for each case study. However,

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

the evaluations required to find optimal solutions and to converge using the proposed criterion ($C_{v,G}$<Tol: see Equation (6)) for the SADE with a larger population size are increased significantly as can be seen from Table 2. In considering both the solution quality and efficiency, population sizes of 50, 200, 100 and 500 were selected for the NYTP, HP, NYTP2 and BN case studies respectively. Note that for these population sizes selected: (i) the SADE algorithms exhibited good performance in solution quality and required a reasonably small computational overhead; and (ii) a further increase in population size for each case study only slightly improved the solution quality at the expense of a significantly increased computational overhead.

By comparing the number of decision variables (given in Table 1) and the selected population sizes for each case study (50 for the NYTP, 200 for the HP, 100 for the NYTP2 and 500 for the BN), an approximate heuristic guideline for the population size of the SADE algorithm applied to a WDS case study is within [1*D* 6*D*], where *D* is the number of decision variables for the WDS. This differs with the rule of thumb for the GAs in that the population size should be within [5*D* 10*D*].

The results of the SADE algorithm with population sizes of 50, 200, 100 and 500 for the NYTP, HP, NYTP2 and BN respectively are now used to compare results with other optimization techniques that have been previously applied to these four case studies.

SADE ALGORITHM PERFORMANCE COMPARISON AND DISCUSSION

Case study 1: New York Tunnels Problem (NYTP: 21 decision variables)

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Table 3 gives the results of the proposed SADE and other previously published results for the NYTP case study. The results including the best solution found, the percentage of different runs with the best known solution found, the average cost solution and the average number of evaluations. The results in Table 3 are ranked based on the percent of trials with best solution found (the column 4).

As can be seen from Table 3, the proposed SADE algorithm was able to locate the current best solution with a frequency of 92%, which is the same or higher than other EAs reported in Table 3. It should be highlighted that the proposed SADE algorithm is significantly more efficient than the majority of other EAs to find the optimal solutions in terms of average number of evaluations. As clearly shown in Table 3, the average number of evaluations required to find the first occurrence of optimal solutions based on 50 different SADE algorithm runs was 6,598, which is less than those required by the majority of other EAs given in Table 3. More importantly, the average number of evaluations required for final convergence of the SADE algorithm (when $C_{v,G}$<Tol) was 9,227, which is significantly less than the maximum number of allowable evaluations used for other EAs given in the last column of Table 3.

Case study 2: Hanoi Problem (HP: 34 decision variables)

Table 4 gives a performance summary of the proposed SADE algorithm and other optimization techniques applied to the HP case study. As can be seen in Table 4, the proposed SADE algorithm found the current best solution for the HP case study with a success rate of 84%, which is an improvement compared to other EAs given in Table 4. The SADE algorithm also produced the lowest average cost solution over the 50 different

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

runs as shown in Table 4 with a cost of \$6.090 million, which deviates only 0.15% from the known best solution.

In terms of efficiency, the proposed SADE algorithm with an average number of evaluations of 60,532 did not perform as well as the DE (Suribabu 2010), Scatter Search algorithm (Lin et al. 2007) and GHEST (Bolognesi 2010). However, in terms of comparing the total computational overhead for each run, the average number of evaluations required for convergence (when $C_{v,G}$<Tol) of the proposed SADE algorithm was 74,876, which is less than the maximum number of evaluations used of the other EAs.

It should be highlighted that the results of other EAs in Table 4 were based on fine-tuning parameter values and only the final results with the calibrated parameter values are reported. In reality, adjusting the parameter values for these EAs by a trial-and-error method requires additional computational overhead. In contrast, for the proposed SADE, ranges of the *F* [0.1, 0.9] and *CR* [0.1, 0.9] were used for the HP case study and no tuning was conducted for these parameters.

Case study 3: Double New York Tunnels Problem (NYTP2: 42 decision variables)

In order to enable a comparison with the proposed SADE, the traditional DE algorithm was also applied to the NYTP2 case study. The population size of 100 was also used in the traditional DE algorithm. Values of *F*=0.5 and *CR*=0.6 were found to be appropriate for the NYTP2 case study based on trials of different parameter values. The newly proposed convergence criteria was also used for the traditional DE. The results of the proposed SADE algorithm, the traditional DE algorithm and other optimization techniques that have been previously applied to the NYTP2 are given in Table 5.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

As shown in Table 5, the proposed SADE algorithm outperformed the traditional DE algorithm, the HD-DDS (Tolson et al. 2009) and MMAS (Zecchin et al. 2007) in terms of the percentage of trials with the best solution found. This is reflected from Table 5 that the proposed SADE found the current best solution for the NYTP2 case study with a frequency of 90%, which is higher than all the other EAs given in Table 5.

For the NYTP2 case study, the proposed SADE exhibited a notably better performance in terms of efficiency than other EAs presented in Table 5, as it required a significantly lesser average number of evaluations (33,810) to find the first occurrence of optimal solutions. The average evaluations required for convergence of 50 different SADE runs applied to the NYTP case study was 40,812. This shows the computational overhead for each proposed SADE run was significantly reduced compared with other EAs that terminated the run using a maximum number of allowable evaluations. A convergence comparison between the proposed SADE algorithm run and a traditional DE algorithm run with the same starting number seeds is illustrated in Figure 4.

As can be seen from Figure 4, at evaluation numbers smaller than 30,000, the traditional DE algorithm found the best solution slightly faster than the proposed SADE algorithm when starting with the same random number seeds. In terms of comparing the average cost solution obtained at each generation, the traditional DE algorithm performed better than the proposed SADE algorithm at evaluation numbers smaller than 30,000 as it generated a lower average cost solution than the SADE algorithm. This is due to the fact that the *F* and *CR* values for the traditional DE algorithm have been fine-tuned, while the *F* and *CR* values in the SADE algorithm are initially randomly generated and in the early stages of generation have not yet self-adapted.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

As clearly shown in Figure 4, the SADE algorithm was able to converge faster than the traditional DE algorithm in later generations (that is after 35,000 evaluations) in terms of finding the best solution as well as the best average cost solution. This is because the *F* and *CR* parameter values have maturely evolved. Thus, the proposed SADE algorithm exhibits an improved performance for later generations. The proposed SADE algorithm found the current best solution at evaluation number 46,131 and converged at 54,100 evaluations based on the convergence criterion in Equation 6 ($C_{v,G}$ <Tol), while the traditional DE algorithm found the current best solution for the NYTP2 case study with 81,525 evaluations and finally converged at 94,382 evaluations.

Case study 4: Balerma Network (BN: 454 decision variables)

In comparison, a traditional DE algorithm with a population size of 500, *F*=0.3 and *CR*=0.5 (these two values were selected after a number of fine-tuning trials) was performed for the BN case study. The newly proposed convergence criteria was used for the traditional DE applied to the BN case study. Table 6 outlines the performance comparison of the SADE algorithm with different *CR* ranges, the traditional DE algorithm with tuned parameter values and other optimization techniques that have been previously applied to the BN case study.

As shown in Table 6, the best solution found by the proposed SADE algorithm for the BN case study was €1.983 million, which is higher than the best known solution (€1.940 million) reported by Tolson et al. (2009) using HD-DDS method, but lower than solutions reported by other EAs given in Table 6. However, the HD-DDS (Tolson et al. 2009)

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

yielded the best solution of €1.940 million requiring 30 million evaluations, while the SADE algorithm used only 1.3 million average evaluations to finally converge.

The average number of evaluations required for the SADE algorithm to first reach the optimal solutions was 1.2 million, which is less than those required by most of the EAs given in Table 6. While GHEST (Bolognesi et al. 2009) converged more quickly, the quality of the final solution was worse than that produced by the proposed SADE.

Table 7 gives an analysis of the computational effort required to find the best solutions and the computational effort used to terminate the SADE run (when $C_{v,G} < \text{Tol}$) based on the proposed convergence criterion (see Equation (6)). It was found that the average number of evaluations required to find the first occurrence of the best solution was around 80% of that required for final convergence ($C_{v,G} < \text{Tol}$) of the SADE runs.

CONCLUSION

The performance of all EAs is sensitive to the parameters used. Determining effective parameter values for each WDS optimization problem, therefore, requires a number of trials with different parameter values. This calibration phase results in a significant increase in computational overhead and hence reduces the attractiveness of EAs being used in engineering practice.

The proposed self-adaptive DE algorithm (SADE) method overcomes the challenge mentioned above. A total of five contributions are presented in this paper in terms of novelty and the computational advantage of the proposed SADE algorithm, which are given as follows:

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

(1) The proposed SADE encodes the parameters (*F* and *CR*) into the strings to be automatically adjusted by means of evolution. Consequently, it reduces the effort required for the trial-and–error process normally used to determine the effective parameters for use in the DE algorithm.

(2) The *F* and *CR* values of the proposed SADE algorithm are applied at the individual level rather than the generation level, which differs with the traditional DE algorithm applied to WDS optimization design.

(3) A new convergence criterion has been proposed in the SADE algorithm to avoid pre-specifying convergence conditions. This convergence criterion is based on the coefficient of variation such that $C_{v,G}$<Tol. It has been successfully implemented as the termination condition for the SADE algorithm applied to the WDS optimization. This represents a significant advantage compared to other EAs, where the maximum number of allowable evaluations is required to be pre-specified.

(4) The only parameter value that needs to be provided for the proposed SADE is the population size. The population size is a relatively easy parameter to adjust since a slight variation of its value does not significantly impact the performance of the SADE. In addition, it has been derived in this study that a population size within [1*D*, 6*D*] is an approximate heuristic for the proposed SADE applied to WDS case studies, which differs to the rule of thumb for the GAs in that the population size should be within [5*D*, 10*D*] (Deb 2001), where *D* is the number of decision variables for the WDS that is being optimized.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

(5) A total of four WDS case studies with the number of decision variable ranging from 21 to 454 have been used to verify the effectiveness of the proposed SADE algorithm. For the NYTP, HP and NYTP2 case studies, the SADE performed the best in terms of the percent of the best solution found and exhibited improved performance in convergence speed compared to the majority of other reported EAs. For the large BN case study, the proposed SADE also exhibited a comparable performance to other EAs. It should be highlighted that the results of ***other*** EAs (excluding the new SADE algorithm as proposed in this paper and the HD-DDS) in Table 3 to 6 were based on fine-tuning parameter values and only the final results with the calibrated parameter values are reported. In reality, adjusting the parameter values for these EAs by trial-and-error requires additional computational overhead. In contrast, for the proposed SADE, ranges of the F [0.1, 0.9] and CR [0.1, 0.9] were used for each case study and no tuning was needed to be conducted for these two parameters. Given this fact, it is fair to draw a conclusion that the proposed SADE was able to yield optimal solutions with greater efficiency than other EAs.

The proposed SADE provides a robust optimization tool for the optimization of the design of WDSs (or rehabilitation of an existing WDS). This is because (i) the proposed SADE algorithm does not require as much fine-tuning of parameter values nor pre-specification of a computational budget; and (2) the proposed SADE algorithm is able to find optimal solutions with good quality and great efficiency. In addition, the proposed SADE algorithm can also be used to tackle other water network management problems such as leakage hotpot detection (Wu and Sage 2006), optimal valve operation (Kang and Lansey 2010) and contaminant detection (Weickgenannt et al. 2010). The potential benefit of the proposed SADE algorithm compared to other EAs that have been used to deal with these water network management optimization problems is that it would need significantly less effort to adjust the parameter values. This is a huge advantage especially dealing with the real-time optimization problems for WDSs (Kang and Lansey 2010), in which decisions have to be made with extremely limited time.

The utility of the proposed SADE algorithm has been demonstrated using the least-cost single objective WDS optimization problems in this paper. A natural extension of this proposed self-adaptation algorithm is to extend it to deal with multi-objective WDS optimization problems, for which in addition to the cost, other objectives such as the reliability or greenhouse gases are considered in order to provide more practical solutions for WDS design. This extension is the focus of future work.

REFERENCES

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

Bäck, T., Hoffmeister, F., and Schwefel, H.-P. (1991). "A survey to evolution strategies." R. K. Belew and L. B. Booker, Eds., *Proc. 4th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, San Mateo, Calif., 2-9.

Bolognesi, A., et al. (2010). "Genetic Heritage Evolution by Stochastic Transmission in the optimal design of water distribution networks." *Advances in Engineering Software*, 41, 792-801.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., and Zumer, V. (2006). "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical

Benchmark Problems." *IEEE Transactions on Evolutionary Computation*, 10(6), 646-657.

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.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithm*, Wiley, London.

Eiger, G., Shamir, U., and Ben-Tal, A. (1994). "Optimal design of water distribution networks." *Water Resour. Res.*, 30(9), 2637-2646.

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.

Eiben, A. E., Hinterding, R., and Michalewicz, Z. (1999). "Parameter control in evolutionary algorithms." *IEEE Transactions on Evolutionary Computation,* 3(2), pp.124-141.

Eiger, G., Shamir, U., and Ben-Tal, A. (1994). "Optimal design of water distribution networks." *Water Resources Research*, 30(9), 2637–2646.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Fujiwara, O., Jenchaimahakoon, B., and Edirisinghe, N. C. P. (1987). “A Modified Linear Programming Gradient Method for Optimal Design of Looped Water Distribution Networks.” *Water Resources Research*, 23(6), 977-982.

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

Geem, Z. W. (2009). “Particle-swarm harmony search for water network design.” *Engineering Optimization*, 41(5), 297-311.

Geem, Z. W., and Sim, K. B. (2010) “Parameter-setting-free harmony search algorithm.” *Applied Mathematics and Computation*, 3881~3889.

Gibbs, M. S., Maier, H. R., and Dandy, G. C. (2010). "Comparison of Genetic Algorithm Parameter Setting Methods for Chlorine Injection Optimization." *Journal of Water Resources Planning and Management*, 136(2), 288-291.

Haan, C. T. (1977). *Statistical Methods in Hydrology*, the Iowa State University Press, Ames, Iowa.

Kang, D., and Lansey, K. (2010). "Real-Time Optimal Valve Operation and Booster Disinfection for Water Quality in Water Distribution Systems." *Journal of Water Resources Planning and Management*, 136(4), 463-473.

Lansey, K. E., and Mays, L. W. (1989). “Optimization Model for Water distribution system design.” *J. Hydraul. Eng*., 115(10), 1401-1417.

Lin, M.-D., Liu, Y.-H., Liu, G.-F., and Chu, C.-W. (2007). "Scatter search heuristic for least-cost design of water distribution networks." *Engineering Optimization*, 39(7), 857-876.

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.

Montalvo, I., et al. (2008). "A diversity-enriched variant of discrete PSO applied to the design of water distribution networks." *Engineering Optimization*, 40(7), 655-668.

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

Perelman, L., and Ostfeld, A. (2007). "An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems." *Engineering Optimization*, 39(4), 413-428.

Price, V. K., Storn, M. R., and Lampinen, A. J. (2005). *Differential Evolution: A Practical Approach to Global Optimization*, Springer, Berlin.

Qin, A. K., Huang, V. L., and Suganthan, P. N. (2009). "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization." *IEEE Trans, Evol. Compt.,* 13(2)**,** 398-417.

Quindry, G. E., Brill, E. D., and Liebman, J. C. (1981). "Optimization of Looped Water Distribution Systems." *J. Environmental Engineering*. Div. Am. Soc. Civ. Eng., 107 (EE4).

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

Rossman, L. A. (2000). "*EPANET 2-User manual*." National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency, Cincinnati, OH.

Savic, D. A., and Waters G.A. (1997). "Genetic Algorithms for least-cost design of water distribution networks." *J. Water Resource Planning and Management*, 123(2), 67-77.

Simpson, A. R., Dandy, G. C., and Murphy, L. J. (1994). "Genetic Algorithms compared to other techniques for pipe optimization." *J. Water Resources Planning and Management*, 120(4), 423-443.

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., and Neelakantan T. R. (2006). "Design of water distribution networks using particle swarm optimization." *Urban Water J.,* 3(2), 111-120.

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.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

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.

Vesterstroem, J., and Thomsen, R. (2004). “A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems.” In *Proc. IEEE Congr. Evolutionary Computation*, Portland, 1980-1987.

Weickgenannt, M., Kapelan, Z., Blokker, M., and Savic, D. A. (2010). "Risk-Based Sensor Placement for Contaminant Detection in Water Distribution Systems." *Journal of Water Resources Planning and Management*, 136(6), 629-636.

Wu, Z. Y., and Simpson, A. R. (2002). “A self–adaptive boundary search genetic algorithm and its application to water distribution systems.” *Journal of Hydraulic Research*, 40 (2), 191–203.

Wu, Z. Y., and Walski, T. (2005). “Self-Adaptive Penalty Approach Compared with Other Constraint-Handling Techniques for Pipeline Optimization.” *J. Water Resources Planning and Management*, 131(3), 181-191.

Wu, Z. Y., and Sage, P. (2006). "Water loss deterction via genetic algorithm optimization-based model calibration." *Proc., ASCE 8th Annual International Symposium on Water Distribution System Analysis*, Cincinnati, Ohio.

Zecchin, A. C.,et al. (2005). “Parametric study for an ant algorithm applied to water distribution system optimization.” *IEEE Tans. Evol. Comput.*, 9(2), 175-191.

Zecchin, A. C., et al. (2007). “Ant colony Optimization applied to water distribution system design: Comparative study for five algorithms.” *J. Water Resources Planning and Management*, 133(1), 87-92.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Zheng, F., Simpson, A. R., and Zecchin, A (2011). "Performance Study of Differential Evolution with Various Mutation Strategies Applied to Water Distribution System Optimization." *Proc., World Environmental and Water Resources Congress 2011*, ASCE, 166-176.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Figure Captions list

Figure 1 Flowchart of the proposed SADE algorithm

Figure 2 Encoding for the proposed SADE algorithm

Figure 3 The $C_{v,G}$ values in each generation for three different SADE algorithm runs applied to the NYTP case study. Points A, B, and C reflect the points at which the best solution was found within each run.

Figure 4 Convergence properties of the SADE and the traditional DE for the NYTP2 case study with the same random number seed of 100.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

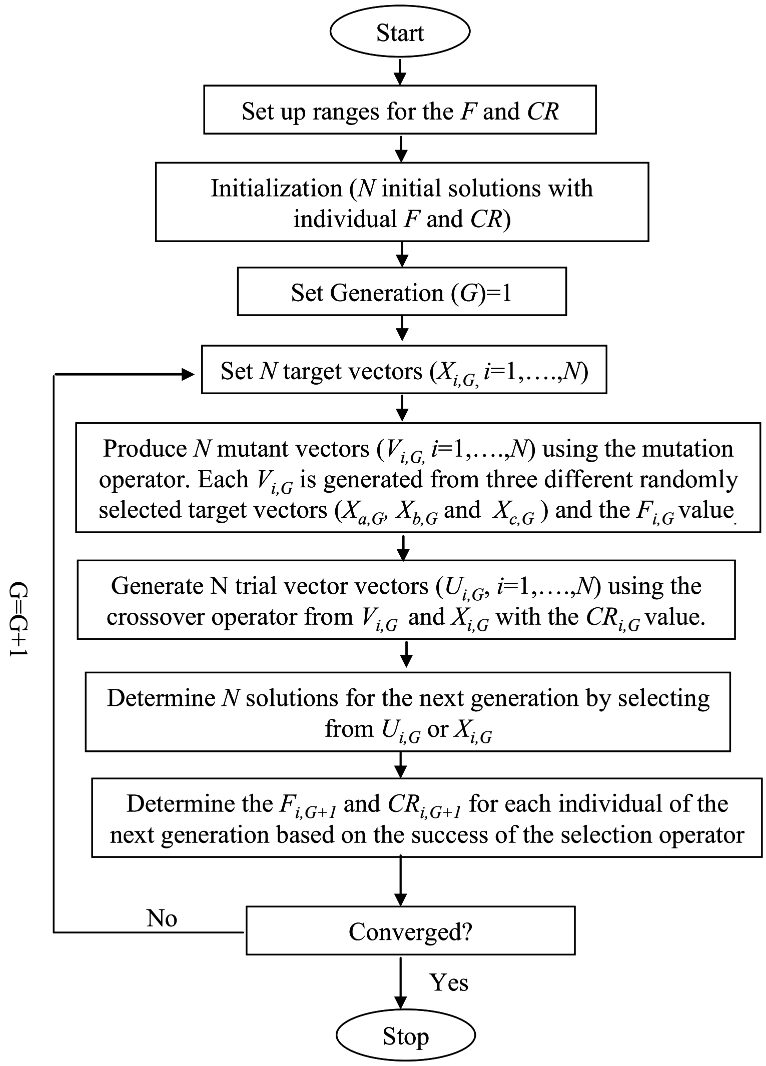

Figure 1 Flowchart of the proposed SADE algorithm

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

$X_{1,G}$	$F_{1,G}$	$CR_{1,G}$
$X_{2,G}$	$F_{2,G}$	$CR_{3,G}$
…	…	…
$X_{N,G}$	$F_{N,G}$	$CR_{N,G}$

Figure 2 Encoding for the proposed SADE algorithm

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

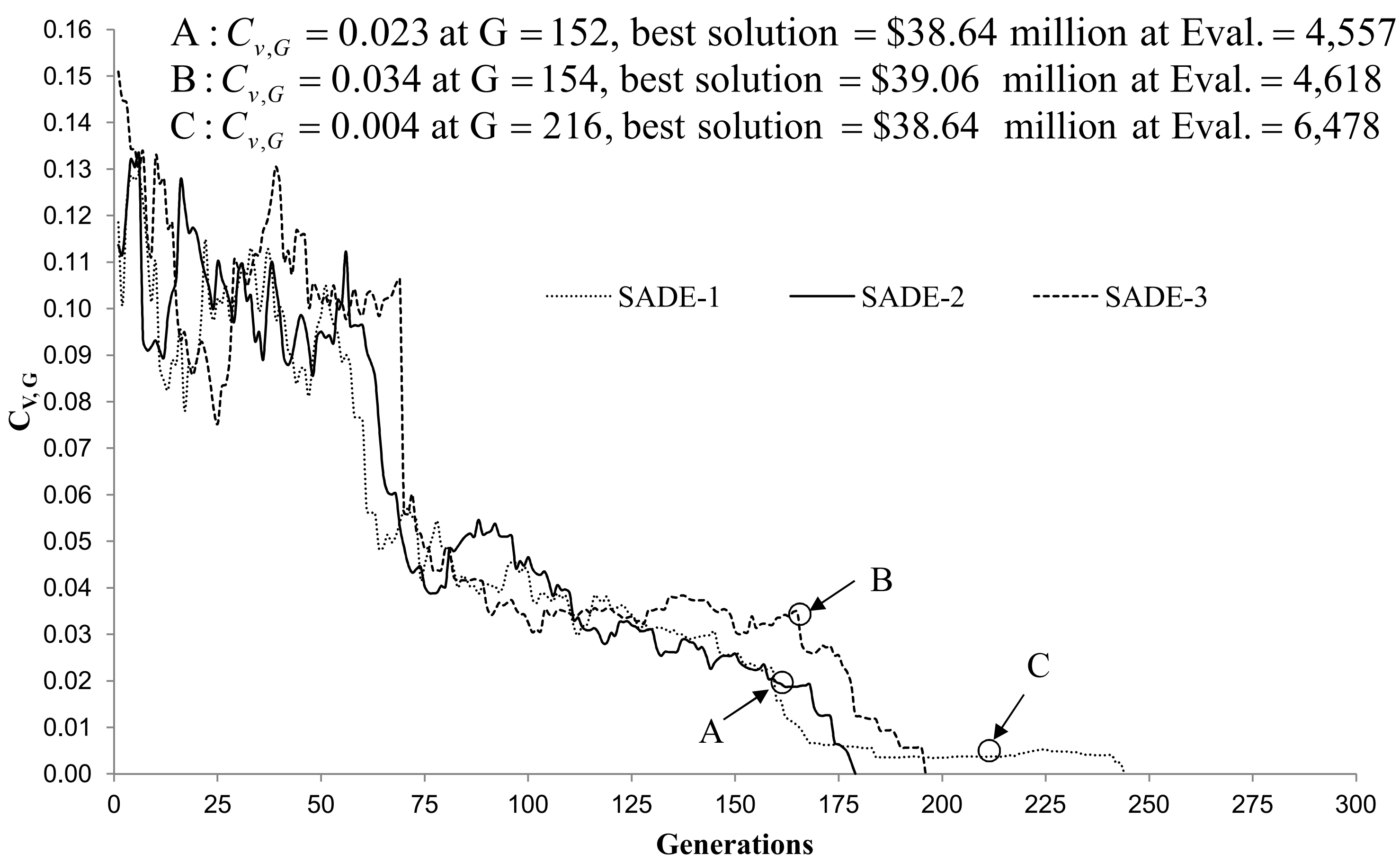

Figure 3. The $\mathbf{C_{V,G}}$ values in each generation for three different SADE algorithm runs applied to the NYTP case study. Points A, B, and C reflect the points at which the best solution was found within each run.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

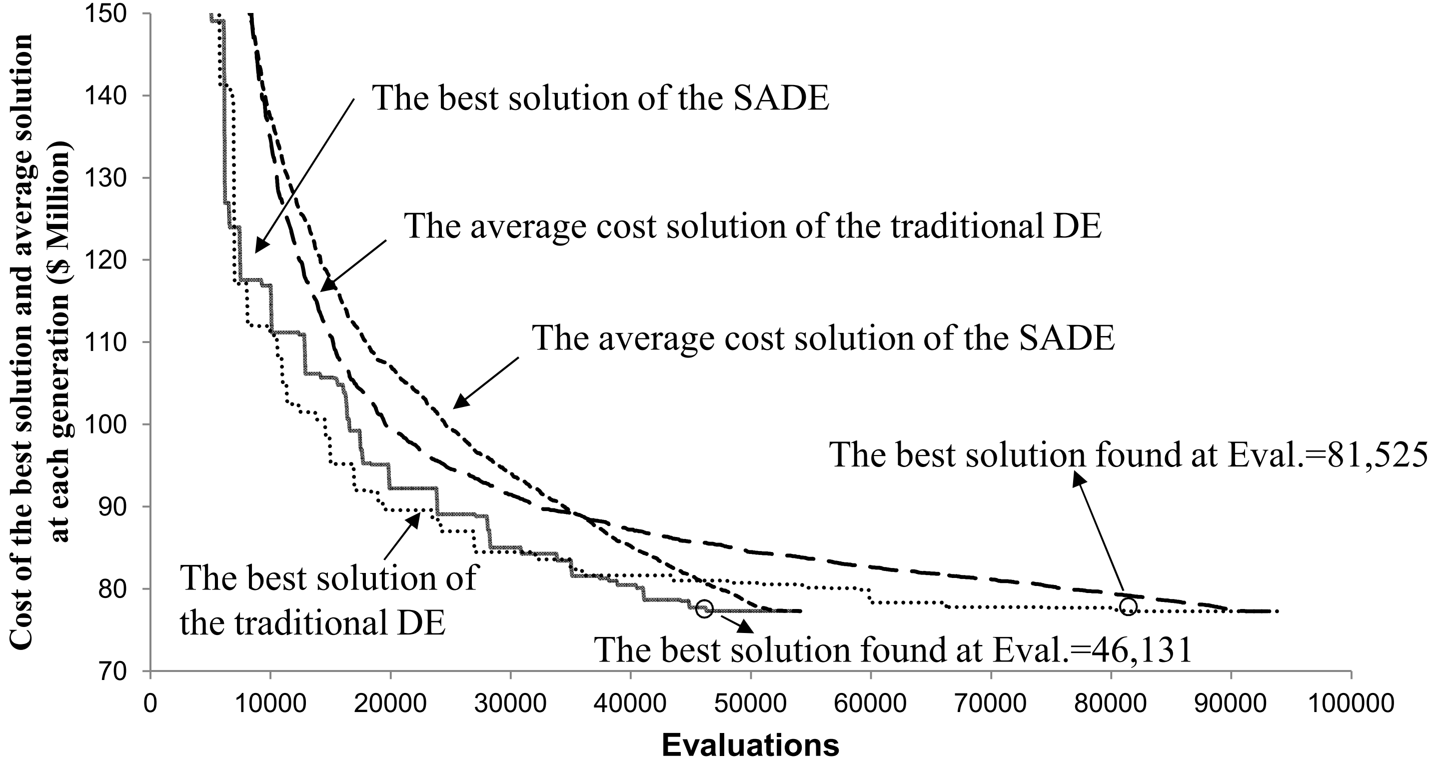

Figure 4 Convergence properties of the SADE and the traditional DE for the NYTP2 case study with the same random number seed of 100.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Table 1 Summary of case study characteristics

WDS case study	Number of decision variables	Number of total available tunnel or pipe diameters that can be used	Search space size
NYTP	21	16	1.934×10^{25}
HP	34	6	2.865×10^{26}
NYTP2	42	16	3.741×10^{50}
BN	454	10	10^{454}

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Table 2 Results of the SADE with different population sizes

Case study	Population size (N)	Best solution found[a]	Percent with the best solution found (%)	Average cost solution[a]	Average number of evaluations to find the final solutions	Average number of evaluations to converge ($C_{v,G}$<Tol)
NYTP (R=50)	30	38.64	64	38.94	4,069	5,375
	50	38.64	92	38.64	6,584	9,227
	100	38.64	98	38.64	12,874	19,270
HP (R=50)	100	6.081	56	6.145	38,210	45,848
	200	6.081	84	6.090	60,532	74,876
	300	6.081	84	6.090	125,454	170,724
NYTP2 (R=50)	100	77.28	90	77.28	33,810	40,812
	200	77.28	98	77.28	70,196	87,592
	300	77.28	100	77.28	109,446	167,472
BN (R=10)	500	1.983	10	1.995	1.2×10^6	1.3×10^6
	1000	1.983	10	1.986	4.1×10^6	4.2×10^6
	2000	1.983	10	1.985	8.5×10^6	8.7×10^6

Note:R=number of runs using different starting random number seeds. [a]The cost unit for the NYTP and HP case studies is $ million and the cost unit for the BN case study is € million.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Table 3 Summary of SADE and other EAs applied to the NYTP case study

(1) Algorithm[9]	(2) No. of runs	(3) Best solution ($M)	(4) Percent of trials with best solution found	(5) Average cost ($M)	(6) Average evaluations to find first occurrence of the best solution	(7) Maximum allowable evaluations or evaluations for convergence
SADE[1]	50	38.64	92%	38.64	6,598	9,227[a]
GHEST[2]	60	38.64	92%	38.64	11,464	-
HD-DDS[3]	50	38.64	86%	38.64	47,000	50,000
Suribabu DE[4]	300	38.64	71%	NA	5,492	10,000
Scatter Search[5]	100	38.64	65%	NA	57,583	-
MMAS[6]	20	38.64	60%	38.84	30,700	50,000
PSO variant[7]	2000	38.64	30%	NA	NA	80,000

[1]Results from this study. [2]Bolognesi et al. (2010). [3]Tolson et al. (2009). [4]Suribabu (2010). [5]Lin et al. (2007). [6]Zecchin et al. (2007). [7]Montalvo et al. (2008). [8]Average evaluations to final convergence. [9]Results are ranked based on column (4).

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Table 4 Summary of SADE and other EAs applied to the HP case study

(1) Algorithm[10]	(2) No. of runs	(3) Best solution ($M)	(4) Percent of trials with best solution found	(5) Average cost ($M)	(6) Average evaluations to find first occurrence of the best solution	(7) Maximum allowable evaluations or evaluations for convergence
SADE[1]	50	6.081	84%	6.090	60,532	74,876[9]
Suribabu DE[2]	300	6.081	80%	NA	48,724	100,000
Scatter Search[3]	100	6.081	64%	NA	43,149	-
GHEST[4]	60	6.081	38%	6.175	50,134	-
GENOME[5]	10	6.081	10%	6.248	NA	150,000
HD-DDS[6]	50	6.081	8%	6.252	100,000	100,000
PSO variant[7]	2000	6.081	5%	6.310	NA	500,000
MMAS[8]	20	6.134	0%	6.386	85,600	100,000

[1]Results from this study. [2]Suribabu (2010). [3]Lin et al. (2007). [4]Bolognesi et al. (2010). [5]Reca and Martínez (2006). [6]Tolson et al. (2009). [7]Montalvo et al. (2008). [8]Zecchin et al. (2007). [9]Average evaluations to final convergence. [10]Results are ranked based on column (4).

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Table 5 Summary of SADE and other EAs applied to the NYTP2 case study

(1) Algorithm[5]	(2) No. of runs	(3) Best solution ($M)	(4) Percent of trials with best solution found	(5) Average cost ($M)	(6) Average evaluations to find first occurrence of the best solution	(7) Maximum allowable evaluations or evaluations for convergence
SADE[1]	50	77.28	90%	77.28	33,810	40,812[4]
Traditional DE[1] (F=0.5, CR=0.6)	50	77.28	86%	77.28	70,104	87,457[4]
HD-DDS[2]	20	77.28	85%	77.28	310,000	300,000
MMAS[3]	20	77.28	5%	78.20	238,300	300,000

[1]Results from this study. [2]Tolson et al. 2009. [3]Zecchin et al. 2007. [4]Average evaluations to final convergence. [5]Results are ranked based on column (4)

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012; posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Table 6 Summary of SADE and other EAs applied to the BN case study

(1) Algorithm[6]	(2) No. of runs	(3) Best solution (€M)	(4) Percent with the best solution found (%)	(4) Average cost (€M)	(5) Average evaluations to find first occurrence of the best solution	(6) Maximum allowable evaluations or evaluations for convergence
HD-DDS[2]	1	1.940	-	NA	NA	30×10^6
SADE[1]	10	1.983	10	1.995	1.2×10^6	1.3×10^6
Traditional DE(F=0.3,CR=0.5)[1]	10	1.998	10	2.031	2.3×10^6	2.4×10^6 [a]
GHEST[3]	10	2.002	10	2.055	2.5×10^5	NA
HS[4]	NA	2.018	NA	NA	10^7	10×10^6
GENOME[5]	10	2.302	10	2.334	NA	10×10^6

[1]Results from this study. [2]Tolson et al. 2009. [3]Bolognesi et al. (2010). [4]Geem (2009). [5]Reca and Martínez (2006). [6]Results are ranked based on column (3). NA means not available. [a]Average evaluations to final convergence.

Journal of Computing in Civil Engineering. Submitted August 17, 2011; accepted March 15, 2012;
posted ahead of print March 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000208

Table 7 Summary of computational effort of SADE for each case study

WDS case study	Number of different runs	Average number of evaluations required to find the best solution (AE1)	Average number of evaluations required to terminate the SADE runs based on the proposed convergence criterion (AE2)	Percent (AE1/AE2)
NYTP	50	6,584	9,227	71.4%
HP	50	60,532	74,876	80.8%
NYTP2	50	33,810	40,812	82.8%
BN	10	1.2×10^6	1.3×10^6	92.3%