



## RESEARCH ARTICLE

10.1002/2016WR019854

# GALAXY: A new hybrid MOEA for the optimal design of Water Distribution Systems

Q. Wang<sup>1</sup> , D. A. Savić<sup>2</sup>, and Z. Kapelan<sup>2</sup> 

<sup>1</sup>School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, China, <sup>2</sup>Centre for Water Systems, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, UK

### Key Points:

- A new and efficient hybrid algorithm is proposed for solving the discrete, multiobjective design of Water Distribution Systems
- The method is robust and easy-to-use by alleviating the parameterization issue of Multiobjective Evolutionary Algorithms to a great extent
- The hybrid algorithm outperforms three baseline methods in terms of exploration and exploitation consistently based on comparative studies

### Supporting Information:

- Supporting Information S1

### Correspondence to:

Q. Wang,  
wangqiguangzhou@163.com

### Citation:

Wang, Q., D. A. Savić, and Z. Kapelan (2017), GALAXY: A new hybrid MOEA for the optimal design of Water Distribution Systems, *Water Resour. Res.*, 53, doi:10.1002/2016WR019854.

Received 1 OCT 2016

Accepted 5 FEB 2017

Accepted article online 18 FEB 2017

**Abstract** A new hybrid optimizer, called **genetically adaptive leaping algorithm for approximation and diversity** (GALAXY), is proposed for dealing with the discrete, combinatorial, multiobjective design of Water Distribution Systems (WDSs), which is NP-hard and computationally intensive. The merit of GALAXY is its ability to alleviate to a great extent the parameterization issue and the high computational overhead. It follows the generational framework of Multiobjective Evolutionary Algorithms (MOEAs) and includes six search operators and several important strategies. These operators are selected based on their leaping ability in the objective space from the global and local search perspectives. These strategies steer the optimization and balance the exploration and exploitation aspects simultaneously. A highlighted feature of GALAXY lies in the fact that it eliminates majority of parameters, thus being robust and easy-to-use. The comparative studies between GALAXY and three representative MOEAs on five benchmark WDS design problems confirm its competitiveness. GALAXY can identify better converged and distributed boundary solutions efficiently and consistently, indicating a much more balanced capability between the global and local search. Moreover, its advantages over other MOEAs become more substantial as the complexity of the design problem increases.

## 1. Introduction

Water Distribution Systems (WDSs) are one of the essential city infrastructure systems. They convey potable water through a network of pipes (and other elements) from treatment plants to households, commercial, and industrial users. Ideally, water is to be supplied continuously, at adequate pressure and flow, while satisfying microbial and chemical quality standards. Water Distribution Systems are designed not only to meet peak demands in the future, but also to ensure efficient operation over short time periods (e.g., a day or a week). A new WDS, especially if it consists of a large network of elements, normally requires substantial financial investment. Therefore, it is necessary to perform capital and operational cost analyses and consider a number of candidate solutions to determine the best overall solution to implement.

Historically, the design of WDSs relied mainly on engineers' knowledge and experience to meet all requirements. However, apart from trivial problems, this is not sufficient to cope with designing large and complex networks, especially for cities undergoing rapid urbanization. A number of optimization algorithms, such as linear and nonlinear programming, were initially used to address this problem [Alperovits and Shamir, 1977; Kim et al., 1994; Schaake and Lai, 1969]. This was followed by heuristic methods, such as Evolutionary Algorithms (EAs), including the most commonly used Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Differential Evolution (DE), to name a few [Simpson et al., 1994; Suribabu and Neelakantan, 2006; Zheng et al., 2014a]. The interest in optimization has been complemented by the developments in network modeling, which is used to check that the WDS is or will be operating to the required standard. A freely available modeling tool EPANET [Rossman, 2000], which is the basis of a number of commercially available software, is regarded as an "industry standard" by the worldwide water community.

Applying optimization, especially EAs to solve the optimal design of WDSs requires hydraulic simulations to predict WDS behavior, which is normally done via EPANET Programmer's Toolkit [Farmani et al., 2005; Fu et al., 2012]. Different problem formulations have been presented to optimize WDS design. For example, the least cost design of WDS was a dominating formulation used by the initial adopters by focusing on the minimization of capital costs. However, this was later criticized mainly due to least cost solutions failing to

capture all decision makers' concerns and other performance criteria for the assessment of WDSs [Engelhardt *et al.*, 2000; Walski, 2001]. This was followed by the introduction of multiobjective formulations, which have received more attention in recent years. Consequently, this trend has stimulated the use of Multiobjective Evolutionary Algorithms (MOEAs) for the design of WDSs, such as the elitist nondominated sorting genetic algorithm II (NSGA-II) [Deb *et al.*, 2002].

However, the main criticisms of these MOEAs related to the robustness, accuracy and parameterization issues [Hadka and Reed, 2012; Kollat and Reed, 2006]. That is, they usually require a large number of function evaluations and multiple runs to find near-optimal solutions. Additionally, individual parameters of these algorithms should be fine-tuned, which is generally based on rules of thumb and/or trial-and-error approaches, thus being computationally expensive for large and/or complex design problems.

Recently, several hybrid algorithms have been proposed in order to improve the effectiveness and efficiency of EAs or MOEAs by combining different algorithms and/or strategies into a unified framework. The ideas of hybridization for the WDS design problems can be generally divided into three categories: (i) those using different techniques to provide a good initial population (rather than randomly generated), such as the heuristic-based, local representative cellular automata approach [Keedwell and Khu, 2005] and the graph decomposition based approach [Zheng *et al.*, 2014b]; (ii) those combining various EAs, MOEAs or local search methods during the evolutionary process; such as the Borg MOEA [Wang *et al.*, 2015] and the hybrid Pareto archived dynamically dimensioned search method [Asadzadeh and Tolson, 2012] (iii) those implementing the search space simplification (or reduction) before optimization, such as the dual-stage multiobjective optimization method [Zheng and Zecchin, 2014].

The comparative studies [Asadzadeh and Tolson, 2012; Hadka and Reed, 2013; Raad *et al.*, 2011; Reed *et al.*, 2013; Zheng *et al.*, 2014a, 2014b; Zheng and Zecchin, 2014] have shown that they can outperform the mainstream MOEAs on various test functions and benchmark problems in different fields. Following on from these works, this paper presents a new hybrid algorithm termed **genetically adaptive leaping algorithm for approximation and diversity (GALAXY)**, and compares its performance in solving the multiobjective WDS design problems with three well-known MOEAs from the literature, including the recently proposed Borg MOEA [Hadka and Reed, 2013].

The remainder of this paper is organized as follows. First, the multiobjective design of a WDS is formulated. Then, the GALAXY method is described in detail followed by a brief introduction to the representative MOEAs. Next, the benchmark problems used in this paper are given, and the experiments are set up. After comparing the results obtained from each algorithm, conclusions are drawn at the end.

## 2. Multiobjective Design of WDSs

Real-world applications of optimal WDS design often involve diverse criteria (e.g., cost, reliability, water quality), which are normally in conflict with each other. Thus, design is intrinsically multiobjective [Fu *et al.*, 2013]. However, many previous studies focused on the two-objective optimal design of WDSs, considering the minimization of the total costs and maximization of system benefits [Farmani *et al.*, 2005; Ostfeld *et al.*, 2014; Raad *et al.*, 2009, 2011]. Given the focus on optimization methods' comparison analysis conducted here (rather than formulating new WDS design approach), the two-objective approach was adopted.

In this paper, the total costs only include the initial capital expenditure for pipes (e.g., purchase, transportation, and installation) since they account for the major part of investment. The system benefit is assessed by a surrogate measure of network reliability, also known as network resilience (denoted as  $I_n$ ), which takes both the nodal surplus head and the uniformity of pipes connected to that node into account [Prasad and Park, 2004]. Resilience  $I_n$  has shown to be a suitable indicator compared to its alternatives [Raad *et al.*, 2010]. The mathematical formulations of the objective functions are shown in equations (1–3):

$$\text{minimize } Cost = \sum_{i=1}^{np} U_c(D_i) \times L_i \quad (1)$$

$$\text{maximize } I_n = \frac{\sum_{j=1}^{nn} U_j Q_j (H_j - H_j^{req})}{\left( \sum_{k=1}^{nr} Q_k H_k + \sum_{i=1}^{np} \frac{P_i}{\gamma} \right) - \sum_{j=1}^{nn} Q_j H_j^{req}} \quad (2)$$

where

$$U_j = \frac{\sum_{i=1}^{npj} D_{ij}}{npj \times \max \{D_{ij}\}} \quad (3)$$

and  $U_c$  is the unit pipe cost as a function of its diameter  $D_i$ ;  $L_i$  is the length of pipe  $i$ ;  $I_n$  is the network resilience;  $nn$  is the number of demand nodes;  $U_j$ ,  $Q_j$ ,  $H_j$ , and  $H_j^{req}$  are the uniformity, demand, actual head, and minimum head of node  $j$ , respectively;  $nr$  is the number of reservoirs;  $Q_k$  and  $H_k$  are the discharge and actual head at reservoir  $k$ ;  $npu$  is the number of pumps;  $P_i$  is the power of pump  $i$ ;  $\gamma$  is the specific weight of water;  $npj$  is the number of pipes connected to node  $j$ ; and  $D_{ij}$  is the diameter of pipe  $i$  connected to node  $j$ .

Discrete pipe diameters are decision variables, which are expressed via the diameter option indices to facilitate the coding scheme. In other words, a candidate design solution is a vector of integer values, which range from one up to the number of commercially available sizes. Therefore, the decision variables are of consecutive integer type, leading to the integer coding scheme as a natural choice.

The constraints for the two-objective WDS design problem can be explicit or implicit. The former are considered as soft constraints, i.e., can be violated during the search process. They normally relate to the hydraulic performance of a network. Nodal pressures at demand nodes and flow velocities in pipes are considered as soft constraints in this work + [Reca *et al.*, 2008]. The latter are the hard constraints that must be satisfied during optimization. They require solving conservation of mass and energy to determine the network's pressure heads, and are automatically satisfied by using the EPANET hydraulic solver.

### 3. Hybrid Algorithm: GALAXY

The majority of modern MOEAs were created following the principles of Darwinian evolution, i.e., the survival of the fittest and adaptation to the environment. *Padhye et al.* [2013] presented a unified approach to EAs in the context of real-parameter optimization of unimodal problems, called Evolutionary Optimization System (EOS). The EOS includes four key steps, which are *Initialization*, *Selection*, *Generation*, and *Replacement*. First, an initial population is randomly generated. The solutions are then evaluated using the objective functions. Second, the parents are selected from the population and offspring are produced. Then, the fitness of the offspring is evaluated using the objective functions. If some of them dominate the members of the current population, the dominated individuals are replaced by these offspring. This procedure is implemented repeatedly until a certain stopping criterion is satisfied. The EOS framework is also suitable for MOEAs, but the *Selection* step needs to adapt to the multiobjective perspective.

#### 3.1. Pseudocode of GALAXY

In theory, it is possible to propose a hybrid algorithm by altering any or all of the key EOS steps. However, the current version of GALAXY implements hybridization only at the *Generation* and *Replacement* steps because they play a key role in the evolutionary process. The pseudocode for GALAXY is illustrated in Figure 1 and the development of main components is explained in following subsections. It is also worth noting that GALAXY is not just an optimization method but more of a framework as it provides basic structure for creation of other hybrid algorithms.

#### 3.2. Search Operators of GALAXY

Typically, an MOEA intrinsically balances two aspects of EA optimization, which are known as exploration and exploitation [Blum and Roli, 2003]. The former is aimed at searching the space globally in order to identify the near-optimal regions. In contrast, the latter attempts to conduct a fine-grained local search in order to improve the quality of current solutions. As such, an ideal process (see Figure 2a) would first focus the search on the exploration aspect; then it would gradually switch from exploration to exploitation to locate the global optimum. In addition, it is important to realize that these two aspects are mainly achieved at the level of search operators. To design a powerful hybrid algorithm, several questions naturally arise, such as: what kinds of operators should be selected and what criteria are used for selecting search operators?

To address the first question, a diverse selection of operators is preferred because it is likely that a single operator cannot be both effective and efficient for a wide range of applications. Therefore, a combined use of different operators should perform better on a variety of problems. Hence, an "optimized" portfolio of these operators is expected to facilitate search and yield satisfactory outcome. However, the answer to the

---

***GALAXY Method***

*Inputs:* population size ( $N$ ), number of function evaluations ( $NFEs$ )

*Outputs:* Pareto approximation set ( $AS$ ), Pareto approximation front ( $AF$ )

---

***Initialization:***

Generate the initial population of  $N$  individuals randomly in the specified variable domains.

Initialise the quotas\* of six search operators equally such that ( $\sum_{j=1}^6 N_j = N$ ).

***Evaluation:***

Evaluate the objective function values of the initial population (by using hydraulic simulations).

Rank the population using the non-dominated sorting procedure [Deb et al., 2002].

Update the current number of function evaluations (i.e., set  $I = N$ ).

**While**  $I \leq NFEs$

***Selection:***

Choose all the members in the current population for the ***Generation*** step.

***Generation:***

**For**  $J = 1$  to 6

Produce  $N$  candidate solutions from the current population using operator  $J$ .

Select  $N_j$  offspring randomly from candidate solutions and save them to the offspring set.

**End**

Check whether the solutions in the offspring set are within the specified variable ranges.

***Evaluation:***

Evaluate the objective function values of the solutions in the offspring set.

***Replacement:***

Combine the current population and the offspring set as an intermediate population of size  $2N$ .

Implement the duplicates handling strategy.

Rank the intermediate population using the non-dominated sorting procedure.

**If** the number of individuals in the top rank  $\leq N$

Implement the normal replacement via the crowded-comparison operator [Deb et al., 2002].

**Else**

Implement the  $\epsilon$ -replacement strategy.

**End**

Form the next population of size  $N$ .

Update the quotas\* of search operators according to their contributions to the next population.

Update the current number of function evaluations (i.e., set  $I = I + N$ ).

**End**

Set the current population as  $AS$ .

Set the objective function values of the current population as  $AF$ .

---

Note: \*A quota of a search operator refers to the number of offspring it is allowed to produce for the next generation.

**Figure 1.** Pseudocode of GALAXY.

second question is not trivial and normally requires a trial-and-error approach to determine the criteria for selecting operators.

Vrugt and Robinson [2007] proposed an advanced hybrid algorithm, known as AMALGAM, and four search operators were involved including GA, PSO, DE, and Adaptive Metropolis search. These operators themselves have been found to be effective and efficient on a wide range of applications. However, when dealing with discrete combinatorial optimization problems, their capabilities cannot be guaranteed. For example, Wang et al. [2014] found that AMALGAM was occasionally performing worse than NSGA-II because not all operators worked well when dealing with discrete variables. This is, however, expected as solving a discrete combinatorial optimization problem is quite different from solving a continuous, real-valued

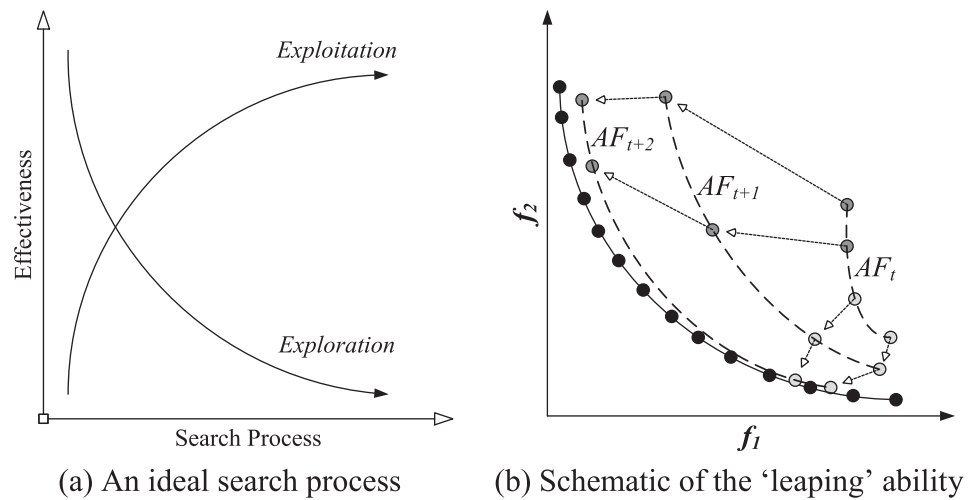


Figure 2. Main concerns of selecting search operators.

problem. In particular, the values of decision variables are restricted to integers, thus the fractional part of decision variable values are omitted during the evaluation of objective functions. In other words, these algorithms have to explore solutions at a limited interval (at least equal to 1) in the search landscape. Therefore, their behaviour is not as ‘smooth’ as that in solving continuous problems, and appears to be “leaping” in both the objective and the decision variable spaces.

Consequently, when developing a hybrid algorithm to discrete problems, it is of great importance to employ search operators that are good at “leaping” in the global or local sense. Figure 2b illustrates this “leaping” concept in more details. The black solid circles denote the targeted solutions in the Pareto-optimal front (PF); whereas the gray solid circles denote the solutions obtained in the Pareto approximation fronts. The light gray circles represent the solutions obtained by the operator with good ‘leaping’ ability in the local sense; in contrast, the dark gray circles are the solutions obtained by the operator with good “leaping” ability in the global sense. The approximation front identified at generation  $t$  is annotated as  $AF_t$ . It can be seen that solutions identified by the operator good at global search (exploration) move rapidly toward the boundary solutions in the PF. While solutions found by the operator good at local search (exploitation) approach slowly to the solutions in the PF. As such, a combined use of search operators which are good at exploration and exploitation is anticipated to drive the hybrid algorithm toward the PF quickly and consistently. To this end, up to six search operators are deployed in GALAXY according to their “leaping” ability in the search space.

There exist many search operators with different features dedicated for solving continuous problems in the literature. Each may require a number of parameters to be set before use. When adapting those to discrete problems, it should be born in mind that if the individual parameters are retained in the hybrid framework they may impose severe parameterization problems. Therefore, to keep the number of parameters to the minimum possible level, only the most effective elements of these operators are identified and implemented in GALAXY. This is achieved by solving a variety of benchmark WDS design problems using each operator alone and monitoring its performance in both visual and statistical ways.

### 3.2.1. Turbulence Factor (TF)

The turbulence factor was previously used in PSO within AMALGAM. It samples a uniformly distributed random vector on the interval between  $[-1,1]$  and perturbs each decision variable (see equation (4)). By applying this operator, each decision variable will be increased or decreased with equal chance. Although the TF is very simple and has no parameter to tune, it is found to be highly efficient thus enabling a superfast global search at the initial stage of optimization.

$$X_{t+1}^i = \lfloor X_t^i + R_T \otimes X_t^i \rfloor \quad (4)$$

where  $X_t^i$  is a vector of decision variables at generation  $t$ .  $R_T$  is a random vector whose each element is uniformly sampled between  $[-1,1]$ .  $\otimes$  is a vector based product.  $\lfloor * \rfloor$  is an operator that rounds down to the nearest integer.

### 3.2.2. Differential Evolution (DE)

DE is a popular stochastic algorithm because of its simplicity yet efficiency [Storn and Price, 1997]. It iteratively improves a candidate solution by using weighted differences between other randomly sampled solution vectors. DE has been proven elsewhere to exhibit rotationally invariant nature, which means that it can cope with the strong interdependencies among decision variables. Equation (5) shows a basic form of DE where three randomly selected solutions are used for illustration. Note that  $K$  and  $F$  were originally uniformly generated numbers between (0.2, 0.6) and (0.6, 1.0), respectively, which encompassed the recommended values (i.e.,  $K = 0.4$  and  $F = 0.8$ ), as reported by Iorio and Li [2004]. Herein,  $K$  and  $F$  are eliminated (made equal to 1) in order to satisfy the integer coding scheme

$$X_{t+1}^i = X_t^i + K(X_t^a - X_t^i) + F(X_t^b - X_t^c) \quad (5)$$

where  $X_t^a$ ,  $X_t^b$ , and  $X_t^c$  are three randomly selected individuals from current population, and they must be different from each other and from  $X_t^i$ .

### 3.2.3. Simulated Binary Crossover for Integers (SBXI) and Uniform Mutation (UM)

In NSGA-II, Simulated Binary Crossover (SBX) and Polynomial Mutation (PM) play a major role in the reproduction of population. They were originally developed for solving continuous problems, and the spread of children from their parents is determined by an exponential function. Distribution of density for this function is controlled by an index for SBX and PM, respectively. However, it was observed in the trial runs that SBX and PM did not work well in the discrete search space. Hence, SBXI and UM are proposed to replace them and fit in the GALAXY method and the integer coding scheme. Equations (6) and (7) show how SBXI and UM are implemented. The mutation rate is equal to the inverse of the number of decision variables ( $1/ND$ ), which is the most recommended value for mutation [Goldberg, 1989]. As a result, SBXI and UM have no parameters to be fine-tuned:

$$\begin{cases} X_{t+1}^a = X_t^a + \text{randi}[0, (X_t^b - X_t^a)], & u \leq 0.5 \\ X_{t+1}^b = X_t^b - \text{randi}[0, (X_t^b - X_t^a)], & u > 0.5 \end{cases} \quad (6)$$

$$X_{t+1}^i = \text{randi}[LB, UB], u \leq 1/ND \quad (7)$$

where  $\text{randi}[a,b]$  is a uniformly distributed random integer between  $[a,b]$ .  $u$  is a uniformly sampled random number between (0,1).  $UB$  and  $LB$  are upper and lower bounds on decision variables.  $X_t^a$  and  $X_t^b$  are parents generated by applying the tournament selection.

### 3.2.4. Gaussian Mutation (GM)

Since the mutation operator alone was observed to show satisfactory performance in the trial runs, GM is also introduced into the GALAXY framework. It differs from UM in the distribution of randomly sampled values in the domain of decision variables. In particular, a discrete Gaussian distribution (also known as normal distribution) is adopted as shown in equation (8). The probability of mutation for each decision variable is equal to  $1/ND$ . In addition,  $\sigma$  is set to a tenth of the range of decision variables because it can prevent a majority of random numbers being generated outside the available pipe sizes.

$$X_{t+1}^i = \left\lfloor \frac{LB+UB}{2} + \sigma \times \text{randn} \otimes X_t \right\rfloor \quad (8)$$

where  $\sigma$  is a scaling factor which is equal to  $\frac{(UB-LB+1)}{10}$ .  $\text{randn}$  is a vector of random numbers drawn from the normal distribution between (0,1) with the same size of  $X_t$ .

### 3.2.5. Dither Creeping (DC)

To facilitate an intensified local search, a new operator, known as the dither creeping mutation [Zheng et al., 2013], is introduced in the GALAXY framework, which combines the creeping mutation [Dandy et al., 1996] and dither mutation strategy [Das et al., 2005] to solve discrete pipe sizing problem. The original DC is characterized by a variant probability of mutation (denoted as  $P_{dcm}$ ) for each individual, which is uniformly sampled from within a small range centered about  $1/ND$ . Moreover, the pipe size is changed to the nearest smaller or larger diameter option, depending on the probability of downward variation (denoted as  $P_d$ ). Zheng et al. [2013] revealed that a GA with DC but without crossover outperformed its counterparts for a range of benchmark design problems.

Note that here the original DC is modified by yielding the probability of dither creeping at the level of gene, rather than at the level of chromosome. This is expected to bring more perturbation to each



---

***Dither Creeping***

$P_{dcm}$  : probability of dither creeping mutation;

$P_{dcm}^{\min}$  and  $P_{dcm}^{\max}$  : lower and upper bounds of probability of dither creeping mutation;

$N$  and  $ND$ : population size and number of decision variable;

$P_d$  : conditional probability of downward variation;

$X_t^i$  : decision variable  $i$  at generation  $t$ ;

$D_{\min}$  and  $D_{\max}$  : lower and upper bounds of diameter option;

*rand*: a uniformly distribution random number sampled between (0,1);

---

**For**  $i=1, 2, \dots, N$

**For**  $j=1, 2, \dots, ND$

$$P_{dcm}^j = P_{dcm}^{\min} + rand^j \times (P_{dcm}^{\max} - P_{dcm}^{\min})$$

**If**  $rand^j < P_{dcm}^j$

**If**  $rand^k \leq P_d$

$$X_{t+1}^i = \max[D_{\min}, X_t^i - 1]$$

**Else**

$$X_{t+1}^i = \min[D_{\max}, X_t^i + 1]$$

**End If**

**End If**

**End For**

**End For**

---

**Figure 3.** Pseudocode of DC.

solution. On the other hand, an interesting feature of DC lies in the fact that the direction of creeping can be simply controlled by  $P_d$ . More specifically, using a smaller  $P_d$  steers the population toward the high cost region (also high in network resilience); whereas using a larger  $P_d$  pushes the population toward the low cost region. This characteristic can be very useful if a particular region of PF is of greater interest, although this results in losing the whole picture of the PF. Consequently,  $P_d$  of the modified DC is uniformly sampled between (0,1) rather than fixed at 0.5 as in the original one, in which case the size of each pipe has an equal chance to be enlarged or reduced. Figure 3 illustrates the flowchart for the modified DC operator.

The reasons for selecting TF, DE, SBXI, UM, GM, and DC as search operators for the GALAXY method are threefold. First, their nature and behaviors are distinct from each other, which makes it possible for each to serve as the main driving force at different search stages. Second, DE and SBXI generate offspring by combining the information passed by other solutions; while the others work on an individual solution independently and perturb the solution at the level of gene. Therefore, the difference in operators in producing solutions may prevent GALAXY from getting trapped at local optima. Third, from the viewpoint of leaping ability, TF, DE, and SBXI are more likely to identify better solutions globally at early generations. In contrast, UM, GM, and DC emphasize local variations, being good at fine-tuning current solutions. As a result, a synergistic effect is anticipated by using a combination of six such operators to ensure a good balance of exploration and exploitation throughout the search.

### 3.3. Strategies of GALAXY

In addition to a careful selection and adaptation of search operators, several strategies are incorporated into the GALAXY method, which also have great impact on its performance. The following subsections explain these strategies in greater details.

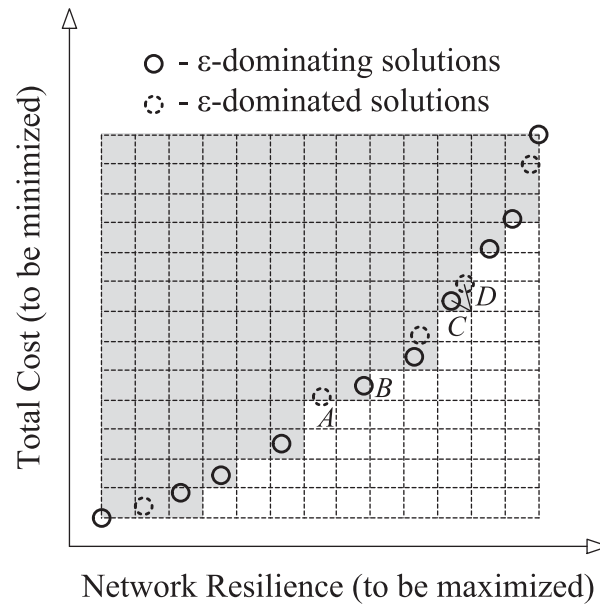


Figure 4. Concept of  $\varepsilon$ -dominated sorting.

next population ( $P_{t+1}$ ). If there are still free spaces left, some  $\varepsilon$ -dominated solutions are also selected which have smaller distances to the ideal global optima.

Figure 4 depicts the  $\varepsilon$ -dominance concept in more details. First of all, the boundary solutions in the first front are identified and used to construct the 2-D boxes with  $\varepsilon_i$  as the side length in the  $i$ th objective. The value of  $\varepsilon_i$  is calculated by equation (9). Note that the boundary solutions must be included in the next population. Therefore, the number of candidate solutions to be selected is  $N-2$ . In Figure 4, the solid circles represent  $\varepsilon$ -dominating solutions (e.g., B and C); while the dashed circles represent  $\varepsilon$ -dominated solutions (e.g., A and D). The gray area indicates the region dominated by  $\varepsilon$ -dominating solutions. Since the number of nondominated solutions in the 2-D boxes is larger than  $N-2$ , they are first compared using the  $\varepsilon$ -domination concept. The  $\varepsilon$ -dominating solutions are then included in the next population. If there are remaining spaces, the  $\varepsilon$ -dominated solutions with smaller Euclidean distances to the global optimum (i.e., the lower-right corner in Figure 4) are also chosen. As a result, this hybrid replacement strategy ensures that the nondominated solutions in the first front, which are near the global optima, are preserved, and a good diversity of these solutions is also achieved

$$\varepsilon_i = \frac{f_i^{\max} - f_i^{\min}}{N-2} \quad (9)$$

where  $\varepsilon_i$  is the side length of each box in the  $i$ th objective.  $f_i^{\max}$  and  $f_i^{\min}$  are maximum and minimum values of the  $i$ th objective, respectively.  $N$  is population size.

This hybrid replacement strategy is used to preserve the nondominated solutions which may have reached the PF. In other words, the  $\varepsilon$ -dominance comparison is employed to overcome the shortcoming of the crowding distance based sorting, which aims to achieve a relatively uniform spread at the price of losing the solutions on the PF. Actually, for a discrete combinatorial optimization problem, it is rare that the PF is uniformly distributed. Therefore, the crowding distance based sorting, which was proposed for solving continuous problems, may become problematic.

On the other hand, the  $\varepsilon$ -replacement procedure implemented in GALAXY differs from the traditional  $\varepsilon$ -dominance comparison in two aspects. First, it does not require any user-specified  $\varepsilon$  precision for each objective. Instead, the  $\varepsilon$  precisions are determined internally according to the extent of solutions in the first front. Second, it allows some  $\varepsilon$ -dominated solutions to be selected, rather than being completely removed, provided that they are close to the global optima.

### 3.3.1. Hybrid Replacement Strategy

In MOEAs, the *Replacement* procedure plays a crucial role in steering the population toward the PF. As shown in Figure 1, a hybrid replacement strategy is implemented in the GALAXY framework. Specifically, after the creation of an intermediate population ( $I_t$ ) using the unique solutions in the current population ( $P_t$ ) and the offspring ( $O_t$ ), the individuals are ranked using the fast nondominated sorting [Deb et al., 2002]. If the number of individuals in the top rank is no greater than the population size, the normal replacement is carried out as in NSGA-II. However, if the number of individuals in the top rank exceeds the population size, the  $\varepsilon$ -replacement is carried out instead, in which the nondominated solutions in the first front are sorted once again based on the  $\varepsilon$ -dominance concept [Deb et al., 2005]. Therefore, the  $\varepsilon$ -nondominated solutions are copied into the



### 3.3.2. Genetically Adaptive Strategy

GALAXY adopts the genetically adaptive strategy first proposed in AMALGAM to take full advantages of six search operators. First, the offspring pool  $Q_0$  of size  $N$  is created from the initial population  $P_0$  using six search operators simultaneously, with each operator contributing nearly the same number of individuals (i.e.,  $N/6$ ). Next, a combination of the parents ( $P_0$ ) and the offspring ( $Q_0$ ), namely  $R_0$  (size  $2N$ ), is produced and ranked via the fast nondominated sorting procedure [Deb et al., 2002]. A number of individuals,  $N$ , from  $R_0$  are then selected based on the  $\epsilon$ -replacement strategy, forming the population for the next generation. Finally, the number of individuals that is allowed to produce by each search operator is determined according to the reproductive rate (ratio of the children alive to the children created) in the previous generation (see equation (10)).

When a search operator fails to contribute even a single individual in the current population, the one-child policy is applied in the next round to release maximum reproductive chances. Specifically, it borrows one opportunity from the topmost operator in the previous round. If two of six operators fail, each of them borrows one opportunity from the two topmost operators, and so on. Therefore, the most successful operator is always favored by getting the highest number of offspring in the reproduction process, and no operator is completely discarded even though it exhibits the worst performance.

$$N_t^i = N \times \frac{(P_t^i / N_{t-1}^i)}{\sum_{i=1}^{ns} (P_t^i / N_{t-1}^i)} \quad (10)$$

where  $N_t^i$  is the number of offspring generated by operator  $i$  at generation  $t$ .  $P_t^i$  is number of offspring contributed by operator  $i$  at generation  $t$ .  $ns$  is number of search operators.

### 3.3.3. Global Information Sharing Strategy

This strategy was first mentioned by *Vrugt and Robinson* [2007], but was not explained in detail. The essential idea is to share the current population among each search operator in the *Generation* step. In particular, each operator generates the same number of candidate solutions, which is equal to the population size, rather than just the number of individuals it is allowed to produce. This permits the not-so-good operators to benefit from the high-quality solutions contributed by other operators. Then, a number of offspring, which is determined by the genetically adaptive strategy, are chosen at random from the candidate solutions produced by each operator. This partially explains why the global information sharing strategy differs from a random search, and the preservation of elitism is mainly achieved at the *Replacement* step. The global information sharing strategy allows the search operators with poor performance in previous generations to recover subsequently, thus increasing their opportunities to contribute better solutions in the later run. Note that this strategy can be only realized in the generational framework.

### 3.3.4. Duplicates Handling Strategy

It is worth noting that there is a much higher possibility that duplicate solutions will be generated in the context of discrete combinatorial problems than in real-parameter continuous situations. Consequently, the possibility of discovering more and/or better solutions is wasted because some solutions are repeated in the population. Furthermore, the actual number of nondominated solutions is less likely to reach the population size at the end of the search. Some researchers [Chaiyaratana et al., 2007; Črepinšek et al., 2013] have suggested discarding duplicate individuals to improve the overall performance (mainly for diversity) of EAs. Hence, a “unique solution strategy” is applied in GALAXY after the combination of the offspring and the current population as shown in Figure 1. This strategy is aimed at checking and removing duplicate solutions that exist in the population thereby accommodating more individuals with potentially better fitness. This strategy is also believed to prevent premature convergence of the population.

### 3.3.5. Constraint Handling Strategy

GALAXY applies a penalty-free, constrained-domination strategy [Deb et al., 2002] to deal with infeasible solutions due to its simplicity and effectiveness. It eliminates the use of penalty function and works as follows. Assuming that two solutions ( $A$  and  $B$ ) are randomly chosen from the population for comparison, solution  $A$  is selected if: (i)  $A$  has a less constraint violation than  $B$ ; or (ii)  $A$  dominates  $B$  when both are feasible solutions. Otherwise, one of them is chosen at random.

## 4. Baseline MOEAs

The baseline MOEAs used in this work are NSGA-II,  $\epsilon$ -MOEA [Deb et al., 2005] and Borg [Hadka and Reed, 2013]. NSGA-II is arguably the most popular MOEA and is regarded as an “industry standard” algorithm,

which has been successfully applied to a variety of fields. NSGA-II features a fast nondominated sorting approach, implicit elitist selection method based on Pareto dominance rank and a secondary selection method based on crowding distance, which significantly improves its performance on difficult multiobjective problems. Moreover, it provides a constraint-handling technique to deal with constrained problems efficiently and supports both binary and real coding representations.

Unlike NSGA-II,  $\epsilon$ -MOEA is a steady state MOEA in which only one solution is generated per iteration. It incorporates the concept of  $\epsilon$ -dominance [Laumanns *et al.*, 2002], being able to preserve a good representation of the PF in terms of convergence and diversity. At the beginning, a population is initialized randomly and the nondominated solutions are retained in an archive. Next, a solution is created via SBX and PM using two parents each of which is selected from the population and the archive. Then, Pareto dominance and  $\epsilon$ -dominance are used to check this solution for acceptance or rejection by the population and the archive, respectively. The preceding procedure is repeated until a stopping criterion is met.

Borg, using  $\epsilon$ -MOEA as its outer framework, employs six search operators simultaneously and implements several advanced strategies concurrently, including  $\epsilon$ -dominance,  $\epsilon$ -progress (a measure of convergence speed), adaptive population sizing, and randomized restart. The methodology proves to be a flexible and robust hybrid optimizer by covering a large area of high-performing parameterizations [Hadka and Reed, 2012]. Due to the space limit, readers are referred to [Hadka and Reed, 2013] for greater details.

The reasons for choosing these three MOEAs are as follows. NSGA-II has been widely used as a benchmark MOEA in water engineering [Farmani *et al.*, 2005; Kollat and Reed, 2006; Raad *et al.*, 2009], and it serves as the prototype of GALAXY in terms of the generational algorithmic framework.  $\epsilon$ -MOEA was introduced after NSGA-II featuring a steady state algorithmic framework and the  $\epsilon$ -dominance concept, which was successful in finding well-converged and well-distributed solutions for a variety of test problems [Deb *et al.*, 2005]. Borg, as a hybrid optimizer, further enhances  $\epsilon$ -MOEA by using the multiple recombination operators as well as a suite of sophisticated runtime strategies. In contrast, GALAXY uses both the nondominated sorting and the  $\epsilon$ -dominance based sorting in the *Replacement* step, and differs from baseline MOEAs, especially from Borg, mainly at the implementations of the multioperator search (tailored for the WDS design) and hybrid replacement strategies. Therefore, the comparison between GALAXY and three selected baseline MOEAs can help verify the effectiveness and efficiency of this new hybrid optimizer. In addition, NSGA-II,  $\epsilon$ -MOEA and Borg implement the real coding scheme and the decision variables are rounded down to the nearest integers when being evaluated in the objective function. Note that this is different from the approach implemented in Creaco *et al.* [2010], in which the search operators were modified to suit the problem with integer variables.

## 5. Benchmark Problems

The performance of GALAXY is tested on a representative subset from the library of benchmark problems collected from the literature [Wang *et al.*, 2015]. In particular, the BakRyan network [Lee and Lee, 2001], the Hanoi network [Fujiwara and Khang, 1990], the Pescara and the Modena networks [Bragalli *et al.*, 2008], and the Balerma network [Reca and Martínez, 2006], are presented to illustrate the capability of GALAXY on a wide scale of WDS design problems. A summary of these benchmark problems is given in Table 1. More information about these benchmark problems, readers are referred to <http://tinyurl.com/cwsbenchmarks/>.

**Table 1.** Summary of the Formulations of Benchmark Problems

Scale	Problem	No. of		Search Space	Design Criteria		
		DVs	Options per DV		$P_{\min}$	$P_{\max}$	$V_{\max}$
Small	BakRyan	9	11	$2.36 \times 10^9$	Yes	No	No
Medium	Hanoi	34	6	$2.87 \times 10^{26}$	Yes	No	No
Intermediate	Pescara	99	13	$1.91 \times 10^{110}$	Yes	Yes	Yes
Large-1	Modena	317	13	$1.32 \times 10^{253}$	Yes	Yes	Yes
Large-2	Balerma	454	10	$1.0 \times 10^{455}$	Yes	No	No

<sup>a</sup>Note: DV is decision variables.  $P_{\min}$  is minimum pressure head requirement.  $P_{\max}$  is maximum pressure head requirement.  $V_{\max}$  is maximum flow velocity.

## 6. Experimental Setup

### 6.1. Algorithmic Setup

It is well established that parameter settings play an important role in MOEAs. These parameters can be divided into two groups, i.e., general and specific ones. General parameters include those that are common for various algorithms, like  $N$  and  $NFEs$ ; whereas the specific parameters involve those only employed by a particular algorithm.

For the settings of general parameters, readers are referred to the computational budget subsection. These settings are adjusted according to the complexity of different WDS design problems, and kept the same among four algorithms. The individual parameters of each algorithm are selected based on the results of sensitivity analyses, using the DC operator and NSGA-II, respectively, and are specified as follows. Borg follows the recommended settings provided in Wang *et al.* [2015].

GALAXY actually needs to specify only one extra parameter, which is the range of  $P_{dcm}$  used in the DC operator (see Figure 3). The range was considered from 0.1 to 0.9 at a step of 0.1. It was found that the performance of DC on various cases was not sensitive to the setting of this range, as the maximum relative deviation among four indicators (see section 6.2) was less than 0.9% on average. Therefore, the range of  $P_{dcm}$  is set to 0.7 based on an overall best performance.

The distribution indices of SBX and PM control the spread of offspring from their parents. Lower values of the distribution index increase the probability of generating solutions far away from the parents [Deb *et al.*, 2002]. As such, different distribution indices, i.e., 1, 5, 10, 15, 20, were tested for SBX and PM separately to confirm the best option from 25 possible combinations. It was found that keeping the distribution indices as 1 for both SBX and PM generally returned the best results; consequently, this setting is adopted for both NSGA-II and  $\epsilon$ -MOEA.

The probabilities of various mutation operators and SBX follow the recommended values in the literature, which are  $1/ND$  and 0.9, respectively [Deb *et al.*, 2002; Goldberg, 1989].

### 6.2. Search Performance Indicators

To facilitate a comprehensive comparison between GALAXY and baseline MOEAs, both quantitative performance indicators and graphical approaches are used in a complementary manner. The former provides an explicit way to assess the quality of solutions, while the latter illustrates the performance visually and intuitively in the objective space. Recall that there are mainly two distinct goals in multiobjective optimization: (1) to approximate the PF as close as possible (convergence); and (2) to maintain a good spread of solutions in the objective space (diversity). Ideally, a good algorithm is expected to discover the solutions lying on the PF and well distributed over the entire objective space.

Four metrics, that is generational distance [Veldhuizen, 1999], unary hypervolume [Zitzler and Thiele, 1999], unary additive  $\epsilon$ -indicator [Zitzler *et al.*, 2003], and  $\epsilon$ -performance [Kollat and Reed, 2005], are employed to evaluate the convergence and diversity of solutions obtained by different MOEAs. Generational distance (denoted as  $I_{GD}$ ) measures an average distance of the nondominated solutions in the Pareto approximation front (denoted as  $AF$ ) from those in the Pareto reference front (denoted as  $RF$ ). Unary hypervolume (denoted as  $I_{HV}$ ) calculates the ratio of the volume of hypercube (in the objective space) dominated by solutions in the  $AF$  to that by solutions in the  $RF$ . Unary additive  $\epsilon$ -indicator (denoted as  $I_{\epsilon+}$ ) seeks the minimum distance that the  $AF$  must be translated in order to completely dominate the entire  $RF$ .  $\epsilon$ -performance (denoted as  $I_{EP}$ ) accounts for the proportion of solutions that are discovered within a user-specified  $\epsilon$ -distance (see Table 2) from the  $RF$ . These  $\epsilon$  precision values are determined by a trial-and-error approach, which are considered sufficient to differentiate the improvements in the objective space. These values are also adopted when  $\epsilon$ -MOEA is executed. The main features of these indicators are given in Table 3. Note that the qualified solutions (in the  $AF$ ) considered for calculating the aforementioned metrics must be feasible and belong to the first rank. In other words, the inferior and infeasible solutions are omitted from the evaluation.

The prerequisite of computing aforementioned metrics is a known  $RF$ . To this end, Wang *et al.* [2015] applied five state-of-the-art MOEAs to find the best-known approximation of the true Pareto front for the

**Table 2.**  $\epsilon$  Precision Values

Benchmark Problem	First Objective: Cost Minimization (in Millions Unit)			Second Objective: $I_n$ Maximization (-)		
	$Cost^{best}$	$Cost^{worst}$	$\epsilon^{cost}$	$I_n^{worst}$	$I_n^{best}$	$\epsilon^{I_n}$
BakRyan	0.378	1.389	0.001	0	1	0.001
Hanoi	1.803	10.970	0.1	0	1	0.001
Pescara	1.346	19.004	0.01	0	1	0.001
Modena	1.989	28.083	0.01	0	1	0.001
Balerma	0.724	21.642	0.01	0	1	0.001

<sup>a</sup>Note: The  $Cost^{best}$  and  $Cost^{worst}$  for each case are obtained by setting the decision variables (i.e., pipe diameters) to the minimum and maximum commercially available sizes, respectively.  $I_n^{worst}$  and  $I_n^{best}$  are extreme cases, which are determined by the definition of  $I_n$  (see equation (2)).

benchmark problems used in this paper, given a massive computational budget. Therefore, these best-known solutions to each problem are used to construct the RF.

In addition to comparing the ultimate performance of GALAXY with baseline MOEAs using four metrics, their dynamic performances are also taken into consideration to further demonstrate some interesting and potentially important information during the search history. *Zitzler et al.* [2002] emphasized the importance of tracing the performance of an algorithm over time. This would help to identify the differences on convergence speed and premature convergence (or stagnation). The abovementioned indicators have already been used to evaluate the dynamic performance in some comparative studies [*Fu et al.*, 2012; *Kollat and Reed*, 2006].

The graphical approach is based on the empirical attainment function (EAF) tool [*López-Ibáñez et al.*, 2010], which is capable of estimating the attainment function [*Fonseca et al.*, 2001] using the data collected from several independent runs. It demonstrates the boundaries of attainment surface detected by an algorithm and highlights the performance differences between two competing algorithms. This feature is very attractive when comparing the performance of MOEAs as it properly addresses their stochastic nature.

In each EAF plot (see Figure 6), the performances of two algorithms are plotted side by side and four types of information are demonstrated: (1) best attainment surface obtained by both algorithms (lower solid lines); (2) worst attainment surface obtained by both algorithms (upper solid lines); (3) median attainment surface of each algorithm (dashed lines); and (4) differences of EAFs between two algorithms. Such differences are encoded using discrete levels of gray color: the darker an area is plotted, the larger the difference exists there.

### 6.3. Computational Budget

In order to ensure a fair comparison between GALAXY and baseline MOEAs, a sufficient computational budget in terms of the NFEs is determined by applying NSGA-II to benchmark problems extensively. Its dynamic performance using four metrics was monitored, thus picking a suitable budget to confirm the convergence of NSGA-II. Specifically, a budget was determined so that the improvement of the NSGA-II's convergence was less than a threshold (0.5% herein), implying that more computation was not necessary. Therefore, 25,000 and 50,000 NFEs are used for the BakRyan and the Hanoi problems and 100,000 NFEs for the Pescara problem with a population size of 100. For the Modena and Balerma problems, such a budget is much more difficult to identify due to the huge search space. As a result, a total number of 400,000 NFEs with a doubled population size (i.e., 200) is applied instead. In addition, for each problem 30 independent runs are implemented using different random seeds.

**Table 3.** Main Features of the Selected Performance Indicators

Metric	Type	Range	Normalization	Ideal Value
$I_{GD}$	Convergence	$[0, \infty)$	$\max(1 - I_{GD}, 0)$	1
$I_{HV}$	Convergence and diversity	$[0, 1]$	N/A	1
$I_{c+}$	Convergence and consistency	$[0, \infty)$	$\max(1 - I_{c+}, 0)$	1
$I_{EP}$	Convergence	$[0, 1]$	N/A	1

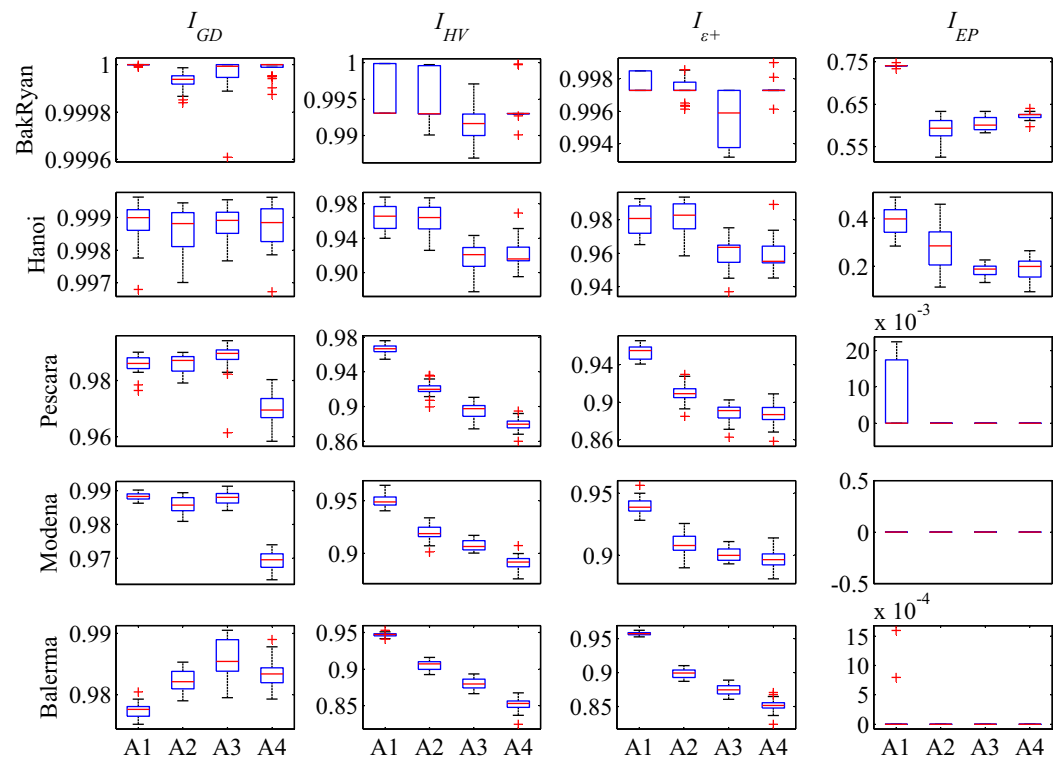
It is also worth noting that there are a number of possible criteria for comparing different algorithms. For example, using an identical/similar CPU time is another alternative. However, the GALAXY algorithm proposed here is developed in the *Matlab* environment, thus being much slower compared to baseline MOEAs that are coded in C

language. Despite of this, it is important to point out that the hydraulic simulations nested inside the optimization loop account for the majority of CPU time. By using the MEX-file technique supported in *Matlab*, this time-consuming part can be implemented much more efficiently. Thus, the real runtime needed by GALAXY has been substantially shortened and is comparable with baseline MOEAs (see supporting information Table S1). On the other hand, the main computational overhead within GALAXY stems from the Pareto dominance sorting, which has a theoretical computational complexity of  $O(MN^2)$ , where  $M$  is the number of objectives and  $N$  is the size of population. This is the same as for NSGA-II, implying that GALAXY can be as fast as NSGA-II if developed in C language. As such, this work considers the identical NFEs rather than the absolute CPU time as the criterion for comparison purposes.

### 7. Results and Discussion

The statistical results obtained by GALAXY and three comparison MOEAs, after 30 independent runs given different computational budgets for five design problems, are shown in the Box and Whisker plots in Figure 5. In each plot, the central line represents the median, and the edges of each box are the 25th and 75th percentiles. The most extreme data are linked by whiskers, but those considered outliers are plotted individually with plus symbols. Recall that the performance indicators have been normalized to fall between 0 and 1 with a larger value indicating better achievement.

It is evident that GALAXY outperformed NSGA-II,  $\epsilon$ -MOEA, and Borg for all five cases according to  $I_{HV}$  and  $I_{EP}$ , which confirm its superior capability in terms of convergence and diversity. The advantages of GALAXY turn out to be more significant as the problem becomes more complex. For the BakRyan problem, it obtained nearly 25% better  $I_{EP}$  compared to the baseline MOEAs; while for the Hanoi problem the differences according to  $I_{EP}$  rise to 39%, 111% and 99% beyond NSGA-II,  $\epsilon$ -MOEA and Borg, respectively. GALAXY was the only algorithm that successfully identified the solutions close enough to the Pareto RF of the Pescara problem (as indicated by  $I_{EP}$ ), given just one-tenth of the computational budget (i.e., 1 million NFEs) used to obtain the corresponding best-known PF [Wang et al., 2015]. Consequently, there is a lack of results in terms of  $I_{EP}$  for the baseline MOEAs for Pescara (plotted as short lines instead). Similarly, no box plot for Modena and



**Figure 5.** Box and Whisker plots of statistical results evaluated by different indicators for each design problem (A1-GALAXY; A2-NSGA-II; A3- $\epsilon$ -MOEA; A4-Borg).

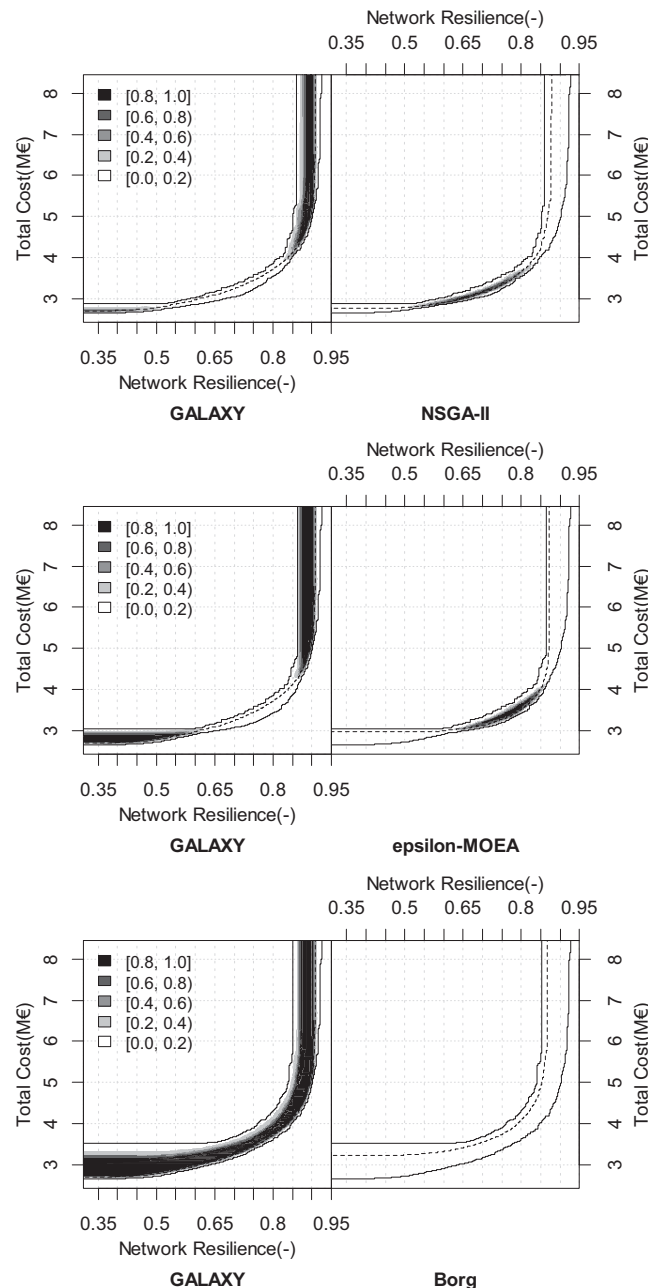


Figure 6. EAF plots of GALAXY versus baseline MOEAs for the Modena problem.

by GALAXY around the “knee point” of the best-known PF. A probable explanation for this phenomenon is that the combination of SBX and PM tends to drive the population toward this particular area at the price of losing the whole picture of the PF. Remember that the goal of multiobjective optimization is twofold: (1) to approximate the PF as close as possible; and (2) to extend the distribution as wide as possible. Therefore, without giving any further information and guidance for the final decision making, a set of better converged and wider spread solutions are generally preferred.

It is also worth noting that NSGA-II,  $\epsilon$ -MOEA and Borg were actually less efficient in exploring the search space, because the current best-known PF of the Modena problem ranges from 2.54 to 24.82 million euros [Wang et al., 2015]. However, the baseline MOEAs reported the solutions between 2.98 and 5.18 million euros on average, which covers less than 10% of the cost spectrum. In contrast, GALAXY captured nearly

Balerna is provided because all the algorithms failed to capture the solutions near the best-known PFs within the specified precision. Surprisingly, GALAXY successfully identified some boundary solutions (shown as outliers in the lower-right subgraph of Figure 5) for the Balerna problem, which has a substantially larger search space compared to that of Modena. In addition, GALAXY clearly outperformed the baseline MOEAs for all design problems except for Hanoi according to  $I_{\epsilon+}$ , in which NSGA-II got a slightly better median (about 0.2%). Although there is a reverse trend in the lower-left subgraph, all four algorithms achieved quite similar levels of  $I_{GD}$  with the maximum relative difference of the medians below 0.8%. This is because baseline MOEAs only converged better to the region with the network resilience less than 0.85 (see supporting information Figure S9). In contrast, GALAXY was able to converge sufficiently along the entire best-known PF, effectively covered the region with the network resilience greater than 0.85.

The comparisons of the four algorithms for solving the Modena problem, one of representative large problems (see Table 1), are next analyzed in detail. It is obvious from Figure 6 that GALAXY was capable of discovering boundary solutions both in the region of low cost and high network resilience compared with the baseline MOEAs. This can be attributed to the usage of some operators emphasizing efficiency (e.g., TF, SBXI, and DC) that are good at global and local search. NSGA-II and  $\epsilon$ -MOEA obtained a set of more converged solutions than those



15% of the cost spectrum (from 2.69 to 5.89 million euros) including both lower and higher cost regions. This means many more alternatives can be presented to decision makers.

From the dynamic performance perspective, GALAXY was superior to other MOEAs in terms of  $I_{EP}$ , especially from the middle to the end of the search history (as shown in supporting information Figures S3 and S6). However, the other indicators are not shown in this way because the differences in performance are not easily identified. In addition, the variations in terms of the usage probability of each search operator within GALAXY and Borg are traced and demonstrated in Figure 7. The blue line in the center represents the mean and the gray range denotes the envelope of variations.

In GALAXY, three local search operators, namely UM, GM, and DC, turned out to be the dominating factors which steered the optimization progress, with the overall usage probabilities of about 25%, 22%, and 41% on average, respectively. The other operators, which are good at global search, e.g., TF, DE, SBXI, controlled the generation of new offspring mainly at the very beginning of the evolutionary process, and took effect occasionally until the end of search. In particular, the DE operator seemed to be inefficient for dealing with the problem of high dimensionality (317 decision variables), and this may be attributed to low selection of parents when it was originally implemented. In contrast, by using the tournament selection of parents, the

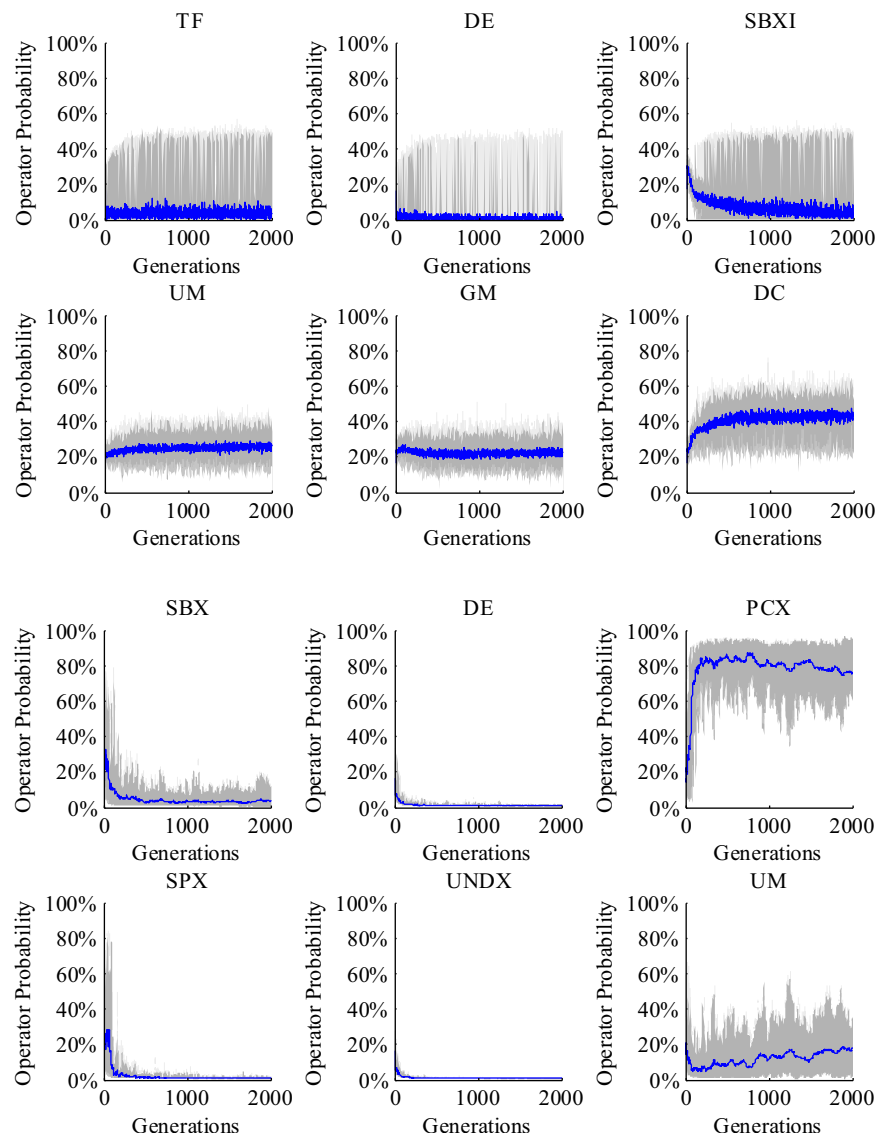


Figure 7. (top) Variations of search operators within GALAXY and (bottom) Borg for the Modena problem.

SBXI operator was frequently employed (more than 10%) in the first one tenth of generations; however, after that it became less efficient compared to other local search operators and its usage probability declined continuously till the end. Moreover, there is a clear trend that GALAXY shifted the computational resources from operators good at exploration (e.g., TF, DE, and SBXI) to those good at exploitation gradually, switching the search directions from global to local automatically (no parameters to be tuned). This pattern of behavior is anticipated when developing the GALAXY method, and is verified in solving the discrete WDS design problems.

By contrast, the PCX operator in Borg consistently dominated the other five operators throughout the optimization. Its usage probability jumped sharply to over 80% after about 200 generations and remained stable above 70%. On the contrary, the SBX, DE, SPX, and UNDX operators were suppressed till the end. Similar to its counterpart in GALAXY, the UM operator consistently accounted for over 10% of the usage probability from the middle to the end of search history.

Again, it is worth emphasizing that GALAXY is substantially different from Borg, although they both incorporate six operators. The main differences are threefold: (i) GALAXY is based on the generational algorithmic framework as NSGA-II, while Borg follows the steady state model proposed in  $\varepsilon$ -MOEA; (ii) GALAXY combines the fast nondominated sorting and the  $\varepsilon$ -dominance based sorting in the *Replacement* step, which ensures a higher level of elitism preservation (as indicated by  $I_{EP}$ ). In contrast, Borg only uses the  $\varepsilon$ -dominance for acceptance of solutions in the archive; (iii) GALAXY employs six search operators which are selected based on their leaping ability in the objective space of the WDS design problems; consequently, the behavior of these operators are quite similar across different design problems. However, the leading operators in Borg change from case to case, and seem to be unpredicted. More importantly, GALAXY tailors the operators to suit the discrete problems by eliminating a number of setup parameters, thus being robust on a wide range of cases. Borg, on the contrary, has a total of nine individual parameters, which obviously increases the complexity of parameterization when applied to different problems.

Results for other cases are provided in supporting information, including the comparisons of EAF plots and the dynamic variations of search operators within two hybrid optimizer. Additionally, the dynamic performances in terms of  $I_{EP}$  for the BakRyan and the Hanoi problems are also supplied to illustrate the differences among the four algorithms.

## 8. Conclusions

This paper develops a new hybrid MOEA, termed GALAXY, for solving the multiobjective WDS design problems. It incorporates six search operators and features several important strategies for dealing with discrete combinatorial optimization problems. The performance of GALAXY has been tested on five representative benchmark design problems with increasing complexity in terms of search space size, and is compared to NSGA-II and  $\varepsilon$ -MOEA, as well as Borg, a very efficient hybrid optimizer recently reported, via four quantitative metrics and the EAF tool. The aforementioned indicators are recorded as the optimizations progress which enables a fair comparison from diversified perspectives.

The results reveal that GALAXY can outperform the baseline MOEAs by finding better converged and distributed solutions especially near the extremities of each objective. For all five cases, GALAXY is superior to NSGA-II,  $\varepsilon$ -MOEA, and Borg evidently and consistently in terms of  $I_{HV}$  and  $I_{EP}$ , as verified by identifying overwhelmingly better  $I_{EP}$  than its competitors for the Hanoi and Pescara problems. This indicates a much more balanced capability between exploration and exploitation without introducing parameters to control this in the search process. In addition, it achieves better or very similar  $I_{e+}$  and  $I_{GD}$  metrics compared to the other algorithms. More importantly, GALAXY's advantages over traditional methods become more highlighted when solving more complex WDS design problems with higher dimensionalities. In addition, it is worth emphasizing that GALAXY is a dedicated optimizer for solving combinatorial problems with discrete variables.

The success of GALAXY is mainly credited to the implementation of the multioperator search, which is tailored for the WDS design problems, and the hybrid replacement strategies. In particular, TF, DC, and SBXI guide the search at the early stage due to good leaping abilities in the global sense; in contrast, UM, GM, and DC strengthen the local search by fine-tuning the solutions found in the near-optimal region. The

hybrid replacement strategy combines the power of Pareto-dominance and  $\varepsilon$ -dominance concepts, thus maintaining a good balance between exploration and exploitation and preventing global optima from being lost. In addition, a subtle modification made to the original genetically adaptive strategy adjusts the best portfolio of six operators and ensures a consistently robust search capability. The global information sharing increases the probability of not-so-good operators to contribute high-quality solutions subsequently. Checking and removing duplicates periodically helps to further improve the convergence and diversity of final solutions.

The practical value of GALAXY lies in the fact that it alleviates the parameterization problem of MOEAs to great extent. The effective ingredients of each operator are identified and tailored to suit the discrete nature of WDS design problems. As a result, GALAXY actually needs to specify only two general parameters (i.e., population size and NFEs) before execution, thus being an easy-to-use and efficient tool for researchers and practitioners in the water community. It is also worth highlighting that GALAXY is not just a hybrid MOEA but a general framework to facilitate other search operators, which provides a better chance to create strengthened MOEAs later.

There is still much future work to be investigated based on the GALAXY method. First, it is important to make in-depth comparisons with other state-of-the-art hybrid methods like AMALGAM and DDMO [Zheng and Zecchin, 2014] for solving WDS design problems, and using newly developed performance metrics [Zheng et al., 2016]. This will help understand why and at what extent the hybridization and/or other associated strategies affect the overall performance of these hybrid algorithms. Second, some recent publications are recommending using the many-objective (more than three) formulations [Fu et al., 2013; Kasprzyk et al., 2012]. The increase in the number of objectives is envisaged to further complicate the search space, thus making the task even harder. Therefore, a systematic study on this aspect using GALAXY and other MOEAs is anticipated to uncover new findings. Third, the underlying reasons why some global search operators (e.g., TF and DE) are inefficient after the early stage, especially for solving larger WDS design problems, need to be further investigated. This can be useful in further improving the current version of GALAXY by allowing longer and more efficient global search. Last but not least, GALAXY, as a hybrid framework, can be used to test other existing search operators and create potentially stronger ones for large and complex WDS design problems in the real world.

#### Acknowledgments

The first author would like to appreciate the financial support given by both the *University of Exeter* and the *China Scholarship Council* (CSC) toward the PhD research. We also appreciate the three anonymous reviewers, who help improve the quality of this paper substantially. The source code of the latest versions of NSGA-II and  $\varepsilon$ -MOEA can be downloaded from the official website of *Kanpur Genetic Algorithms Laboratory* via <http://www.iitk.ac.in/kangal/codes.shtml>. The description of each benchmark problem used in this paper, including the input file of EPANET and the associated best-known Pareto front, can be accessed from the following link to the *Centre for Water Systems* (<http://tinyurl.com/cwsbenchmarks/>). GALAXY can be accessed via <http://tinyurl.com/cws-galaxy>.

#### References

- Alperovits, E., and U. Shamir (1977), Design of optimal water distribution systems, *Water Resour. Res.*, 13(6), 885–900, doi:10.1029/WR013i006p00885.
- Asadzadeh, M., and B. Tolson (2012), Hybrid Pareto archived dynamically dimensioned search for multi-objective combinatorial optimization: Application to water distribution network design, *J. Hydroinform.*, 14(1), 192–205, doi:10.2166/hydro.2011.098.
- Blum, C., and A. Roli (2003), Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Comput. Surv.*, 35(3), 268–308, doi:10.1145/937503.937505.
- Bragalli, C., C. D'Ambrosio, J. Lee, A. Lodi, and P. Toth (2008), *Water Network Design by MINLP*, Res. Rep. RC24495, IBM Res., Yorktown Heights, New York.
- Chaiyaratana, N., T. Piroonratana, and N. Sangkawelert (2007), Effects of diversity control in single-objective and multi-objective genetic algorithms, *J. Heuristics*, 13(1), 1–34, doi:10.1007/s10732-006-9003-1.
- Creaco E., M. Franchini, and S. Alvisi (2010), Optimal placement of isolation valves in water distribution systems based on valve cost and weighted average demand shortfall, *Water Resour. Manage.*, 24(15), 4317–4338, doi:10.1007/s11269-010-9661-5.
- Črepinšek, M., S.-H. Liu, and M. Mernik (2013), Exploration and exploitation in evolutionary algorithms: A survey, *ACM Comput. Surv.*, 45(3), 35, doi:10.1145/2480741.2480752.
- Dandy, G. C., A. R. Simpson, and L. J. Murphy (1996), An improved genetic algorithm for pipe network optimization, *Water Resour. Res.*, 32(2), 449–458, doi:10.1029/95WR02917.
- Das, S., A. Konar, and U. K. Chakraborty (2005), Two improved differential evolution schemes for faster global search, paper presented at the 7th annual conference on Genetic and evolutionary computation, ACM New York, Washington, D. C., 25–29 June.
- Deb, K., M. Mohan, and S. Mishra (2005), Evaluating the  $\varepsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions, *Evol. Comput.*, 13(4), 501–525, doi:10.1162/106365605774666895.
- Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE T. Evolut. Comput.*, 6(2), 182–197, doi:10.1109/4235.996017.
- Engelhardt, M. O., P. J. Skipworth, D. A. Savic, A. J. Saul, and G. A. Walters (2000), Rehabilitation strategies for water distribution networks: A literature review with a UK perspective, *Urban Water*, 2(2), 153–170, doi:10.1016/S1462-0758(00)00053-4.
- Farmani, R., D. A. Savic, and G. A. Walters (2005), Evolutionary multi-objective optimization in water distribution network design, *Eng. Optimiz.*, 37(2), 167–183, doi:10.1080/03052150512331303436.
- Fonseca, V. G. d., C. M. Fonseca, and A. Hall. (2001), Inferential performance assessment of stochastic optimizers and the attainment function, paper presented at the 1st international conference on evolutionary multi-criterion optimization, Springer, Zurich, Switzerland, 7–9 March.
- Fu, G., Z. Kapelan, and P. Reed (2012), Reducing the complexity of multiobjective water distribution system optimization through global sensitivity analysis, *J. Water Resour. Plann. Manage.*, 138(3), 196–207, doi:10.1061/(ASCE)WR.1943-5452.0000171.

- Fu, G., Z. Kapelan, J. Kasprzyk, and P. Reed (2013), Optimal design of water distribution systems using many-objective visual analytics, *J. Water Resour. Plann. Manage.*, 139(6), 624–633, doi:10.1061/(ASCE)WR.1943-5452.0000311.
- Fujiwara, O., and D. Khang (1990), A two-phase decomposition method for optimal design of looped water distribution networks, *Water Resour. Res.*, 26(4), 539–549, doi:10.1029/WR026i004p00539.
- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st ed., 372 pp., Addison-Wesley Longman Publ. Co., Inc., Boston, Mass.
- Hadka, D., and P. Reed (2012), Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization, *Evol. Comput.*, 20(3), 423–452, doi:10.1162/EVCO\_a\_00053.
- Hadka, D., and P. Reed (2013), Borg: An auto-adaptive many-objective evolutionary computing framework, *Evol. Comput.*, 21(2), 231–259, doi:10.1162/EVCO\_a\_00075.
- Iorio, A. W., and X. Li (2004), Solving rotated multi-objective optimization problems using differential evolution, paper presented at the 17th Australian joint conference on Advances in Artificial Intelligence, Springer, Berlin, Heidelberg, Cairns, Australia, 4–6 December.
- Kasprzyk, J. R., P. M. Reed, G. W. Characklis, and B. R. Kirsch (2012), Many-objective de Novo water supply portfolio planning under deep uncertainty, *Environ. Modell. Softw.*, 34, 87–104, doi:10.1016/j.envsoft.2011.04.003.
- Keedwell, E., and S.-T. Khu (2005), A hybrid genetic algorithm for the design of water distribution networks, *Eng. Appl. Artif. Intel.*, 18(4), 461–472, doi:10.1016/j.engappai.2004.10.001.
- Kim, J. H., T. G. Kim, J. H. Kim, and Y. N. Yoon (1994), A study on the pipe network system design using non-linear programming, *J. Korea Water Resour. Assoc.*, 27(4), 59–67.
- Kollat, J. B., and P. M. Reed (2005), The value of online adaptive search: A performance comparison of NSGA-II, ε-NSGAII, and εMOEA, paper presented at the 3rd international conference on evolutionary multi-criterion optimization, Springer, Berlin, Heidelberg, Guanajuato, Mexico, 9–11 March.
- Kollat, J. B., and P. M. Reed (2006), Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design, *Adv. Water Resour.*, 29(6), 792–807, doi:10.1016/j.advwatres.2005.07.010.
- Laumanns, M., L. Thiele, K. Deb, and E. Zitzler (2002), Combining convergence and diversity in evolutionary multiobjective optimization, *Evol. Comput.*, 10(3), 263–282, doi:10.1162/106365602760234108.
- Lee, S. C., and S. I. Lee (2001), Genetic algorithms for optimal augmentation of water distribution networks, *J. Korea Water Resour. Assoc.*, 34(5), 567–575.
- López-Ibáñez, M., L. Paquete, and T. Stutzle (2010), Exploratory analysis of stochastic local search algorithms in biobjective optimization, *Res. Rep. TR/IRIDIA/2009-015*, 209–222 pp., Springer, Berlin.
- Ostfeld, A., N. Olikar, and E. Salomons (2014), Multi-objective optimization for least cost design and resiliency of water distribution systems, *J. Water Resour. Plann. Manage.*, 140(12), 04014037, doi:10.1061/(ASCE)WR.1943-5452.0000407.
- Padhye, N., P. Bhardawaj, and K. Deb (2013), Improving differential evolution through a unified approach, *J. Global Optim.*, 55(4), 771–799, doi:10.1007/s10898-012-9897-0.
- Prasad, T. D., and N.-S. Park (2004), Multiobjective genetic algorithms for design of water distribution networks, *J. Water Resour. Plann. Manage.*, 130(1), 73–82, doi:10.1061/(ASCE)0733-9496(2004)130:1(73).
- Raad, D., A. Sinske, and J. van Vuuren (2009), Robust multi-objective optimization for water distribution system design using a meta-meta-heuristic, *Int. Trans. Oper. Res.*, 16(5), 595–626, doi:10.1111/j.1475-3995.2009.00705.x.
- Raad, D. N., A. N. Sinske, and J. H. van Vuuren (2010), Comparison of four reliability surrogate measures for water distribution systems design, *Water Resour. Res.*, 46, W05524, doi:10.1029/2009WR007785.
- Raad, D. N., A. Sinske, and J. H. van Vuuren (2011), Water distribution systems design optimisation using metaheuristics and hyperheuristics, *ORION: J. ORSSA*, 27(1), 17–44, doi:10.5784/27-1-96.
- Reca, J., and J. Martínez (2006), Genetic algorithms for the design of looped irrigation water distribution networks, *Water Resour. Res.*, 42, W05416, doi:10.1029/2005WR004383.
- Reca, J., J. Martínez, C. Gil, and R. Baños (2008), Application of several meta-heuristic techniques to the optimization of real looped water distribution networks, *Water Resour. Manage.*, 22(10), 1367–1379, doi:10.1007/s11269-007-9230-8.
- Reed, P. M., D. Hadka, J. D. Herman, J. R. Kasprzyk, and J. B. Kollat (2013), Evolutionary multiobjective optimization in water resources: The past, present, and future, *Adv. Water Resour.*, 51, 438–456, doi:10.1016/j.advwatres.2012.01.005.
- Rossman, L. A. (2000), EPANET 2 users manual, *Tech. Rep. EPA/600/R-00/057*, U.S. Environ. Prot. Agency, Cincinnati, O. H.
- Schaake, J., and D. Lai (1969), Linear programming and dynamic programming application to water distribution network design, *Res. Rep. 116*, Dep. of Civ. Eng., Massachusetts Inst. of Technol., Cambridge, Mass.
- Simpson, A., G. Dandy, and L. Murphy (1994), Genetic algorithms compared to other techniques for pipe optimization, *J. Water Resour. Plann. Manage.*, 120(4), 423–443, doi:10.1061/(ASCE)0733-9496(1994)120:4(423).
- Storn, R., and K. Price (1997), Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.*, 11(4), 341–359, doi:10.1023/A:1008202821328.
- Suribabu, C. R., and T. R. Neelakantan (2006), Design of water distribution networks using particle swarm optimization, *Urban Water J.*, 3(2), 111–120, doi:10.1080/15730620600855928.
- Veldhuizen, D. A. V. (1999), Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations, Doctoral thesis, 272 pp., Air Force Inst. of Technol., Wright-Patterson AFB, O. H.
- Vrugt, J. A., and B. A. Robinson (2007), Improved evolutionary optimization from genetically adaptive multimethod search, *Proc. Natl. Acad. Sci. U. S. A.*, 104(3), 708–711, doi:10.1073/pnas.0610471104.
- Walski, T. M. (2001), The wrong paradigm, Why water distribution optimization doesn't work, *J. Water Resour. Plann. Manage.*, 127(4), 203–205, doi:10.1061/(ASCE)0733-9496(2001)127:4(203).
- Wang, Q., D. A. Savić, and Z. Kapelan (2014), Hybrid metaheuristics for multi-objective design of water distribution systems, *J. Hydroinform.*, 16(1), 165–177, doi:10.2166/hydro.2013.009.
- Wang, Q., M. Guidolin, D. Savić, and Z. Kapelan (2015), Two-Objective design of benchmark problems of water distribution system via MOEAs: Towards the best-known approximation to the True Pareto Front, *J. Water Resour. Plann. Manage.*, 141(3), 04014060, doi:10.1061/(ASCE)WR.1943-5452.0000460.
- Zheng, F., and A. Zecchin (2014), An efficient decomposition and dual-stage multi-objective optimization method for water distribution systems with multiple supply sources, *Environ. Modell. Softw.*, 55, 143–155, doi:10.1016/j.envsoft.2014.01.028.
- Zheng, F., A. C. Zecchin, A. R. Simpson, and M. F. Lambert (2013), Noncrossover dither creeping mutation-based genetic algorithm for pipe network optimization, *J. Water Resour. Plann. Manage.*, 140(4), 553–557, doi:10.1061/(ASCE)WR.1943-5452.0000351.

- Zheng, F., A. C. Zecchin, and A. R. Simpson (2014a), A self-adaptive differential evolution algorithm applied to water distribution system optimization, *J. Comput. Civil Eng.*, 23(4), 825–835, doi:10.1061/(ASCE)CP.1943-5487.0000208.
- Zheng, F., A. R. Simpson, and A. C. Zecchin (2014b), An efficient hybrid approach for multiobjective optimization of water distribution systems, *Water Resour. Res.*, 50, 3650–3671, doi:10.1002/2013WR014143.
- Zheng, F., A. C. Zecchin, H. Maier, and A. R. Simpson (2016), Comparison of the searching behavior of NSGA-II, SAMODE, and Borg MOEAs applied to water distribution system design problems, *J. Water Resour. Plann. Manage.*, 142(7), 04016017, doi:10.1061/(ASCE)WR.1943-5452.0000650.
- Zitzler, E., and L. Thiele (1999), Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE T. Evolut. Comput.*, 3(4), 257–271, doi:10.1109/4235.797969.
- Zitzler, E., M. Laumanns, and L. Thiele (2002), SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in *Evolutionary Methods for Design, Optimisation and Control*, edited by K. Giannakoglou et al., pp. 95–100, Int. Cent. for Numer. Methods in Eng., Barcelona, Spain.
- Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca (2003), Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Trans. Evolut. Comput.*, 7(2), 117–132, doi:10.1109/tevc.2003.810758.