Developed Swarm Optimizer: A New Method for Sizing Optimization of Water Distribution Systems

R. Sheikholeslami[1] and S. Talatahari[2]

Abstract: The introduction of metaheuristic algorithms in water resources engineering has greatly raised the need for continued development of appropriate optimization methodologies for analysis, planning, design, and operation of water resources systems. This paper proposes a novel developed swarm-based optimization algorithm named DSO, which integrates the accelerated particle swarm optimization (PSO) with the big bang-big crunch algorithm (BB-BC) to optimize the design of water distribution systems (WDSs). Traditional PSO is easy to fall into stagnation when no particle explores a position that is better than its previous best position for several iterations. To deal with the problem of maintaining diversity within the swarm and to enhance the exploration in the search, the concepts of the Big Crunch and Big Bang strategies from the BB-BC algorithm are incorporated into the global and local searching steps of the accelerated PSO, respectively. In addition, a harmony search–based strategy is used to control the location of generated particles, and finally a modified version of the feasible-based mechanism is applied to handle the constraints. The DSO approach obtains competitive results on three well-known benchmark WDS optimization problems, with a number of decision variables ranging from 30 to 454, at a relatively low computational cost. **DOI: 10.1061/(ASCE)CP.1943-5487.0000552.** © *2016 American Society of Civil Engineers.*

Author keywords: Water distribution system; Optimum pipe sizing; Swarm intelligence; Metaheuristic algorithms.

Introduction

Optimization has long been recognized as an inseparable component of water resources planning and management, and has been used extensively to deal with a large number of problems including hydrologic model calibration, data collection activities, water supply system design and operation, water quality and wastewater management, and groundwater management (Maier et al. 2014). Especially in the case of common water distribution systems (WDSs), for which a reasonable final design may yield a significant cost reduction, locating the optimum solution through efficient approaches becomes crucial.

In contrast, the optimal design of a WDS, while at the same time satisfying various requirements on hydraulic response, cost, maintenance, and construction, is a complicated task. This problem is usually formulated in such a way that the objective is to minimize the total cost of the WDS in which the design variables are pipe sizes. Generally, the optimization process involves simultaneous consideration of continuity equation, energy conservation, and the relation between fluid friction and energy dissipation, which makes the analytical solution of the problem complicated because the resulting problem is a mixed integer nonlinear programming problem (MINLP). The existence of integer variables and the nature of the continuity equations make this formulation nonconvex. In addition, the problem constraints are not smooth; therefore, the application of smooth algorithms may not be appropriate (Vairavamoorthy and Ali 2005). It is a well-established fact that the WDS optimization problem belongs to the class of nondeterministic polynomial-time hard (NP-hard) problems (Yates et al. 1984). Essentially, the NP-hard indicates that for an N-pipe network, the computational time required for a rigorous algorithm is at best an exponential function of N, and is thus enormous even for relatively small WDS. Specifically, when large-scale problems are considered, metaheuristics are one of the best alternatives on which experts can often rely, as exact algorithms take exponential time to find an optimal solution to NP-hard problems.

The metaheuristic optimization techniques have been used to obtain global or near-global optimum solutions because of their capability of exploring and finding promising regions in the search space in an affordable time. Many of these heuristic search techniques are created, for example, by the simulation of the following natural processes: Genetic algorithm (GA) (Holland 1975) from biology; harmony search (HS) algorithm (Geem et al. 2001) from music; simulated annealing (SA) (Kirkpatrick et al. 1983), big bang-big crunch (BB-BC) (Erol and Eksin 2006), and charged system search algorithm (CSS) (Kaveh and Talatahari 2010) from physics; and ant colony optimization (ACO) (Dorigo et al. 1996), particle swarm optimization (PSO) (Eberhart and Kennedy 1995), and chaotic swarming of particles (CSP) (Kaveh et al. 2014) from collective behavior of animal species. Among these phenomenon-mimicking metaheuristics, algorithms inspired by the collective behavior of species such as ants, bees, fishes, and birds are referred to as swarm intelligence algorithms. Swarm intelligence methods originate from the social behavior of those species that compete for foods (Yang 2010). These algorithms impose fewer mathematical requirements and do not require very well defined mathematical models that are hard to organize for system modeling and cannot be used for large-scale problems because of high time costs, even though the mathematical model has been organized (Alatas 2011). However, the original version of these nature-inspired heuristic search techniques has some difficulties in solving different problems (Kaveh and Talatahari 2009a). These methods are developed in a way that they should solve different complex optimization problems; therefore, they may show a good performance for a type of problem, but their performance becomes unacceptable for

[1]Global Institute for Water Security, School of Environment and Sustainability, Univ. of Saskatchewan, Saskatoon, SK, Canada S7N 3H5.

[2]Dept. of Civil Engineering, Univ. of Tabriz, 51664 Tabriz, Iran (corresponding author). E-mail: siamak.talat@gmail.com.

Note. This manuscript was submitted on May 12, 2015; approved on September 30, 2015; published online on January 12, 2016. Discussion period open until June 12, 2016; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

another problem. As a result, some modifications and improvements are necessary when meta-heuristic algorithms are applied to solving a special problem. Although some attempts have been made to facilitate a deeper understanding of the search behavior of these methods (e.g., Zecchin et al. 2012; Gibbs et al. 2014), because of the stochastic nature of these algorithms, much more research is needed. Therefore, using a new metaheuristic optimization technique is one of the interesting, if not the best, way of treating the WDS optimization problem.

This paper presents a newly developed swarm-based optimization (DSO) algorithm for optimizing the design of WDSs. Basically, the DSO involves the accelerated particle-swarm optimization and big bang-big crunch algorithm in both the global and local searching steps. In this way, the new formulations are introduced to improve the performance of the traditional PSO algorithm. Another essential feature of the proposed algorithm is the use of the harmony search scheme for controlling the boundary constraints of the design variables. Moreover, the DSO handles the problem-specific constraints using a modified feasibility-based mechanism. To investigate the performance of the proposed DSO algorithm, the optimal cost design of WDSs is implemented, which is a large-scale, mixed-integer, and nonlinear problem. Thus, the objectives of the current study are to (1) review the swarm-based algorithms for WDS optimization and describe the evolution of these algorithms; (2) develop a new swam-based method; and (3) demonstrate the proposed algorithm's performance for solving the WDS's design problem.

Review of Swarm-Based Optimization Algorithms for WDS Optimization

The state-of-the-art reviews of the metaheuristics and their applications in WDS design optimization are outlined in Corte and Sörensen (2013) and Maier et al. (2014). Among these techniques, swarm-based methods have been repeatedly applied to WDS optimization. From the group of swarm-based methods, it is common to find papers that apply ACO and PSO. However, other techniques have also been investigated, such as the charged system search (CSS), differential evolution (DE), and artificial immune algorithm (AI).

The ACO algorithm is a simulation of the ant's swarm intelligent behavior in food seeking, in which ants exchange the information by releasing pheromone and can always find a shortest routing path from their nest to food source by swarm cooperation. For the first time, Maier et al. (2003) used ACO for WDS optimization and found that ACO can be computationally efficient. Zecchin et al. (2005) provided a deeper understanding of the ACO parameters and developed parametric guidelines for the application of the ACO to WDS optimization. They presented a new formulation to WDS optimization, namely the iteration-best ant system. In addition, the max-min ant system (MMAS) was proposed by Zecchin et al. (2006). Ostfeld and Tubaltzev (2008) applied ACO to the minimization of the WDSs design and operation costs. The main contribution of their study was the extension of previous works on the application of the ACO for WDS optimal design, to the general case of WDS with pumping stations, storage, and multiple extended-period loading conditions. To realize the searching behavior of the ACO and its variants, i.e., MMAS, ant system (AS), elitist ant system (AS_{elite}), and elitist-rank ant system (AS_{rank}), Zecchin et al. (2012) introduced different quantitative descriptions to determine the behavioral characteristics of these algorithms. For almost all of the case studies, these metrics revealed the high performance of the MMAS algorithm.

The PSO algorithm, which stems from the simulation of the food-seeking behavior of bird flock, is based on the properties of swarms. This algorithm, after some modifications, has been used successfully to solve a range of water-resource-optimization problems. PSONET, a customized program developed for the optimization of WDS by Suribabu and Neelakantan (2006), found the best solutions for the benchmark networks in a fewer number of iterations, compared with the GA and SA. The satisfactory performance of the PSO was reported by Montalvo et al. (2008) when applied to minimum-cost design of the WDSs. Bansal and Deep (2009) used a PSO-based approach for the optimal design of the two types of WDSs, namely, a serial network and a branched network. In their study, the results obtained by the PSO were comparable and sometimes better than the results obtained by conventional techniques. For determining the optimal design of WDSs, Montalvo et al. (2010) developed a new variant of the PSO with a self-adaptive strategy in which no parameter tuning is necessary. Their study also indicated that this framework, together with other modifications previously introduced by the authors, can be greatly useful when applied to other real-world problems. Sedki and Ouazar (2012) proposed a hybrid PSO and DE method for minimizing the cost of WDS. They suggested that PSO-DE is a promising method for solving WDS design problems.

The DE algorithm is similar to GAs and pattern search. It uses multi-agents or search vectors to carry out search, and can be considered as a further development to GAs. It is a stochastic search algorithm with self-organizing tendency. Vasan and Simonovic (2010) described the development of a DENET computer model that involves the application of the DE, linked to the hydraulic simulation solver *EPANET*, for the optimal design of WDSs. Zheng et al. (2011) combined a nonlinear programming (NLP) and a DE algorithm for optimizing WDS design, in which an NLP was used to approximate the optimal solutions, and a DE algorithm was subsequently employed to enhance the identified approximate solutions. Dong et al. (2012) provided a comprehensive comparison study for solving various sets of WDS optimization problems, and concluded that DE obtained better results than GAs. In a more recent study, Zheng et al. (2014) coupled a binary linear programming with a DE algorithm to improve the optimization efficiency of the WDS design problems. Moreover, the DE's searching behavior in WDS optimization was investigated by Zheng et al. (2015).

As another swarm-based method, Sheikholeslami et al. (2014) applied the CSS algorithm to the optimal design of WDSs. Compared with other metaheuristics such as GA, PSO, HS and ACO, the CSS has less computational cost and can determine the optimum result with a smaller number of function evaluations. The CSS is inspired by the governing laws of electrostatics in physics and the governing laws of motion from the Newtonian mechanics. AI is another effective swarm-based algorithm in simulating biological processes, inspired by the defense mechanism of the biological immune system. Chu et al. (2008) proposed an optimization procedure based on AI framework to optimize the designs of WDSs. They also developed a modified AI that exploits the global search capability of the GA.

Utilized Optimization Algorithms

Particle Swarm Optimization

PSO involves a number of particles that are initialized randomly in the space of the design variables. These particles fly through the search space, and their positions are updated based on the best positions of individual particles and the best position among all particles in the search space, which in minimum-cost design problems corresponds to a particle with the least cost (Eberhart and Kennedy 1995). In the original PSO, a swarm consists of N particles moving around in a D-dimensional search space. The position of the jth particle at the kth iteration is used to evaluate the quality of the particle and represents candidate solution(s) for the search or optimization problems. The update moves a particle by adding a change velocity $\mathbf{V}_j^{k+1}$ to the current position $\mathbf{X}_j^k$ as follows:

$$\mathbf{X}_j^{k+1} = \mathbf{X}_j^k + \mathbf{V}_j^k \tag{1}$$

$$\mathbf{V}_j^{k+1} = \omega \mathbf{V}_j^k + c_1 \times \mathbf{r}_{\mathbf{1}j}^k \otimes (\mathbf{P}_j^k - \mathbf{X}_j^k) + c_2 \times \mathbf{r}_{\mathbf{2}j}^k \otimes (\mathbf{P}_g^k - \mathbf{X}_j^k) \tag{2}$$

where ω = inertia weight to control the influence of the previous velocity; $\mathbf{r}_{1j}^k$ and $\mathbf{r}_{2j}^k$ = random vectors uniformly distributed in the range of (0,1); c_1 and c_2 = two acceleration constants known as the cognitive and social parameters, respectively; $\mathbf{P}_j^k$ = best position of the jth particle up to iteration k; and $\mathbf{P}_g^k$ = best position among all particles in the swarm up to iteration k. The product $\otimes$ denotes entrywise multiplications (Hadamard product).

In the standard PSO, the reason for using the individual best, $\mathbf{P}_j^k$, is primarily to increase the diversity in the quality solutions, but this diversity can be achieved by using some randomness. A simplified version that could accelerate the convergence of the algorithm is to use the global best only. Therefore, the velocity vector is generated by a simpler formula as (Yang 2010)

$$\mathbf{V}_j^{k+1} = \mathbf{V}_j^k + c_1 \times \mathbf{rn}_j^k + c_2 \times (\mathbf{P}_g^k - \mathbf{X}_j^k) \tag{3}$$

where $\mathbf{rn}_j^k$ = random vector whose elements are normally distributed with a zero mean and a unit standard deviation, to replace the second term of Eq. (2). The update of the position is simply like Eq. (1). To increase the convergence even further, Eq. (1) could be rewritten as the update of the location in a single step, as follows:

$$\mathbf{X}_j^{k+1} = (1 - c_2) \times \mathbf{X}_j^k + c_1 \times \mathbf{rn}_j^k + c_2 \times \mathbf{P}_g^k \tag{4}$$

The velocity does not appear in Eq. (4), so there is no need to deal with the initialization of the velocity vectors. Therefore, the accelerated PSO is much simpler to understand and implement. This simpler version will give the same order of convergence (Gandomi et al. 2013). Typically, $c_1 = [0.1L \quad 0.5L]$ and $c_2 = [0.2 \quad 0.7]$ are sufficient for most applications in which L is the scale of each variable (Yang 2010). Generally, these parameters should be related to the scales of the independent variables and the search domain. According to the previously used and suggested values in Wang et al. (2014), Gandomi et al. (2013), and Yang et al. (2011), in this study $c_2 = 0.5$ is used for the accelerated PSO. The pseudocode of the accelerated PSO algorithm can be summarized as follows:

Step 1: *Initialization*. Initialize an array of particles with random positions;

Step 2: *Global best updating*. Determine the current best for each particle, and update the global best position, $\mathbf{P}_g^k$;

Step 3: *Solution construction*. Move each particle to the new position considering the related equation [Eq. (4)]; and

Step 4: *Terminating criterion controlling*. Repeat Steps 2 and 3 until a terminating criterion is satisfied.

Big Bang–Big Crunch Algorithm

The BB-BC method developed by Erol and Eksin (2006) consists of two phases: a big bang phase and a big crunch phase. During the big bang phase, new solution candidates are randomly generated around a *center of mass* that is later calculated in the big crunch phase with respect to their fitness values. After the big bang phase, a contraction operation is applied during the big crunch. In this case, the contraction operator takes the current positions of each candidate solution in the population and its associated fitness function value, and computes a center of mass. In the BB-BC algorithm, the point representing the center of mass at the kth iteration, denoted by $\mathbf{X}_c^k$, is calculated as

$$\mathbf{X}_c^k = \frac{\sum_{j=1}^{N} \frac{1}{f_j^k} \mathbf{X}_j^k}{\sum_{j=1}^{N} \frac{1}{f_j^k}} \tag{5}$$

where $\mathbf{X}_j$ = position of jth solution in a D-dimensional search space; f_j^k = fitness function value of this point at the kth iteration; and N = population size.

After the big crunch phase, the algorithm creates new positions of candidate solutions for the next iteration of the big bang around the center of mass using a normal distribution operation, in which the standard deviation of this normal distribution function decreases as the number of iterations of the algorithm increases. The following equation is used to generate the new candidates around $\mathbf{X}_c^k$:

$$\mathbf{X}_j^{\text{new}} = \mathbf{X}_c^k + \mathbf{rn}_j^k \otimes \frac{\alpha(\mathbf{X}^{\max} - \mathbf{X}^{\min})}{k+1} \tag{6}$$

where $\mathbf{X}_j^{\text{new}}$ = new position of the jth candidate solution; $\mathbf{X}^{\min}$ and $\mathbf{X}^{\max}$ = lower and upper bounds of the design variables, respectively; $\mathbf{rn}_j^k$ = random vector from a standard normal distribution; and α = parameter for limiting the size of the search space. The pseudocode of the BB-BC algorithm can be summarized as follows:

Step 1: *Initialization*. Form the initial population by randomly spreading the solution candidates over all of the search space in a uniform manner (first big bang);

Step 2: *Center of mass updating*. Evaluate the fitness function of each individual point and update the center of mass position (big crunch phase);

Step 3: *Solution construction*. Generate new solution candidates using normal distribution [Eq. (6)] (second big bang); and

Step 4: *Terminating criterion controlling*. Repeat Steps 2 and 3 until a terminating criterion is satisfied.

Developed Swarm-Based Algorithm

The PSO algorithm has its own disadvantages, such as the high speed of convergence, which often implies a rapid loss of diversity during the optimization process and necessarily leads to undesirable premature convergence (Xinchao 2010). In this section, a newly developed swarm-based algorithm is presented to escape from the local optimal trap. The proposed DSO is based on the PSO and the BB-BC optimization algorithms, in which the ideas of the big crunch and big bang strategies are incorporated into the global and local searching steps of the accelerated PSO, respectively. In addition, an HS-based method is used to control the location of generated particles. Furthermore, a modified feasible-based mechanism is used to handle the constraints. The following subsections describe the DSO in more detail.

Global Searching Step

The first work on hybridizing the standard PSO and BB-BC algorithm was suggested by Talatahari et al. (2013a), in which the velocity formulation [Eq. (2)] of the standard PSO algorithm is modified by adding the term of the center of mass from the BB-BC algorithm. Inspired by their study, the accelerated PSO and the BB-BC algorithms are integrated next. In PSO, the swarm converges rapidly within the intermediate vicinity of the $\mathbf{P}_g^k$. However, such a high convergence speed often results in (1) the diversity loss, and (2) premature convergence if the $\mathbf{P}_g^k$ corresponds to a local optima. To cope with this issue, one or more additional terms are often added to the velocity formula (Talatahari et al. 2013a, b).

The main searching step of the proposed DSO is based on the accelerated PSO and the big crunch level of the BB-BC optimization, which is obtained by modifying the position-updating equation, Eq. (4). If the effect of all other particles can be used in terms of position or velocity-updating equations, a far more effective PSO-based algorithm would be obtained. This motivates the development of the DSO by incorporating the center of mass ($\mathbf{X}_c^k$) into the position-updating equation. In the big crunch strategy of the BB-BC method, $\mathbf{X}_c^k$ is a good agent of all particles; therefore, one can use $\mathbf{P}_g^k$, the global best position, and/or $\mathbf{X}_c^k$, the center of mass point. To fulfill this aim, in the proposed algorithm the movement equation of the accelerated PSO is reformulated as follows:

$$\mathbf{X}_j^{k+1} = (1 - c_2) \times \mathbf{X}_j^k + c_1 \times \mathbf{rn}_j^k + c_2 \times [\mathbf{r}_{\mathbf{1}j}^k \otimes \mathbf{P}_g^k + (1 - \mathbf{r}_{\mathbf{1}j}^k) \otimes \mathbf{X}_c^k] \quad (7)$$

Similar to the PSO algorithm, $\mathbf{r}_{\mathbf{1}j}^k$ = random vectors uniformly distributed in the range of [0, 1]. This new modification does not increase the number of required parameters compared with the PSO algorithm. It needs only two parameters to be adjusted.

In the proposed DSO algorithm, Eq. (7) shows how a particle position is updated based on three parts. The first part is referred to as *momentum term*, which represents the influence of the previous position toward the current position. The second part represents the randomization, which makes the algorithm explore the global search space effectively. The third part represents the cooperation among the particles in finding the global optimal solution, known as the social component, which always moves the particles toward the global best position found so far.

Local Searching Step

In this paper, to enhance the performance of the DSO algorithm, a new particle-updating strategy is presented to model the lack of information about the true optimality of the global best position and to attain rapidly the feasible solution space. The new particle-updating strategy is based on the concept of possibility and normal distribution function, $N(\mu, \sigma)$. In the case of knowledge representation or artificial intelligence, the normal distribution model can reflect the basic principles of the evolution of biological adaptive systems or the swarming behavior of animal species. For example, the mean value, μ, represents the good individual genetic characteristics of the parents, and is the offspring's inheritance from the parents; it can also be considered as the best historical position that the entire swarm has passed, which is referred to as *global best*. The standard deviation, σ, indicates the uncertainty of the parameters, showing the mutation characteristics of species in the evolutionary process and represents the degree of uncertainty about the optimality of the global best.

By defining two digital characteristics of the normal distribution model, the proposed local searching step in the DSO algorithm is described next. After performing an iteration using the global searching step using Eq. (7), in the local searching step each particle generates a solution ($\mathbf{Z}_j^k$) around the global best point, which can be calculated using a normal distribution function as

$$\mathbf{Z}_j^k = N\{[\mathbf{r}_{\mathbf{1}j}^k \otimes \mathbf{P}_g^k + (1 - \mathbf{r}_{\mathbf{1}j}^k) \otimes \mathbf{X}_c^k], \sigma\} \quad (8)$$

To account for the information received over time that reduces uncertainty about the global best position, σ in the kth iteration is modeled using a nonincreasing function as follows:

$$\sigma = \mathbf{rn}_j^k \otimes \frac{\alpha(\mathbf{X}^{\max} - \mathbf{X}^{\min})}{k+1} \quad (9)$$

Similar to the big bang phase of the BB-BC, $\mathbf{rn}_j^k$ is the random vector from a standard normal distribution, and α is a parameter for limiting the size of the search space.

In the proposed method, the objective function value, $f(\mathbf{Z}_j^k)$, is computed and the new position of jth particle, $\mathbf{Z}_j^k$, is replaced with the current position of particle j in the swarm, if and only if $f(\mathbf{Z}_j^k) < f(\mathbf{X}_j^k)$ and new particle is in the feasible space. This simple particle-updating strategy should be distinguished from the conventional mutation operator, which applies a random perturbation to the particles. By considering the uncertainty associated with each global best point as a function of time [Eq. (9)], a normal-distribution model provides a simple and efficient exploration at the early stage when σ is large and encourages local fine-tuning (exploitation) at the latter stage when it is small. Subsequently, this approach helps to reduce the likelihood of premature convergence and guides the search toward the promising search area.

Location Controlling Step

It is possible in both the global and local search stages that the particles move out of the search space; therefore, their locations must be corrected. In this paper, the harmony memory (HM) concept from the harmony search method is used in the DSO algorithm (the other operators of the HS have not been employed).

The location-correction strategy in the DSO algorithm is similar to the one defined for heuristic particle swarm ant colony optimization (HPSACO) (Kaveh and Talatahari 2009b) and heuristic particle swarm optimization (HPSO) (Li et al. 2007). In the HS algorithm, the HM stores the feasible vectors, which are all in the feasible space. The HM size determines how many vectors it stores. A new vector is generated by randomly selecting the components of different vectors in the HM. Therefore, the new vector does not violate the variables' boundaries, but it is not certain whether it violates the problem-specific constraints. When it is generated, the HM will be updated by accepting this new vector if it gets a better solution and deletes the worst vector (Li et al. 2007). In the same way, the proposed DSO stores the *feasible* and *good* solutions, as does the HM in the HS scheme. Hence, the particle violating the variables' boundaries can be generated randomly again by the same technique from the best position of the particle up to current iteration.

Modified Feasible-Based Mechanism Added to the DSO

In the proposed DSO algorithm, a modified feasible-based mechanism (FBM) is also used to handle the problem-specific constraints. The original FBM, also known as the constraint tournament selection, was introduced by Deb (2000), in which pair-wise solutions are compared using the following rules:

1. Any feasible solution is preferred to any infeasible solution;
2. Between two feasible solutions, the one having better objective function value is preferred; and
3. Between two infeasible solutions, the one having a smaller sum of constraint violation is preferred. This sum is calculated by

$$\text{Viol} = \sum_{j=1}^{n_g} \max[0, g_j(\mathbf{X})] \tag{10}$$

where $g_j = j$th inequality constraint; $\mathbf{X}$ = set of decision variables; and n_g = total number of inequality constraints.

By using the first and third rules, the search tends to the feasible region rather than infeasible region, and the second rule persuades the search to remain in the feasible region with good solutions. However, the original version of the FBM has problems in maintaining diversity in the population (Liu et al. 2010). To overcome this issue, in the proposed DSO an additional rule is defined as follows:

4. Infeasible solutions containing slight violations of the constraints (from 0.01 in the first iteration to 0.001 in the last iteration) are considered as feasible solutions.

For most of the engineering optimization problems, the global minimum locates on or close to the boundary of a feasible design space; therefore, by applying Rule 4, the particles can approach to the boundaries and can move toward the global minimum with a high probability.

Mathematical Model for WDS Optimization

Consider a WDS formed by a set of n_n nodes interconnected by n_p pipelines, given nodal demand and minimum head requirements. The mathematical model of minimum-cost design using available pipeline sizes can be expressed as in the following:

Find

$$\mathbf{\Phi} = \{d_1, d_2, \ldots, d_{n_p}\}$$

to minimize

$$C(\mathbf{\Phi}) = \sum_{i=1}^{n_p} l_i c_i \tag{11}$$

and subject to

$$\mathbf{G}(\mathbf{H}, \mathbf{\Phi}) = 0 \tag{12}$$

$$h_{\min} \leq \mathbf{H} \tag{13}$$

$$d_i \in S = \{S_m | m = 1, 2, \ldots, M\} \tag{14}$$

where design $\mathbf{\Phi}$ = set of n_p decision variables; and d_i = selected diameter for pipe i. In the objective function, Eq. (11), for each diameter there is an associated cost c_i of implementing its diameter and l_i is the pipe length. Eq. (12) defines constraints in which $\mathbf{G}$ is a vector function based on the conservation laws of the energy and mass. Inequality [Eq. (13)] is the constraint on nodal pressure head vector, $\mathbf{H}$, showing that the design pressure head at each demand node is above (or equal to) its corresponding minimum allowable pressure head, $h_{\min}$. S is a set of commercially available pipe diameters including M elements.

The constraints of the optimal design problem of WDSs can be grouped into the following: size limitation [Eq. (14)], minimum required pressure head [Eq. (13)], and hydraulic constraints [Eq. (12)]. Size limitation constraints reduce the parameter space to a discrete one. The proposed DSO-based model has an alternative to fix the resolution of the parameter space to be searched. This can be adjusted to the number of commercially available pipe diameters, and each parameter can take values from one to the number of commercial pipe sizes (M). This number is used as an index for the choice of diameters; therefore, the DSO algorithm will search for the optimal set of pipe indices instead of the optimal set of diameters. To handle the minimum nodal head constraints, a modified FBM approach is used as described previously. Finally, the hydraulic constraints are handled by a network simulation model. In this study, the DSO algorithm is coupled with the widely used water distribution network software, *EPANET 2* (Rossman 2000).

Numerical Results

In this section, three commonly used benchmark-design problems are optimized to test the effectiveness of the proposed method. The algorithms were first implemented in *MATLAB* and the optimization runs were carried out on a computer with an Intel Core i5 CPU, 2.53-GHz processor, and 3.00 GB RAM. These examples include three well-known networks:

- GoYang network;
- Hanoi network; and
- Balerma irrigation network.

The optimum design problem of the GoYang network was presented by Kim et al. (1994) in South Korea. It consists of 22 nodes, 30 pipes, and 9 loops, and is fed by a pump (4.52 kW) from a reservoir with a 71-m fixed head. The pipeline network for water supply in Hanoi (Vietnam) was proposed by Fujiwara and

Table 1. Summary of Case Study Characteristics

Case study	Number of decision variables	Number of commercially available pipe sizes	Search space size
GoYang	30	8	$8^{30} = 1.24 \times 10^{27}$
Hanoi	34	6	$6^{34} = 2.86 \times 10^{26}$
Balerma	454	10	10^{454}

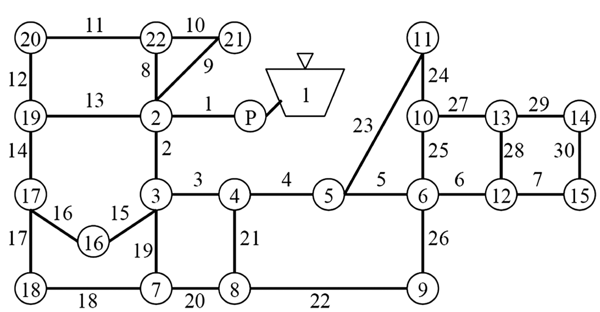

Fig. 1. Layout of the GoYang network

Table 2. Performance Comparison for Case Study 1

Method	Best cost ($)	Average cost ($)	Average number of evaluations to find best solutions	Maximum allowable evaluations	Percentage of trials with best solution found (%)
NLP[a]	179,143	—	—	—	—
HS[b]	177,136	—	—	10,000	—
GA[c]	177,061	177,706	12,683	25,000	4
DE[c]	177,010	177,013	8,750	25,000	52
DSO[d]	177,010	177,012	7,632	10,000	70

[a]Kim et al. (1994).
[b]Geem et al. (2002).
[c]Dong et al. (2012).
[d]Results for the DSO are based on 50 runs.

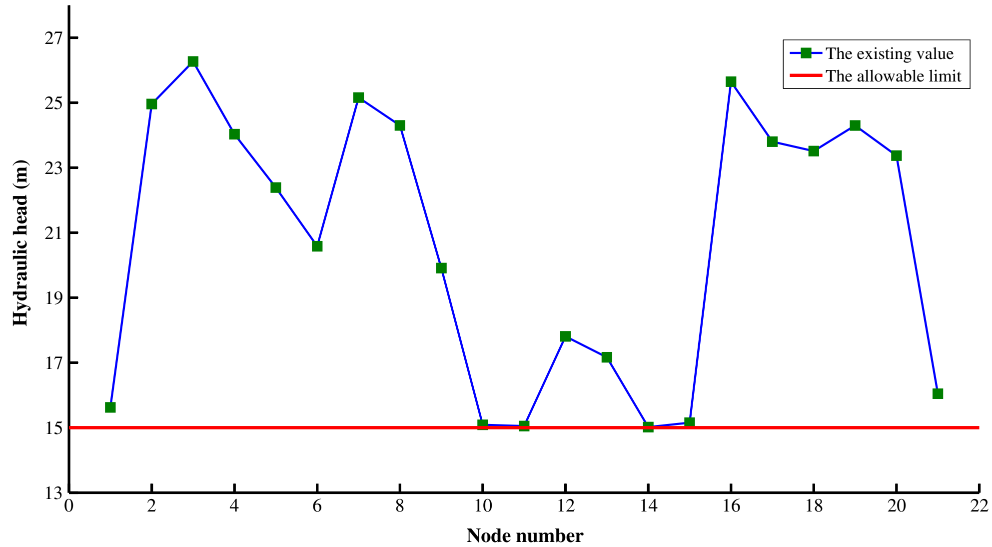

Fig. 2. Allowable and existing pressure head for case study 1 obtained by DSO

Khang (1990). This network consists of 32 nodes, 34 pipes, and 3 loops. The network has no pumping station as it is fed by gravity from a reservoir, and the minimum pressure-head requirement at all nodes is fixed at 30 m. The third test problem concerns the optimization of the Balerma irrigation network, located in the province of Almería, Spain, which was presented by Reca and Martinez (2006). This large-scale network has a total of 443 demand nodes supplied by four source nodes. There are 454 pipes, arranged in eight loops. The minimum required pressure head is 20 m for each node. The number of decision variables and the search-space size for each case study are given in Table 1.

Case Study 1: GoYang Network

The pipeline of the GoYang network, shown in Fig. 1, is derived from a WDS in South Korea. The cost of commercially available pipe sizes (80, 100, 125, 150, 200, 250, 300, 350; in mm) is (37.89; 38.933; 40.563; 42.554; 47.624; 54.125; 62.109; 71.524; in dollars/meter), which have a Hazen–Williams coefficient of 100. The minimum head limitation is 15 m above the ground level.

Table 2 reports the best results and the required number of evaluations to find the best solutions by the proposed DSO algorithm and some selected metaheuristic algorithms. In this case study, the population size is set to $N = 20$. The proposed DSO found the best feasible solution of \$177,010, whereas the original cost of the GoYang network was \$179,429. Kim et al. (1994) solved this problem using a projected Lagrangian algorithm (NLP) supported by GAMS/MINOS, and then converted the continuous diameters to discrete commercial diameters. Dong et al. (2012) found the current best-known solution (\$177,010) for the GoYang network by DE algorithm, spending 8,750 evaluations. The proposed DSO took 7,632 evaluations to find the current best solution, which is lower than those of the other algorithms. Comparing with other advanced algorithms such as HS and GA demonstrates the effectiveness and efficiency of the DSO method in solving this problem.

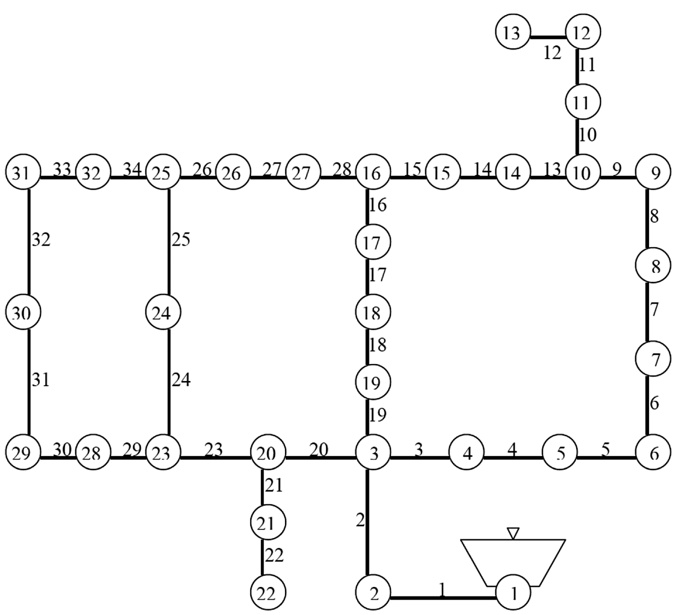

Fig. 3. Layout of the Hanoi problem

Table 3. Performance Comparison for Case Study 2

Method	Best cost ($\$10^6$)	Average cost ($\$10^6$)	Average evaluations to find best solutions	Maximum allowable evaluations	Number of runs
BB-BC[a]	6.224	6.292	26,000	30,000	50
PSO[b]	6.160	6.445	16,500	30,000	10
HBMO[c]	6.117	6.180	15,955	—	50
MMAS[d]	6.134	6.394	85,571	100,000	20
HD-DDS[e]	6.081	6.252	N/A	100,000	50
SADE[f]	6.081	6.090	60,532	74,876	50
GHEST[g]	6.081	6.175	50,134	—	60
DE[h]	6.081	—	48,724	100,000	50
PSO-DE[b]	6.081	6.366	40,200	60,000	10
DSO	6.081	6.135	39,280	60,000	50

[a]Tahershamsi et al. (2012).
[b]Sedki and Ouazar (2012).
[c]Mohan and Babu (2010).
[d]Zecchin et al. (2006).
[e]Tolson et al. (2009).
[f]Zheng et al. (2013).
[g]Bolognesi et al. (2010).
[h]Suribabu (2010).

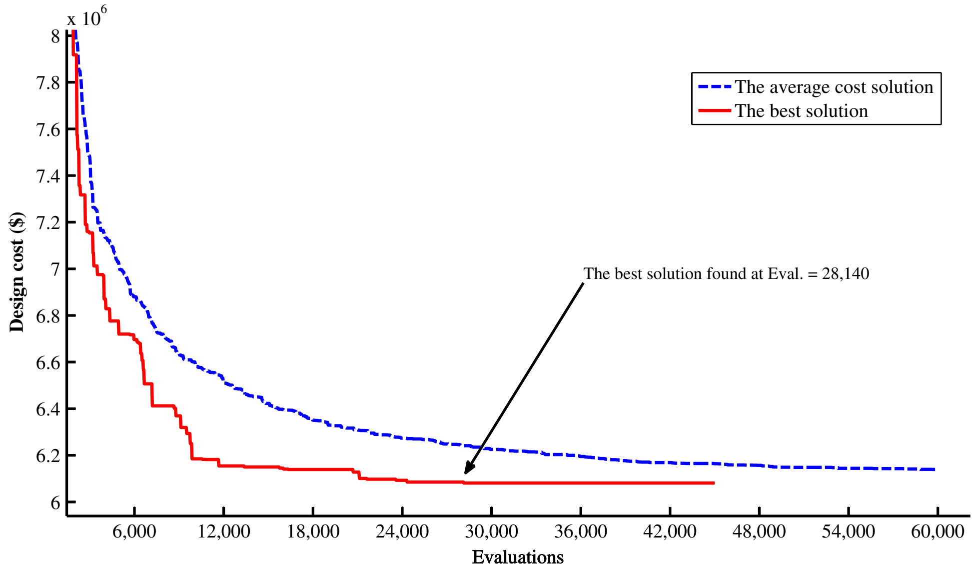

Fig. 4. Convergence curve of the DSO for case study 2 (results are based on 50 runs)

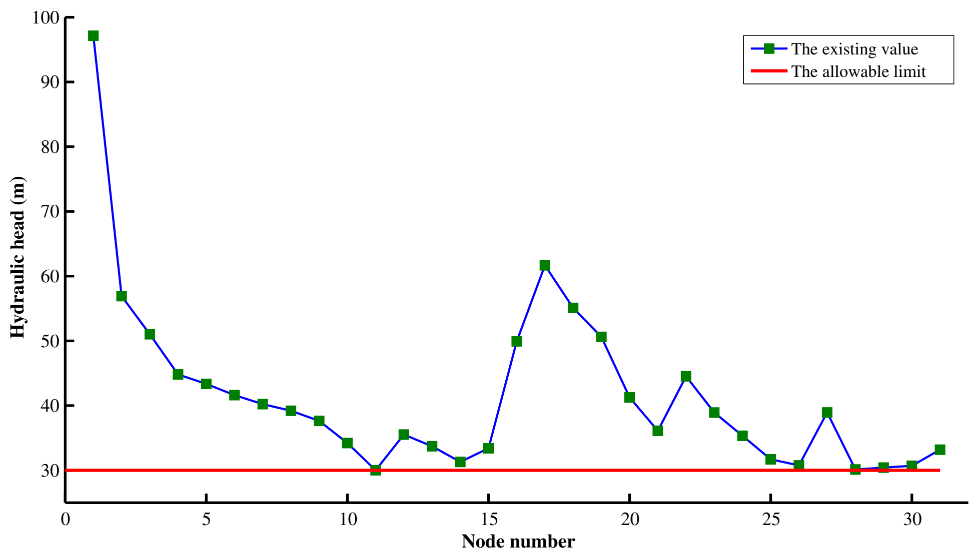

Fig. 5. Allowable and existing pressure head for case study 2 obtained by DSO

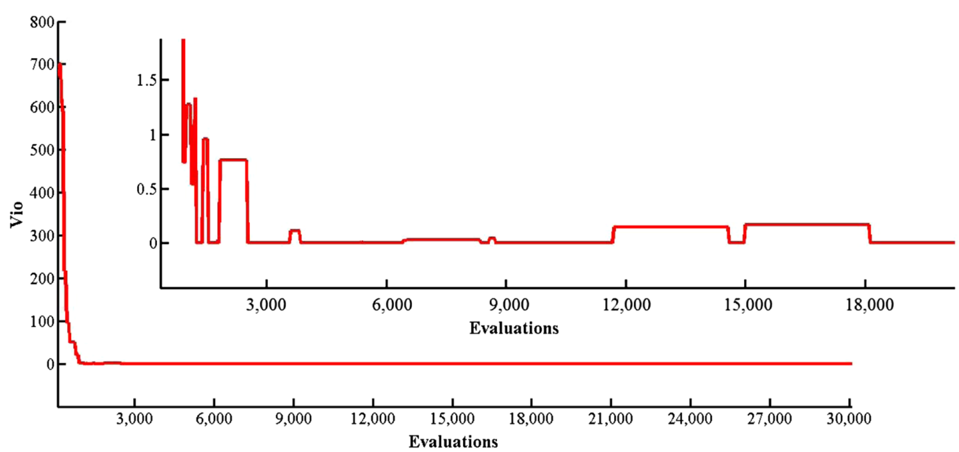

Fig. 6. Variation of constraint violation with the number of evaluations for case study 2

As given in Table 2, the DE algorithm obtained the current best solution with a frequency of 52%, whereas it is 70% for the DSO algorithm, which is better than those of the GA and DE techniques. In addition, using the DSO model, the maximum cost and standard deviation obtained from 50 runs with different initial conditions are $177,065 and 12.62, respectively. To illustrate the performance of the constraint handling approach, Fig. 2 shows the hydraulic head for each node in the final optimal design. As shown in Fig. 2, the minimum value for pressure head is equal to 15.018 m (in node 14), which is above the minimum required head.

Case Study 2: Hanoi Network

The configuration of the WDS in Hanoi, Vietnam, is shown in Fig. 3. The costs of commercially available pipe sizes [304.8, 406.4, 508.0, 609.6, 762.0, 1016.0; in mm (12, 16, 20, 24, 30, 40; in inches)] are (45.73, 70.40, 98.38, 129.30, 180.80, 278.30) in dollars/meter. In this case study, the population size is set to $N = 30$.

The results obtained using the DSO algorithm and those previously reported in the literature are summarized in Table 3. The new DSO algorithm found the best feasible solution of $\$6.0811 \times 10^6$ in a minimum of 28,140 evaluations and the average cost of $\$6.1346 \times 10^6$, as shown in Fig. 4. Using the DSO model, the maximum cost and standard deviation obtained from 50 runs with random initial points are $\$6.2763 \times 10^6$ and 7.3318×10^4, respectively. The better performance of the DSO, comparing with the PSO and BB-BC for case study 2, could be attributed to its greater ability to explore efficiently the search space with the aid of hybridization of PSO and BB-BC operators, thus enhancing the chances of finding the global optimum. Moreover, a comparison with the other standard and hybrid swarm-based algorithms such as MMAS, self-adaptive differential evolution (SADE), and PSO-DE demonstrates the superiority of the proposed method in terms of its searching capability (i.e., the ability to find near globally optimal solutions), while being significantly more computationally efficient.

In terms of efficiency, the proposed DSO used an average number of 39,280 function evaluations to find the optimal solutions, which is fewer than all of the other algorithms except BB-BC, PSO, and HMBO; however, the quality of the final solution (average and best cost) obtained by these methods was worse than that produced by the DSO. This indicates that the proposed DSO is able to find optimal solutions more quickly than other algorithms. Finally, Fig. 5 compares the allowable and existing hydraulic head values at the demand nodes corresponding to the least-cost design of case study 2. Based on Fig. 5, the minimum value for pressure head is equal to 30.006 m (in node 11), which is above the minimum required head. Moreover, to illustrate the performance of the constraint handling approach, Fig. 6 shows the rate of reduction on infeasibilities with the number of evaluations. As indicated in Fig. 6, when the particles tend to the infeasible region (Vio > 0), the algorithm is forced to fly back to the feasible space (Vio = 0). In this figure, the Vio (constraint violation) parameter is calculated by Eq. (10).

Case Study 3: Balerma Irrigation Network

There are 454 pipes in the Balerma network (Fig. 7), which should be designed using a set of 10 PVC pipes with diameters between 125 and 600 mm and an absolute roughness coefficient set to 0.0025 mm. In addition, the Darcy-Weisbach equation was adapted to calculate the head losses.

Table 4 outlines the performance comparison of the DSO algorithm with those of the previously reported ones in the literature. The optimal design obtained using the proposed DSO showed an

Fig. 7. Layout of the Balerma network

excellent agreement with the previous designs. As indicated in Table 4, the best solution found by the DSO algorithm for case study 3 is €1.9869×10^6, which is 0.20 and 2.31% higher than the best solutions reported by Zheng et al. (2013) (€1.983×10^6) and Tolson et al. (2009) (€1.941×10^6) using SADE and HD-DDS methods, respectively; however, they were better than solutions obtained by the GA, HS, and DE.

From Table 4, the HD-DDS obtained the best solution requiring the total computational budget of 30 million evaluations, whereas the DSO used only 3 million evaluations to finally converge, which is less than those required by the GA and HS. In addition, the average number of evaluations required for the DSO algorithm to first reach the best solutions was 1.6 million. Fig. 8 depicts the

Table 4. Performance Comparison for Case Study 3

Method	Best cost (€10^6)	Average cost (€10^6)	Average evaluations to find best solutions	Maximum allowable evaluations
GA[a]	2.302	2.334	—	10×10^6
HS[b]	2.018	—	—	10×10^6
DE[c]	1.998	2.031	2.3×10^6	2.4×10^6
SADE[c]	1.983	1.995	1.2×10^6	1.3×10^6
HD-DDS[d]	1.941	—	—	30×10^6
DSO[e]	1.987	2.017	1.6×10^6	3×10^6

[a]Reca and Martinez (2006).
[b]Geem (2009).
[c]Zheng et al. (2013).
[d]Tolson et al. (2009).
[e]Results for the DSO are based on 10 runs.

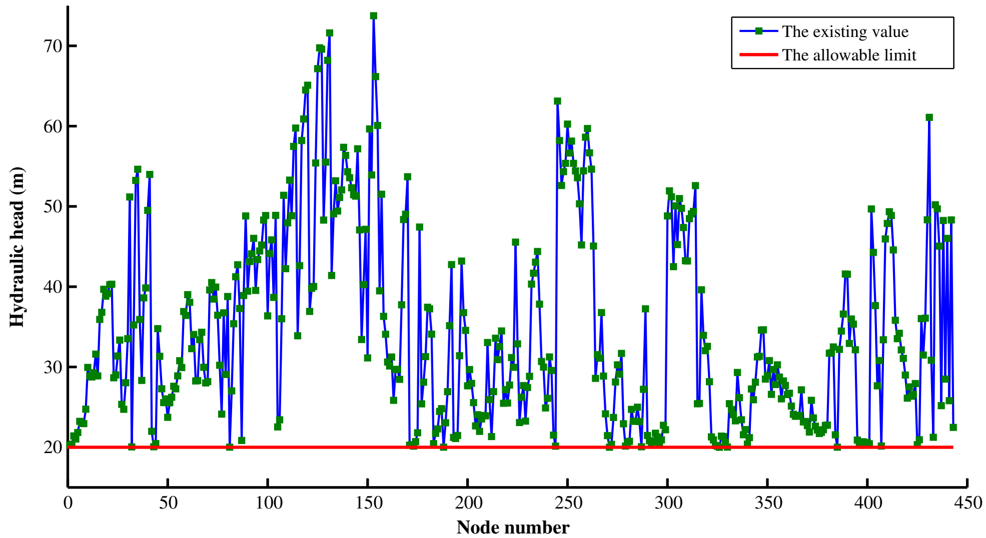

Fig. 8. Allowable and existing pressure head for case study 3 obtained by DSO

Table 5. Performance Comparison for Case Study 3 with the Same Computational Budget

Method	Cost (€10^6)	Fixed number of evaluations
GA[a]	3.555	454,000
SA[a]	3.476	454,000
MSATS[b]	3.298	454,000
PSHS[c]	2.633	454,000
HS[c]	2.601	454,000
DSO	2.403	454,000

[a]Baños et al. (2010).
[b]Reca et al. (2008).
[c]Geem (2009).

allowable (20 m) and actual pressure-head values for the optimal solution of case study 3 obtained by the DSO algorithm. The minimum value for pressure head is equal to 20.0009 m (in node 326), which illustrates the effectiveness of the modified feasible-based mechanism.

To guarantee the same conditions for comparison with the performance of different algorithms, one approach is to carry out the fixed number of evaluations of the fitness function. The maximum number of evaluations, E_m, should depend on the network complexity, which is a function of the number of links, n_p, and the number of commercially available pipe diameters. Baños et al. (2010) proposed the following equation to set the number of evaluations as

$$E_m = K \times n_p \times \log_{10}(M)$$

Considering a constant $K = 1{,}000$, $n_p = 454$, and $M = 10$, the resulting number of function evaluations is 454,000 for the Balerma network (Baños et al. 2010). The efficiency of the DSO in solving this real-world and complex optimization problem is verified by Table 5.

As indicated in Table 5, the DSO algorithm outperforms the advanced algorithms such as GA, SA, and HS in terms of finding the best solution in the same computational overhead. In addition, a comparison with the hybrid particle-swarm harmony search (PSHS) and mixed simulated annealing and tabu search (MSATS) algorithms demonstrates the effectiveness of the proposed method in solving this problem with the best solution quality.

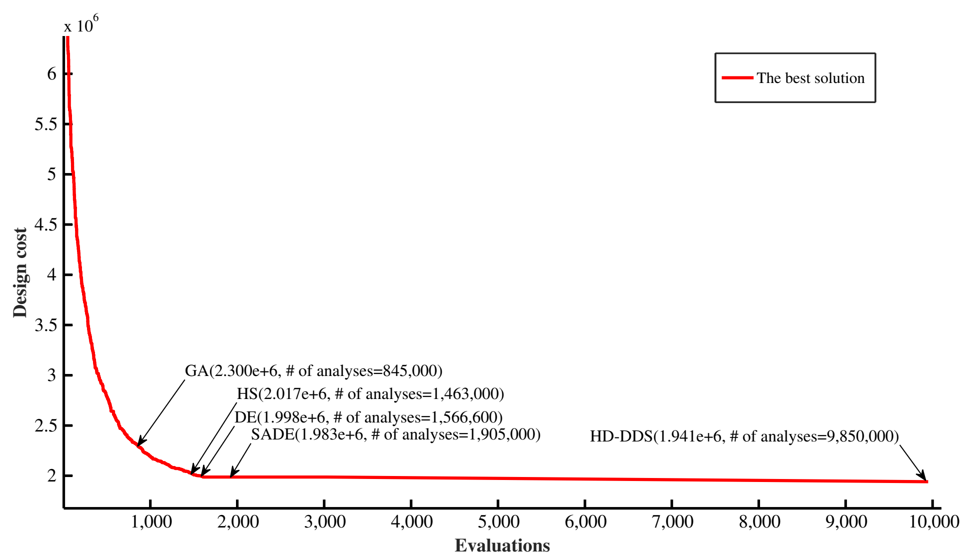

Fig. 9. Convergence history for the DSO and compared with the other methods

When studying the ability of the new algorithm to find the optimum point, the problem is resolved without considering any limit on the number of analyses. The stopping criteria are defined in a way that the algorithm works until reaching the so far known optimum design (€1.941×10^6). As shown in Fig. 9, the DSO can find results better than those of the GA and HS after 845,000 and 1,463,000 analyses, respectively (GA and HS need 10,000,000 analyses). It is also better than the DE after 1,566,600 analyses (DE needs 2,400,000 analyses), and finally it needs just 9,850,000 analyses to reach the result of the HD-DDS, whereas the HD-DDS needs 30,000,000 analyses. This clearly shows the superiority of the new algorithm.

Summary and Conclusions

Swarm-based optimization algorithms, inspired by the study of collective behavior in self-organizing systems, usually consist of a population of individuals that takes effect between each other and environment. In this paper, a vast review of swarm-based methods and their applications to the WDS design problem has been performed, and most of the improvements proposed in the literature have been surveyed.

Although many examples of successful applications of swarm-based methods to the WDS optimization have been given, it is known that the canonical version of these heuristic search techniques, particularly the PSO algorithm, has its own disadvantages. To overcome the disadvantages of the original PSO, this paper presents a newly developed swarm-based optimizer for the cost optimization of WDSs. The proposed DSO is based on the accelerated PSO, BB-BC algorithm, HS scheme, and the modified feasible-based mechanism.

The global searching step of the DSO uses a modified position-updating formula obtained by hybridizing the accelerated PSO and the BB-BC algorithm so that the center of mass point from the big crunch phase of the BB-BC is incorporated into the position-updating equation. A local searching step is followed after the global searching step and in this step; more searches around the current best solutions are performed. In the local searching step, based on the big bang phase of the BB-BC, a new particle-updating strategy is presented to overcome the premature convergence of the PSO. The new updating strategy is based on the concept of possibility measurement to model the lack of information about the true optimality of the global best solution. In addition, an HS-based method is used to correct the location of generated particles that move out of search space in both the local and global levels. Furthermore, a modified version of the feasible-based mechanism is used to handle the problem-specific constraints to demonstrate the performance of the DSO, it is applied to the minimum-cost single-objective WDS optimization problems. A total of three WDS case studies (GoYang, Hanoi, and Balerma networks) with increased scales and complexity were used to verify the effectiveness of the proposed DSO algorithm. For each case study, a summary of the results are given in Table 6. For case studies 1 and 2, numerical experiments indicate that compared with other metaheuristics such as the PSO, BB-BC, HS, ACO, GA and DE, the new algorithm performed the best in terms of finding optimal solutions with good quality and great efficiency. For the last large-scale problem (case study 3), the proposed DSO also exhibited a comparable performance to the other algorithms. Its optimal design demonstrated excellent agreement with the previous designs reported in the literature, because the DSO updating strategy can effectively avoid the local optimality with a nonincreasing uncertainty. The coupled DSO-EPANET model is a multipurpose framework and can be extended to tackle other water-network-management problems, which is appealing for its practical applications. Future studies on this area may include (1) an investigation of the performance of the proposed DSO algorithm in solving more complex and real large-size networks, (2) the application of the proposed DSO algorithm to deal with the optimal operation of WDSs, and (3) the application of the proposed DSO algorithm to tackle the multiobjective problems.

Table 6. Summary of Results of the Proposed DSO Method for Each Case Study

Case study	Best cost	Average cost	Average evaluations to find best solutions	Maximum allowable evaluations	Number of runs
GoYang	177,010	177,012	7,632	10,000	50
Hanoi	6.081×10^6	6.135×10^6	39,280	60,000	50
Balerma	1.987×10^6	2.017×10^6	1,600,000	3,000,000	10

Acknowledgments

The authors gratefully acknowledge the facilities provided by the Global Institute for Water Security (GIWS) at the University of Saskatchewan and the Department of Civil Engineering at the University of Tabriz.

References

Alatas, B. (2011). “Uniform big bang–chaotic big crunch optimization.” *Commun. Nonlinear Sci. Numer. Simul.*, 16(9), 3696–3703.

Baños, R., Gil, C., Reca, J., and Montoya, F. G. (2010). “A memetic algorithm applied to the design of water distribution networks.” *Appl. Soft Comput.*, 10(1), 261–266.

Bansal, J. C., and Deep, K. (2009). “Optimal design of water distribution networks via particle swarm optimization.” *Proc., Advance Computing Conf.*, IEEE, 1314–1316.

Bolognesi, A., Bragalli, C., Marchi, A., and Artina, S. (2010). “Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks.” *Adv. Eng. Software*, 41(5), 792–801.

Chu, C. W., Lin, M. D., Liu, G. F., and Sung, Y. H. (2008). “Application of immune algorithms on solving minimum-cost problem of water distribution network.” *Math. Comput. Modell.*, 48(11–12), 1888–1900.

Corte, A., and Sörensen, K. (2013). “Optimisation of gravity-fed water distribution network design: A critical review.” *Eur. J. Oper. Res.*, 228(1), 1–10.

Deb, K. (2000). “An efficient constraint handling method for genetic algorithms.” *Comput. Methods Appl. Mech. Eng.*, 186(2–4), 311–338.

Dong, X., Liu, S., Tao, T., Li, S., and Xin, K. (2012). “A comparative study of differential evolution and genetic algorithms for optimizing the design of water distribution systems.” *J. Zhejiang Univ. -SCIENCE A*, 13(9), 674–686.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). “The ant system: Optimization by a colony of cooperating agents.” *IEEE Trans. Syst. Man Cybern. Part B: Cybern.* 26(1), 29–41.

Eberhart, R. C., and Kennedy, J. (1995). “A new optimizer using particle swarm theory.” *Proc., 6th Int. Symp. on Micro Machine and Human Science*, IEEE, 39–43.

EPANET [Computer software]. U.S. Environmental Protection Agency, Washington, DC.

Erol, O. K., and Eksin, I. (2006). “New optimization method: Big bang–big crunch.” *Adv. Eng. Software*, 37(2), 106–111.

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

Gandomi, A. H., Yun, G. J., Yang, X. S., and Talatahari, S. (2013). "Chaos-enhanced accelerated particle swarm optimization." *Commun. Nonlinear Sci. Numer. Simul.*, 18(2), 327–340.

Geem, Z. W. (2009). "Particle-swarm harmony search for water networks design." *Eng. Optim.*, 41(4), 297–311.

Geem, Z. W., Kim, J. H., and Loganathan, G. V. (2001). "A new heuristic optimization algorithm: Harmony search." *Simulation*, 76(2), 60–68.

Geem, Z. W., Kim, J. H., and Loganathan, G. V. (2002). "Harmony search optimization: Application to pipe network design." *Int. J. Modell. Simul.*, 22(2), 125–133.

Gibbs, M. S., Maier, H. R., and Dandy, G. C. (2014). "Using characteristics of the optimisation problem to determine the genetic algorithm population size when the number of evaluations is limited." *Environ. Modell. Software*, 69, 226–239.

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

Kaveh, A., Sheikholeslami, R., Talatahari, S., and Keshvari-Ilkhichi, M. (2014). "Chaotic swarming of particles: A new method for size optimization of truss structures." *Adv. Eng. Software*, 67, 136–147.

Kaveh, A., and Talatahari, S. (2009a). "A particle swarm ant colony optimization for truss structures with discrete variables." *J. Constr. Steel Res.*, 65(8–9), 1558–1568.

Kaveh, A., and Talatahari, S. (2009b). "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures." *Comput. Struct.*, 87(5–6), 267–283.

Kaveh, A., and Talatahari, S. (2010). "A novel heuristic optimization method: Charged system search." *Acta Mech.*, 213(3–4), 267–289.

Kim, J. H., Kim, T. G., Kim, J. H., and Yoon, Y. N. (1994). "A study on the pipe network system design using non-linear programming." *J. Korean Water Resour. Assoc.*, 27(4), 59–67.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). "Optimization by simulated annealing." *Science*, 220(4598), 671–680.

Li, L. J., Huang, Z. B., Liu, F., and Wu, Q. H. (2007). "A heuristic particle swarm optimizer for optimization of pin connected structures." *Comput. Struct.*, 85(7–8), 340–349.

Liu, H., Cai, Z., and Wang, Z. (2010). "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization." *Appl. Soft Comput.*, 10(2), 629–640.

Maier, H. C., et al. (2003). "Ant colony optimization for design of water distribution systems." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)0733-9496(2003)129:3(200), 200–209.

Maier, H. R., et al. (2014). "Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions." *Environ. Modell. Software*, 62, 271–299.

MATLAB [Computer software]. MathWorks, Natick, MA.

Mohan, S., and Babu, K. (2010). "Optimal water distribution network design with honey-bee mating optimization." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000018, 117–126.

Montalvo, I., Izquierdo, J., Perez, R., and Herrera, M. (2010). "Improved performance of PSO with self-adaptive parameters for computing the optimal design of water supply systems." *Eng. Appl. Artif. Intell.*, 23(5), 727–735.

Montalvo, I., Izquierdo, J., Perez, R., and Tung, M. M. (2008). "Particle swarm optimization applied to the design of water supply systems." *Comput. Math. Appl.*, 56(3), 769–776.

Ostfel, A., and Tubaltzev, A. (2008). "Ant colony optimization for least-cost design and operation of pumping water distribution systems." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)0733-9496(2008)134:2(107), 107–118.

Reca, J., and Martinez, J. (2006). "Genetic algorithms for the design of looped irrigation water distribution networks." *Water Resour. Res.*, 42(5).

Reca, J., Martinez, J., Gil, C., and Baños, R. (2008). "Application of several meta-heuristic techniques to the optimization of real looped water distribution networks." *Water Resour. Manage.*, 22(10), 1367–1379.

Rossman, L. A. (2000). *EPANET 2 user's manual*, U.S. EPA, Cincinnati.

Sedki, A., and Ouazar, D. (2012). "Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems." *Adv. Eng. Inform.*, 26(3), 582–591.

Sheikholeslami, R., Kaveh, A., Tahershamsi, A., and Talatahari, S. (2014). "Application of charged system search algorithm to water distribution networks optimization." *Int. J. Optim. Civ. Eng.*, 4(1), 41–58.

Suribabu, C. R. (2010). "Differential evolution algorithm for optimal design of water distribution networks." *J. Hydroinform.*, 12(1), 66–82.

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

Tahershamsi, A., Kaveh, A., Sheikholeslami, R., and Talatahari, S. (2012). "Big bang-big crunch algorithm for least-cost design of water distribution systems." *Int. J. Optim. Civ. Eng.*, 2(1), 71–80.

Talatahari, S., Kheirollahi, M., Farahmandpour, S., and Gandomi, A. H. (2013a). "A multi-stage particle swarm for optimum design of truss structures." *Neural Comput. Appl.*, 23(5), 1297–1309.

Talatahari, S., Mohajer Rahbari, N., and Kaveh, A. (2013b). "A new hybrid optimization algorithm for recognition of hysteretic non-linear systems." *KSCE J. Civ. Eng.*, 17(5), 1099–1108.

Tolson, B. A., Asadzadeh, M., Maier, H. R., and Zecchin, A. (2009). "Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization." *Water Resour. Res.*, 45(12), W12416.

Vairavamoorthy, K., and Ali, M. (2005). "Pipe index vector: A method to improve genetic algorithm-based pipe optimization." *J. Hydr. Eng.*, 10.1061/(ASCE)0733-9429(2005)131:12(1117), 1117–1125.

Vasan, A., and Simonovic, S. P. (2010). "Optimization of water distribution network design using differential evolution." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)0733-9496(2010)136:2(279), 279–287.

Wang, G. G., Gandomi, A. H., Yang, X. S., and Alavi, A. H. (2014). "A novel improved accelerated particle swarm optimization algorithm for global numerical optimization." *Eng. Comput.*, 31(7), 1198–1220.

Xinchao, Z. (2010). "A perturbed particle swarm algorithm for numerical optimization." *Appl. Soft Comput.*, 10(1), 119–124.

Yang, X. S. (2010). *Engineering optimization: An introduction with metaheuristic applications*, Wiley, New York.

Yang, X. S., Deb, S., and Fong, S. (2011). "Accelerated particle swarm optimization and support vector machine for business optimization and applications." *Proc., Networked Digital Technologies*, Springer, Berlin, 53–66.

Yates, D. F., Templemen, A. B., and Boffey, T. B. (1984). "The computational complexity of the problem of determining least capital cost designs for water supply networks." *Eng. Optim.*, 7(2), 143–155.

Zecchin, A. C., Simpson, A. R., Maier, H. R., Leonard, M. L., Roberts, A. J., and Berrisford, M. J. (2006). "Application of two ant colony optimisation algorithms to water distribution system optimization." *Math. Comput. Modell.*, 44(5–6), 451–468.

Zecchin, A. C., Simpson, A. R., Maier, H. R., Marchi, A., and Nixon, J. B. (2012). "Improved understanding of the searching behavior of ant colony optimization algorithms applied to the water distribution design problem." *Water Resour. Res.*, 48(9), W09505.

Zecchin, A. C., Simpson, A. R., Maier, H. R., and Nixon, J. B. (2005). "Parametric study for an ant algorithm applied to water distribution system optimization." *IEEE Trans. Evol. Comput.*, 9(2), 175–191.

Zheng, F., Simpson, A. R., and Zecchin, A. C. (2011). "A combined NLP differential evolution algorithm approach for the optimization of looped water distribution systems." *Water Resour. Res.*, 47(8), W08531.

Zheng, F., Simpson, A. R., and Zecchin, A. C. (2014). "Coupled binary linear programming-differential evolution algorithm approach for water distribution system optimization." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)WR.1943-5452.0000367, 585–597.

Zheng, F., Zecchin, A. C., and Simpson, A. R. (2013). "Self-adaptive differential evolution algorithm applied to water distribution system optimization." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000208, 148–158.

Zheng, F., Zecchin, A. C., and Simpson, A. R. (2015). "Investigating the run-time searching behavior of the differential evolution algorithm applied to water distribution system optimization." *Environ. Modell. Software*, 69, 292–307.