

Efficient multi-objective optimization methods for computationally intensive urban water resources models

Mohammad Mortazavi-Naeini, George Kuczera and Lijie Cui

ABSTRACT

Multi-objective optimization methods require many thousands of objective function evaluations. For urban water resource problems such evaluations can be computationally very expensive. The question as to which optimization method is the best choice for a given function evaluations budget in urban water resource problems remains unexplored. The main objective of this paper is to address this question. The second objective is to develop a new optimization algorithm, efficient multi-objective ant colony optimization-I (EMOACO-I), which exploits the good performance of ant colony optimization enhanced using ideas borrowed from evolutionary optimization. Its performance was compared against three established methods (NSGA-II, SMPSO, ϵ MOEA) using two case studies based on the urban water resource systems serving two major Australian cities. The case study problems involved two or three objectives and 10 or 13 decision variables affecting infrastructure investment and system operation. The results show that NSGA-II was the worst performing method. However, none of the remaining methods was unambiguously superior. For example, while EMOACO-I converged more rapidly, its diversity was comparable but not superior to the other methods. Greater differences in performance were found as the number of objectives and case study complexity increased. This suggests that pooling the results from a number of methods could help guard against the vagaries in performance of individual methods.

Key words | computationally intensive models, efficient methods, multi-objective, optimization, urban water management

Mohammad Mortazavi-Naeini (corresponding author)

George Kuczera

Lijie Cui

School of Engineering,

The University of Newcastle,

Callaghan,

NSW 2308,

Australia

E-mail: seyedmohammad.mortazavinaeini@uon.edu.au

INTRODUCTION

A large research literature exists on the development of models to simulate complex water resource systems and to optimize decisions (Labadie 2004). A recent development has been the application of multi-objective optimization (MOO) techniques to help decision makers deal with conflicting objectives-applications include reservoir operation (Ko *et al.* 1992; Liang *et al.* 1996; Consoli *et al.* 2008; Kim *et al.* 2008; Chang & Chang 2009; Shiau 2009), water supply (Savic *et al.* 1997; Farmani *et al.* 2006; Mortazavi *et al.* 2009) and groundwater (Kollat & Reed 2007). In a multi-objective problem, there exists a set of alternative solutions, referred to as the Pareto optimal solution set, for which there are no other solutions that are superior when all objectives are simultaneously

considered. The goal of multi-objective optimization is to find the Pareto optimal set.

The development of MOO algorithms has seen considerable advances, particularly since the advent of probabilistic optimization methods (Deb 2001). A number of heuristic algorithms, including evolutionary, particle swarm and ant colony optimization (ACO) methods, have been developed for solving multi-objective problems (Czyzak & Jaszkiewicz 1998; Deb 2001; Deb *et al.* 2002a; Coello Coello 2006; Huang *et al.* 2006; Martínez *et al.* 2007). The performance of these algorithms has been investigated mostly using well-known benchmark problems (Zitzler *et al.* 2000; Deb *et al.* 2002b) with their results being compared using a range of

indicators that measure the convergence and diversity of the solutions after a relatively large pre-defined number of function evaluations.

Unlike the benchmark problems, water resource applications typically use computationally expensive methods for computing their objective functions (Pierro *et al.* 2009). For example, in the case study presented in this paper involving the Canberra headworks system, a 30-year simulation with 50 replicates at monthly time steps takes approximately 6 CPU seconds, which is several orders of magnitude longer than the standard benchmark problems. Hence for an optimization involving 10,000 function evaluations, the turnaround time of nearly 17 hours is totally dominated by the simulation model rather than by the optimization algorithm. Our experience with urban water supply headworks models using long stochastically generated streamflow at monthly time steps is that simulation run times of the order of several minutes are typical. For instance, in the case study presented in Mortazavi *et al.* (2012) involving the Sydney headworks system, a 10,000-year simulation at monthly time steps takes about 40 seconds. These long simulation run times are considered an impediment to the practical uptake of MOO. While parallel computing can reduce turnaround times (Cui & Kuczera 2005), it is also imperative to identify or develop MOO methods which not only converge to the Pareto-optimal front with good diversity but do so with the fewest possible function evaluations. This is the subject of this paper.

In recent years a considerable number of studies have sought to address this issue; for example Santana-Quintero *et al.* (2006), Eskandari *et al.* (2007), Toscano-Pulido *et al.* (2007) and Pierro *et al.* (2009). The focus of these studies was identifying which algorithm can produce a better Pareto front after a relatively small number of evaluations. Durillo *et al.* (2010) took a different perspective comparing the number of evaluations to reach a certain convergence criterion threshold for seven multi-objective methods, namely NSGA-II, SPEA2, PAES, SMPSO, GDE3, AbYSS and MOCcell. They used three convergence criteria, the number of Pareto-optimal solutions and the convergence and hypervolume (HV) metrics, and concluded that SMPSO was the best of the seven algorithms.

In the context of urban water resource optimization, the question as to which MOO method is the best choice for a given function evaluations budget remains unexplored. No work has been done either to compare the efficacy of MOO methods constrained by a limited number of function evaluations or to evaluate the number of evaluations to reach convergence thresholds. The primary objective of this paper is to address this question.

A recent development in probabilistic optimization, called ACO, was proposed by Dorigo *et al.* (1996). ACO emulates the foraging behaviour exhibited by ant colonies in their search for food. ACO algorithms have been successfully applied to a number of benchmark combinatorial optimization problems, such as the travelling salesman and quadratic assignment problems (Dorigo & Stützle 2004; Stützle *et al.* 2010). The good performance of ACO in single objective optimization motivated researchers to apply ACO to multi-objective problems (Iredi 2001; Shelokar *et al.* 2002; García-Martínez 2004; Alaya *et al.* 2007; Angus 2007b; Bui *et al.* 2008; Angus & Woodward 2009). However, these studies focussed on combinatorial problems, while many engineering problems involve decision variables with a continuous, real-valued domain. Very few studies have applied multi-objective ACO (MOACO) methods to problems with continuous real-valued search spaces (Shelokar *et al.* 2002; Angus 2007a; Afshar *et al.* 2009).

Usually optimizing problems in urban water resources is computationally demanding. This provides a strong motivation to develop new optimization methods that require fewer evaluations to converge. The secondary objective of this paper is to explore whether the MOACO approach can be successfully adapted to solve computationally intensive problems typical of urban water resources.

This paper is organized as follows. First, a review of existing MOO methods is presented from which three benchmark methods are selected. Then the principles of ACO are described after which a new MOACO algorithm is proposed. This is followed by a discussion of the performance metrics and a brief description of the two urban water resource case studies. Finally, the performance of the new MOACO method is compared against three benchmark methods using benchmark problems and water resource case studies.

REVIEW OF EXISTING MOO METHODS

During the last decade, there has been considerable effort towards developing efficient MOO methods for computationally intensive problems. [Eskandari *et al.* \(2007\)](#) proposed a new algorithm called fast Pareto genetic algorithm (FastPGA) which uses a new fitness assignment and ranking strategy. They compared their method against NSGA-II using four benchmark problems known as the Zitzler–Deb–Thiele (ZDT) test suite and found that FastPGA outperformed NSGA-II in terms of convergence and diversity after completion of a relatively small number of evaluations (6,500 and 10,000). [Pierro *et al.* \(2009\)](#) applied two hybrid algorithms, ParEGO and LEMMO, to optimize cost and pressure deficit in water distribution network systems and compared their performance against an evolutionary algorithm called PESA-II. They found for a medium sized network that LEMMO produced solutions after 10,000 evaluations that were comparable to those produced by PESA-II after 100,000 evaluations. However, for a large network involving 600 decisions LEMMO did not perform well. [Santana-Quintero *et al.* \(2006\)](#) developed a new particle swarm optimization (PSO) in conjunction with a local search method. They tested their method for two sets of benchmark problems, ZDT and DTLZ. After 4,000 evaluations, the method was shown to outperform the well-established benchmark NSGA-II algorithm for the ZDT problems but it did not perform well for DTLZ problems. In a similar study, [Toscano-Pulido *et al.* \(2007\)](#) presented an efficient multi-objective PSO method EMOPOS, which, after 2,000 evaluations, produced a solution closer to the optimal Pareto front than did NSGA-II for the same number of evaluations.

The focus of the above-mentioned studies was evaluating the performance of algorithms after a pre-defined number of evaluations. This type of analysis can be approached from a different perspective. For instance, [Nebro *et al.* \(2008\)](#) ranked five multi-objective optimization methods, namely NSGA-II, SPEA2, PAES, OMOPSO, AbYSS and MOCcell, according to the number of evaluations required to produce Pareto front with a certain accuracy. They found MOCcell, OMOPSO, and AbYSS the most competitive algorithms. In a similar study, [Durillo *et al.* \(2010\)](#) analysed the performance of similar multi-objective methods, except SMPSO replaced

OMOPSO and GDE3 was added, using three criteria, the number of Pareto-optimal solutions, the convergence metric and the HV metric. They concluded that SMPSO performed the best of the seven algorithms.

In this study three methods, namely NSGA-II, ϵ MOEA and SMPSO, were selected for comparison based on their usage and performance reported in the literature and on the availability of computer codes. NSGA-II has been widely applied in the MOO literature, often being used as a benchmark for new developed methods in computationally intensive problems ([Santana-Quintero *et al.* 2006](#); [Eskandari *et al.* 2007](#); [Toscano-Pulido *et al.* 2007](#); [Nebro *et al.* 2008](#); [Durillo *et al.* 2010](#)). There is no study evaluating the performance of ϵ MOEA with a limited number of evaluations. Finally, SMPSO was chosen due to its superior performance among the seven state-of-the-art MOO methods evaluated by [Durillo *et al.* \(2010\)](#).

In the following sections SMPSO and ϵ MOEA algorithms are described briefly. For more details readers are referred to [Mortazavi-Naeini \(2013\)](#).

SMPSO

PSO is a population-based metaheuristic method mimicking the social behaviour of bird flocking. The initial ideas on particle swarms were proposed by [Kennedy & Eberhart \(1995\)](#). The good performance of PSO in single objective optimization applications motivated researchers to extend it to multi-objective problems. Moore and Chapman developed the first multi-objective implementation of PSO in 1999 and since then more than 20 different methods have been reported ([Reyes-Sierra & CoelloCoello 2006](#)).

ϵ -dominance multi-objective evolutionary algorithm

ϵ MOEA

ϵ MOEA is a member of the evolutionary algorithm family. Its distinctive feature is the use of the ϵ -dominance concept which divides the objective space into hyperboxes of size ϵ and allows only one non-dominated solution to reside in each box ([Laumanns *et al.* 2002](#)). In this study the ϵ MOEA implementation described by [Jefferson *et al.* \(2005\)](#) was used.

ENHANCED MOACO

ACO is a recently developed heuristic optimization method. It was inspired by the fact that some species of ants are blind but nonetheless can find the minimum path between their nest and food. This is because of a chemical substance called pheromone that ants deposit when they travel on a route (Dorigo & Stützle 2004). Based on the behaviour of real ants, Dorigo *et al.* (1991, 1996) developed the first ACO method called Ant System (AS) to solve the travelling salesman problem and job-shop scheduling problem.

Several issues need to be addressed when adapting ACO to multi-objective optimization, including the number of pheromone and heuristic matrices and the pheromone updating procedure. The fact that there is considerable choice has resulted in a wide range of published MOACO methods (Martínez *et al.* 2007; Angus & Woodward 2009).

Most MOACO approaches are extensions of well-known single objective ACO methods. For example, Baran & Schaerer (2003) and Doerner *et al.* (2003) adapted the Ant Colony System (ACS) while Bui *et al.* (2008) adapted the AS. Although MOACO methods differ in detail, all share the following common steps:

- Step 1: Initialize parameters
- Step 2: Construct solutions
- Step 3: Find and archive non-dominated solutions
- Step 4: Update pheromone
- Step 5: Go to Step 2 if the termination condition is not satisfied

Generally MOACO methods can be categorized according to their number of pheromone and heuristic matrices (Martínez *et al.* 2007; Angus & Woodward 2009). Iredi (2001) proposed an approach for bi-criterion optimization problems which uses cooperative ant colonies and multiple pheromone and heuristic matrices. Doerner & Gutjahr (2004) and Cardoso *et al.* (2003) developed the Pareto-ACO (P-ACO) and multi-objective network optimization based on ACO respectively with a single heuristic matrix and several pheromone matrices. In contrast, crowding population-based ant colony optimization and multiple ant colony system methods were applied with multiple heuristic matrices and a single pheromone matrix (Baran & Schaerer

2003; Angus 2007b). Indeed, in these methods, diversity is achieved across the Pareto front through the use of heuristic rather than pheromone information. McMullen (2001), Gravel *et al.* (2002) and T'Kindt *et al.* (2002) developed methods which used a single pheromone and a single heuristic matrix.

In some of above methods, a single ant colony was used (Doerner & Gutjahr 2004; Alaya *et al.* 2007), while in the other methods, multiple colonies were used (Iredi 2001; Baran & Schaerer 2003; Doerner *et al.* 2003). The main reason for having multiple colonies is to treat objectives independently. Doerner *et al.* (2003) introduced COMPET-ants with multiple colonies. Each colony corresponds to an objective. One drawback of this approach is that by allowing ants to explore individual objectives independently, they are more likely to explore the extremes of the Pareto front and neglect the compromise trade-off points. For this reason Doerner *et al.* (2003) introduced the *spies* idea to facilitate sharing and exchanging information between colonies. In a similar way, Alaya *et al.* (2007) used multiple ant colonies with each colony dedicated to a single different objective using its own pheromone and heuristic information to build solutions. To avoid exploring extremes of the Pareto front, they introduced an extra colony that aimed at optimizing all objectives. They compared four MOACO methods with different numbers of colonies and pheromone matrices and found the method using a single colony and multiple pheromone matrices performed best. Other researchers used multiple colonies for other purposes. Iredi (2001) used multiple colonies to force ants to find good solutions along the whole of the Pareto front. The difference between Iredi's (2001) approach and the approaches of Alaya *et al.* (2007) and Doerner *et al.* (2003) is that the former used multiple pheromone matrices in each colony. This approach is conceptually similar to having a single ant colony with multiple start points but it is different because altered weights have been applied in each colony to weight the pheromone and heuristic information.

Doerner & Gutjahr (2004) in P-ACO assigned a set of weights randomly at each iteration for each ant. Alaya *et al.* (2007) did not apply weights. However, since at each iteration a randomly selected objective was optimized,

they implicitly applied binary weights (0 or 1). Angus (2007b) used the average-rank-rate method in which higher scoring objectives were assigned a greater weighting.

The use of different approaches to set weights can result in different search behaviours (López-Ibáñez et al. 2004). The increase in required memory associated with multiple pheromone matrices can be of concern if the actual problem size is sufficiently large (Angus & Woodward 2009). Most of the above mentioned approaches were applied to bi-objective cases. Thus, there is little guidance on application of these approaches to problems with three or more objectives.

One of the challenging aspects of MOACO is the definition of a heuristic information matrix. This is problem-specific and not necessarily easy to establish (Doerner et al. 2003; Coello Coello et al. 2007). Developing a proper heuristic information matrix is likely to be even more challenging in a multi-objective setting because of the need to define heuristic information for every objective.

One of the most important steps of MOACO is the pheromone update. This involves two considerations that affect performance. The first is the selection of the routes to be updated and the second is the amount of pheromone to be deposited on the selected routes.

Several approaches have been proposed in the literature for selecting routes. The selected routes may be based on the non-dominated solutions within an iteration (Iredi 2001), the non-dominated solutions found so far (Baran & Schaerer 2003; Alaya et al. 2007; Afshar et al. 2009) or the best (and the second-best) solutions according to each objective (Doerner & Gutjahr 2004). Bui et al. (2008) compared several pheromone updating methods including updating based on all solutions from the current iteration, non-dominated solutions in the current iteration, and non-dominated solutions of all iterations. Their conclusion was that updating the non-dominated solutions of all iterations outperformed other updating methods. This finding is consistent with the good performance of the elitism strategy in single objective optimization.

However, a drawback of updating non-dominated solutions found in all iterations is that adding pheromone to these routes continuously may induce premature convergence and thus prevent the algorithm from generating an even coverage of the Pareto front (Angus & Woodward 2009). To abate loss of diversity, several methods have been

suggested in the literature. Angus (2007b) used dominance ranking according to a non-dominated sorting technique to produce an even coverage of the Pareto front. Alaya et al. (2007) updated the pheromone value of a route only once despite how many solutions contained it. Bui et al. (2008) introduced an aging factor to deal with this issue. The main idea is to deposit more pheromone on routes associated with more recent solutions in the archive of non-dominated solutions. They defined the age factor (AF) as

$$AF = \frac{1}{\text{Current iteration number} - \text{iteration number solution added to archive} + 1} \quad (1)$$

A variety of pheromone updating methods has been developed. In a single objective ACO minimization problem, pheromone is updated according to the inverse of objective function value – that is, depositing more pheromone on routes with the smaller objective function values encourages ants to follow those routes. Several researchers have tried to extend this idea to multi-objective problems. Baran & Schaerer (2003) used the inverse of the product of two objective function values for updating pheromone. A drawback of this method is that the amount of pheromone can be very sensitive to the objective value. This can cause premature convergence. Alaya et al. (2007) used the following equation to update pheromone:

$$\Delta\tau^i(c) = \frac{1}{1 + f_i(S^i) - f_i(S_{\text{best}}^i)} \quad (2)$$

where $\Delta\tau^i(c)$ is the quantity of pheromone deposited on a route (c) at the i th iteration, $f_i(S^i)$ is the value of an objective function for the current iteration and $f_i(S_{\text{best}}^i)$ is the value of the best solution found so far. The value of $\Delta\tau^i(c)$ is scaled between 0 and 1.

To avoid the drawbacks associated with objective-dependent updating pheromone approaches, several researchers introduced updating methods independent of the objectives. Iredi (2001) suggested an updating rule where every ant is allowed to update the amount of pheromone equal to $1/L$ where L is the number of ants that are allowed to update in the current generation. Doerner et al. (2003) updated pheromone for only a number of the best

ants ranked according to solution quality. The ants deposit pheromone based on their rank. In a similar way, Angus (2007b) updated pheromone based on the ant's ranking. He used dominance ranking according to a non-dominated sorting approach such as that of the NSGA-II algorithm. López-Ibáñez et al. (2004) suggested all ants deposit a constant amount of pheromone. Similarly, Doerner & Gutjahr (2004) used a constant amount of 10 and 5 for the best and the second-best ants respectively.

Towards an improved MOACO algorithm

The review of existing MOACO methods in the preceding section showed that there are several important aspects that need to be addressed in MOACO algorithms. These include the number of heuristic and pheromone matrices, the transition rule, the pheromone updating procedure and the specification of heuristic information.

In this study an enhanced MOACO method called efficient MOACO (EMOACO) is proposed. The main features of EMOACO are as follows:

1. Use a single colony with a single pheromone matrix regardless of the number of objectives. This avoids the complexity of colonies communicating in the search process and also avoids the need to assign weights.
2. Do not use problem-specific heuristic information. This ensures EMOACO is problem-independent.
3. Apply a constant amount of pheromone, defined as C , when updating routes corresponding to non-dominated solutions to maximize diversity in the Pareto front. This is motivated by the fact that all the points on the Pareto front should be treated equally.
4. Use the MAX-MIN AS method with τ_{\max} defined by:

$$\tau_{\max} = \frac{C}{\rho} \quad (3)$$

where ρ is evaporation rate.

5. Introduce a pheromone aging factor to reduce the chance of premature convergence. This is accomplished using the following pheromone update:

$$\Delta\tau = \frac{C}{AF} \quad (4)$$

where AF is the number of iterations since the current non-dominated solution was added to the archive as described in Equation (1).

6. Introduce a mutation concept, similar to the mutation operator in evolutionary algorithms, to avoid the algorithm being trapped at local minima and foster diversity (Srinivas & Deb 1994; Deb et al. 2002a). To facilitate this feature in MOACO, the following route selection process is proposed. For ant k and decision i , the route r_{ki} is selected using:

$$r_{ki} = \begin{cases} \text{Randomly select one of the } N_i \text{ routes} & \text{if } q \leq q_0 \\ \text{Select route applying probability rule} & \text{otherwise} \end{cases} \quad (5)$$

where N_i is number of routes available for decision i , q_0 is a parameter in the interval (0,1) and q is a random sample from a uniform distribution over the interval (0,1). The probability rule is central to ACO. It defines the probability that an ant at node i will travel on link j :

$$P_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{i=1}^N [\tau_{ij}]^\alpha [\eta_{ij}]^\beta} \quad (6)$$

where τ_{ij} is the pheromone trail strength and η_{ij} is heuristic information. The parameters α and β are introduced to control the relative importance of the pheromone and heuristic information respectively.

7. Introduce an operator similar to the crossover operator in evolutionary algorithms to allow, for much of the time, exploration of the search space in the neighbourhood of the parent decisions. This motivated the introduction of the adjacency concept in pheromone updating as a way of mimicking the ability of the crossover operator to explore the search space. The adjacency concept is implemented by depositing pheromone on routes adjacent to the current non-dominated solution routes. The proximity of route (i, j) (defined as the j th segment between decisions i and $i+1$) to the nearest non-dominated route is given by its adjacency score k , defined as $|j-j^*|$ where (i, j^*) is the closest non-dominated route to route (i, j) . The pheromone deposit on route (i, j) then

becomes:

$$\Delta\tau_{ij} = \begin{cases} \frac{c}{(AF)} & \text{if } (i, j) \text{ has an adjacency score } k = 0 \\ \frac{c}{(AF)(1+k)} & \text{if } (i, j) \text{ has an adjacency} \\ & \text{score } 0 < k \leq k_{\max} \text{ and } u < P_{adj} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where u is a random uniformly distributed number over the interval $(0,1)$, P_{adj} is the adjacency probability which determines the probability of depositing pheromone on an adjacent route and k_{\max} is the maximum number of adjacent routes.

8. Introduce revisiting non-dominated solutions to overcome potential stagnation. It was found that the adjacency pheromone update improved convergence to the Pareto-optimal frontier as long as new solutions were being added to the set of non-dominated solutions. However, the longer it took to find a new non-dominated solution, the greater was the likelihood of stagnation. To overcome this problem, the strategy of revisiting and mutating one of the current non-dominated solutions was introduced. The strategy commences when the number of iterations during which no new non-dominated solution is added exceeds a predefined value, No Improvement. One of the current non-dominated solutions is then selected randomly and one of its decisions is changed randomly. This procedure continues until a new non-dominated solution is found. The No Improvement value is defined as the number of iterations required to reduce pheromone by evaporation from the maximum allowable pheromone τ_{\max} to the minimum allowable pheromone τ_{\min} . It can be shown to be:

$$\text{No improvement} = \frac{-\log \left[\frac{(avg-1)^{\sqrt[n]{p_{best}}}}{1-\sqrt[n]{p_{best}}} \right]}{\log(1-\rho)} \quad (8)$$

where avg is the average number of different choices available to an ant at each step and n is the number of segments. It is assumed that a run of MAX-MIN AS has converged if the best found route has been constructed with a sufficiently significant probability p_{best} (Stützle & Hoos 2000).

To summarize these changes more formally, Figure 1 presents the pseudo code for EMOACO.

EMOACO, like all other MOACO methods, randomly selects the initial routes traversed by each ant. The rate of convergence is affected by how close these initial routes are located to the Pareto-optimal routes. With this in mind, the following simple heuristic was adopted: EMOACO starts with the decision space being split into eight rather than 256 segments – this reduces the number of decision combinations and thus improves the chance of EMOACO finding routes in the neighbourhood of Pareto-optimal solutions. Once a predetermined number of evaluations (500 in this study) is completed, the routes of the current non-dominated solutions are mapped, as initial routes, to the final search space where the number of segments for each decision is substantially increased (256 segments in this study). This enhancement to EMOACO is referred to as EMOACO-I.

EVALUATION OF MULTI-OBJECTIVE PERFORMANCE

Various performance metrics for measuring the quality of a Pareto-optimal set have been proposed to compare the performance of different multi-objective algorithms (Deb 2001). However, there is no clear consensus in the literature on how the performance of multi-objective methods should be evaluated or compared. Deb & Jain (2002) suggested the use of metrics to characterize the two main functional objectives of MOO methods, proximity and diversity. Hadka & Reed (2011) investigated a broad range of performance metrics including HV, generational distance (GD), inverse generational distance, additive epsilon indicator (ϵ_+ -indicator) and spread. They recommended three metrics to characterize the three main functional objectives of MOO methods associated with proximity, diversity and consistency. The GD and HV metrics are used to assess proximity and diversity respectively, while the ϵ_+ -indicator is used to assess the consistency of the proximity of solutions. In this paper the three measures recommended by Hadka & Reed (2011) were used to compare the competing MOO algorithms. In all of these measures, normalized objective values are used. The following sections discuss each of these measures in more detail.

```

Assign parameters /* adjacency probability( $P_{adj}$ ), Number of decisions, Number of objectives, Number of segments (n),
Number of ants, Initial pheromone ( $\tau_0$ ), pheromone deposit (C), evaporation rate ( $\rho$ ), Number of adjacency routes ( $K_{max}$ ), mutation
rate ( $q_0$ ),  $avg$ ,  $P_{best}$  */
Calculate /*  $\tau_{max}$ ,  $\tau_{min}$ ,  $NoImprovement$  */
Initialization
For  $i = 1$  to number of decisions do
  For  $j = 1$  to number of segments do
    Create a route from decision  $i$  to decision  $i+1$  ( $R_{ij}$ )
  End for
End for
Initialize pheromone trail /*put initial pheromone ( $\tau_0$ ) on all routes*/
Initialize archive of non-dominated solutions found so-far (ND) =  $\phi$ 
Initialize Count = 0 /*count number of iterations without any changes in ND */
While stopping criteria is not met do
  /*a sufficient good fitness or a maximum number of iterations or no results improvements after a specific number of iterations*/
   $N = -1$  /*flag changes in ND */
  For all ants do
    If (Count >  $NoImprovement$ ) then /*revisiting solutions */
      Chose randomly one of the solutions in ND
      Change randomly one of its routes
      Save the changed solution as a new solution
    Else
      For all decisions do
        Select a random number between 0 and 1 ( $r$ )
        If ( $r < q_0$ ) then /* mutation */
          Select route randomly
        Else
          Select route based on probabilistic transition rule
        End if
      End for
      Assign decision values to solutions based on the route traversed by
      the ant
    End if
  End for
  For all solutions do
    Evaluate the objective function values corresponding to the solution
    If (new solution dominates any of ND ) then
      Delete dominated solutions
      Add the new solution to ND
       $N=0$  /*flag changes in ND */
    Else if (none of ND dominate the new solution) then
      Add the new solution to ND
       $N=0$  /*flag changes in ND */
    End if
  End for
  If ( $N \neq 0$ ) then /* No changes in ND */
    Count = Count + 1
  Else
    Count = 0
  End if
  Evaporate pheromone of all routes
  For all ND do
    Calculate aging factor (Eq. (1))
    Calculate  $\Delta C$  (Eq.(4))
    Update pheromone trail for the solution
    For all decisions do
      Select a random number between 0 and 1 ( $r$ )
      If ( $r < P_{adj}$ ) then /* adjacency */
        Add pheromone to adjacency routes (Eq.(7))
      End if
    End for
  End for
  For all decisions do
    If (pheromone trail <  $\tau_{min}$ ) then
      pheromone trail =  $\tau_{min}$ 
    Else if (pheromone trail >  $\tau_{max}$ ) then
      pheromone trail =  $\tau_{max}$ 
    End if
  End for
End while
Output
  Print set ND

```

Figure 1 | Pseudo code for EMOACO algorithm.

Convergence (generational distance) metric

The convergence metric is a proximity or distance measure describing how close a set of non-dominated solutions is to the Pareto-optimal front (Van Veldhuizen & Lamont 2000). The minimum normalized Euclidean distance from each point i in the non-dominated solution set (Q) to the reference solution set (P^*) is calculated using the following equation (Deb & Jain 2002):

$$d_i = \min_{j \in P^*} \sqrt{\sum_{n=1}^K \left(\frac{f_n(i) - f_n(j)}{f_n^{\max} - f_n^{\min}} \right)^2} \quad (9)$$

where f_n^{\max} and f_n^{\min} are the maximum and minimum function values of the n th objective function in P^* . $f_n(i)$ is the n th function value of point i in the set Q and $f_n(j)$ is the n th function value of point j in the set P^* . K is the number of objectives. The average of d_i is taken to be the convergence metric. The smaller the value of this metric, the closer the solutions are to the reference solution set.

Hypervolume ratio (HVR)

The HV metric is defined as the volume (in objective space) enclosed by a reference point and the non-dominated solution set. The reference point can be defined using the worst objective function values. In this study, the method developed by Fonseca et al. (2006) is used. This method is coded in an R package called 'emoa' which can be downloaded from www.statistik.tu-dortmund.de/~olafm/software/ (last accessed 24/5/2012).

The HVR normalizes the HV to facilitate comparisons (Deb 2001). It is defined as the ratio of the HV for a non-dominated solution set Q and the HV of a reference solution set P^* which is taken to be the approximate Pareto-optimal solution set:

$$HVR = \frac{HV(Q)}{HV(P^*)} \quad (10)$$

Additive epsilon indicator ($I_{\epsilon+}$)

The convergence and HV metrics measure the proximity and diversity of a non-dominated solution set. However,

these measures fail to identify a non-dominated solution set which contains one or more solutions with poor proximity. To deal with this, another measure called $I_{\epsilon+}$ is introduced. It is defined as the smallest distance one would need to translate every point in the non-dominated solution set Q so that it dominates a reference solution set P^* (Zitzler et al. 2002). Formally, if x_1 is an element of Q , x_2 is an element of P^* and K is the number of objectives, the $I_{\epsilon+}$ metric is (Durillo & Nebro 2011)

$$I_{\epsilon}(Q) = \inf_{\epsilon \in R} \{ \forall x_2 \in P^* \exists x_1 \in Q: x_1 < \epsilon x_2 \} \quad (11)$$

where $x_1 < \epsilon x_2$ if and only if $\forall 1 \leq i \leq K: f_i(x_1) < \epsilon + f_i(x_2)$.

It is stressed that the convergence, HVR and $I_{\epsilon+}$ metrics require knowledge of the reference solution set, which in this paper is referred to as the approximate Pareto-optimal solutions set.

OVERVIEW OF CASE STUDIES TO EVALUATE MOO METHODS

Traditionally, performance of new multi-objective optimization methods is compared against well-known methods using benchmark problems such as ZDT and DTLZ problems (Zitzler et al. 2000; Deb et al. 2002b). In this study, six benchmark problems including ZDT1, ZDT3, ZDT4, ZDT6, DTLZ1 and DTLZ6 were employed to compare the performance of new developed methods, EMOACO and EMOACO-I.

Two urban water resource case studies based on the Canberra and Sydney headworks supply systems are used to assess the performance of different MOO methods. The purpose of this section is to summarize briefly the decisions and objectives used in these case studies.

Canberra headworks system

The Canberra headworks system has four reservoirs supplying water to the city of Canberra. Water is harvested from two catchments, Cotter and Googong, which flank the city to the west and east, respectively. A network of pipelines, pumping stations and treatment plants connects four

reservoirs to the Canberra demand zone. Releases from the reservoirs have to meet not only the consumptive needs of the Canberra urban area, but also environmental flow requirements defined in the water authority's operating licence.

Thirteen decision variables are considered which are categorized either as operational in that they control the running of the system or as infrastructure in that they define the physical characteristics of the system. The decisions are summarized in Table 1. Operational decisions include storage triggers for imposing restrictions, a pump mark for turning on the Murrumbidgee–Googong diversion, and parameters that determine the balance of storage between the Googong and Cotter catchments. Infrastructural decisions involve major capital works to upgrade the capacity of Cotter Dam, a pumping station, the Stromlo water treatment plant and the construction of the Murrumbidgee diversion. In addition, the installation of rainwater tanks on individual allotments is supported to harvest roof water and use it for non-potable indoor and outdoor uses.

The following objectives were considered:

1. **Minimize the frequency of restrictions** expressed as the percentage of months during which restrictions on water consumption are imposed. The restriction time fraction criterion is an important level-of-service measure.

2. **Minimize the expected present worth cost (\$)** defined as the sum of capital and discounted expected operating costs and the costs of unplanned shortfalls. The capital cost represents the cost of building new infrastructure such as dams or water treatment plant capacity upgrades. Two capital items involve a binary choice: if the item is selected by the optimization, then there is fixed setup cost along with a unit cost; if the item is not selected, there is zero capital cost. The operating cost includes the costs for pumping from Cotter reservoir and the Murrumbidgee River to the Stromlo water treatment plant and pumping from the Murrumbidgee River to Googong Reservoir, and the transfer and treatment costs associated with Stromlo and Googong water treatment plants. A 5% discount rate was used.

An unplanned shortfall arises when the system is unable to supply demand that may be restricted; in most cases, an unplanned shortfall arises when reservoirs empty and there is insufficient streamflow. To steer the optimization away from solutions that result in unplanned shortfalls, a penalty of \$1,000,000 per ML (megalitres) unplanned shortfalls is added to the present worth cost.

1. **Minimize the fraction of time that total reservoir storage falls below 20%.** This objective measures the vulnerability of the supply system to drought condition.

Table 1 | List of decision variables used in Canberra case study

Decision	Decision variable	Lower limit	Upper limit	Category
1	Murrumbidgee pump trigger	0	1	Operational
2	Murrumbidgee diversion capacity	0	6,000	Infrastructure
3	Cotter pump capacity	3050	10,000	Infrastructure
4	Googong incremental gain	20	500	Operational
5	Googong base gain	9000	11,000	Operational
6	First restriction trigger level 1	0	0.999	Operational
7	Trigger intervals	0.05	0.25	Operational
8	Stromlo water treatment plant capacity	7625	15,000	Infrastructure
9	Cotter incremental gain	20	500	Operational
10	Cotter base gain	9000	11,000	Operational
11	Googong water treatment plant capacity	8235	15,000	Infrastructure
12	Cotter Reservoir capacity upgrade	0	100,000	Infrastructure
13	Number of houses used water tank	0	15,000	Infrastructure

Apart from the constraint on unplanned shortfalls, which was implemented using a penalty function approach, the only other constraints were the limits on the decision variables summarized in Table 1.

Sydney headworks system

This case study considers a 7-million population scenario for the Sydney system. The existing system has a total storage capacity of 3,343,487 ML. Warragamba reservoir is the largest reservoir in the system with a capacity of 2,031,000 ML. The Sydney demand zone, which serves approximately 90% of the population, is supplied by Warragamba reservoir together with a number of smaller reservoirs, Avon, Woronora, Cataract, Nepean and Cordeaux. In contrast, the South demand zone, which serves the remaining 10% of the population, is only supplied by Nepean and Avon reservoirs. Readers are referred to Mortazavi et al. (2012) for a more detailed description of the system.

In this case study, 11 decision variables, listed in Table 2, were selected. Decisions 1 and 2 control the pump transfer of water from the Shoalhaven basin. Decisions 3 and 4 define the first stage of the drought contingency plan (DCP) to determine restriction levels. Decisions 5 and 6 define the second stage of the DCP. When the total storage fraction falls below the trigger given by decision 6, the already-constructed desalination plant with capacity given by decision 5 is activated. Decision 7 defines the capacity of Welcome Reef

Reservoir. Decisions 8 and 9 define the priority for storing water in Warragamba. Depending on the values assigned to decisions 8 and 9, water may be preferentially stored in Warragamba or in the rest of the system. Decisions 10 and 11 define the maximum monthly Wollondilly transfer capacity during September to March and at other times respectively. The lower limit on these decisions corresponds to that recommended by Scott & Grant (1997). These two decisions are active in the three-objective scenario and fixed in the other scenarios.

The following three objectives were considered:

1. **Minimize frequency of restrictions (%)** defined as the percentage of months during which restrictions on water consumption are imposed.
2. **Minimize the present worth cost (\$)** defined as the sum of capital and discounted expected operating costs and the costs of unplanned shortfalls. The capital cost represents the cost of building new infrastructure, which in this case study is the Welcome Reef dam and/or the desalination plant. The operating cost includes the costs for pumping transfers from the Shoalhaven and operation of the desalination plant. A 5% discount rate was used.
3. **Minimize environmental stress on the Wollondilly River.** The following environmental stress metric was adopted to penalize the adoption of maximum regulated flow limits, defined by decisions 10 and 11, in excess of

Table 2 | List of decision variables used in Sydney case study

Decision variable	Description	Lower limit	Upper limit	Category
1	Pump mark Warragamba	0.3	1	Operational
2	Pump mark Avon	0.3	1	Operational
3	Level 1 restriction trigger	0.05	0.95	Operational
4	Trigger increment	0.05	0.25	Operational
5	Desalination plant capacity (ML/day)	0	1,000	Infrastructure
6	Desalination plant trigger	0.05	0.95	Operational
7	Welcome Reef capacity (ML)	0	100,000	Infrastructure
8	Warragamba base gain	8,000	12,000	Operational
9	Warragamba incremental gain	10	200	Operational
10	Maximum Wollondilly flow during September to March (ML/month)	12,200	100,000	Operational
11	Maximum Wollondilly flow at other times (ML/month)	18,300	100,000	Operational

those recommended by Scott & Grant (1997).

$$\text{Stress}(m) = \begin{cases} \max \left[0, 5 \left(\frac{q_m - 12200}{12200} \right) \right] \\ \text{if } m \in \{\text{September, ...March}\} \\ \max \left[0, \left(\frac{q_m - 18300}{18300} \right) \right] \\ \text{if } m \in \{\text{April, ...August}\} \end{cases} \quad (12)$$

where q_m is the actual regulated release in the Wollondilly in month m and $\text{Stress}(m)$ is the penalty for exceeding the recommended flow limits in month m . The environmental stress criterion is the sum of the monthly stresses over the simulation.

Apart from the constraint on unplanned shortfalls, which was implemented using a penalty function approach, the only other constraints were the limits on the decision variables summarized in Table 2.

EVALUATION OF MOO ALGORITHM PERFORMANCE FOR BENCHMARK PROBLEMS

This section evaluates the performance of two MOACO methods, EMOACO and EMOACO-I, against three benchmark MOO algorithms using six benchmark problems.

In Tables 3–5, the average rank for each method is presented for a selected number of evaluations for the three performance metrics. The average rank is the arithmetic mean of the ranks assigned to each method for each problem. Ranks 5 and 1 are assigned to the best and worst performing method respectively. The maximum average rank is in bold. EMOACO was found to be the best

Table 3 | Average rank values based on the convergence measure for five MOO methods

Number of evaluations	EMOACO-I	EMOACO	εMOEA	NSGA-II	SMPSO
1,000	3.5	4.83	2.66	1.16	2.83
2,000	3.83	4.66	2.5	1.16	2.83
3,000	4	4.5	2.33	1.16	3
4,000	4.5	4.5	2.5	1.16	3.16
5,000	4.33	4.33	2.83	1.16	3.16
10,000	4.33	4.16	2.83	1.16	4.16

Table 4 | Average rank values based on the HVR measure for five MOO methods

Number of evaluations	EMOACO-I	EMOACO	εMOEA	NSGA-II	SMPSO
1,000	2.33	3.5	1.33	1.16	2.83
2,000	3	3.83	2	1.5	2.66
3,000	3.16	3.83	2	1.33	3.5
4,000	3	4	2.16	1.33	4
5,000	3.5	3.83	2	1.33	4.16
10,000	3.16	3.33	2.16	1.66	4.5

Table 5 | Average rank values based on the $I_{\epsilon+}$ measure for five MOO methods

Number of evaluations	EMOACO-I	EMOACO	εMOEA	NSGA-II	SMPSO
1,000	3.33	4.83	2.5	1.5	2.83
2,000	3.83	4.16	2.33	1.33	3.33
3,000	3.83	4.16	2.33	1.33	3.33
4,000	3.66	4	2.83	1.33	3.16
5,000	4	4.16	2.83	1.83	3.66
10,000	3.83	4	2.66	2	4.16

method for all three performance metrics for evaluations up to 5,000 while EMOACO-I was the second best method.

To elaborate more on the performance of MOO methods, we selected two benchmark problems, ZDT1 and DTLZ1. Figures 2 and 3 present the three performance metrics for the MOO methods for ZDT1 and DTLZ1 respectively. These graphs demonstrate the superiority of EMOACO and EMOACO-I for all three measures except HVR for DTLZ1. It is noted that the performance of EMOACO and SMPSO is very similar for more than 10,000 iterations.

EVALUATION OF MOO ALGORITHM PERFORMANCE FOR URBAN WATER RESOURCE PROBLEMS

This section evaluates the performance of three benchmark MOO algorithms, NSGA-II, εMOEA and SMPSO, and two MOACO methods, EMOACO and EMOACO-I. The primary objective is to identify the most efficient algorithm for typical urban water resources problems using a relatively small number of evaluations – this is motivated by the fact that function evaluations for urban water resource problems

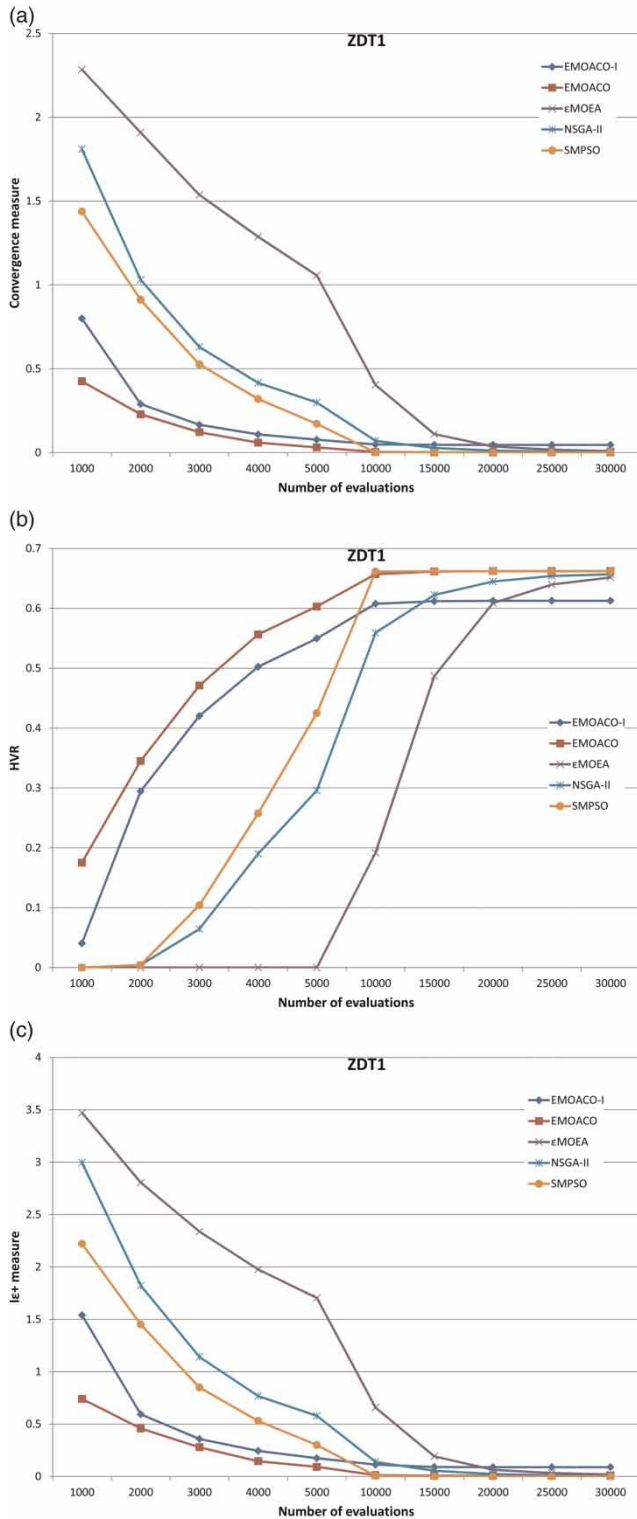


Figure 2 | Comparison of performance for five MOO methods as a function of the number of evaluations for the ZDT1 problem. (a) Convergence, (b) HVR, (c) $I_{\epsilon+}$.

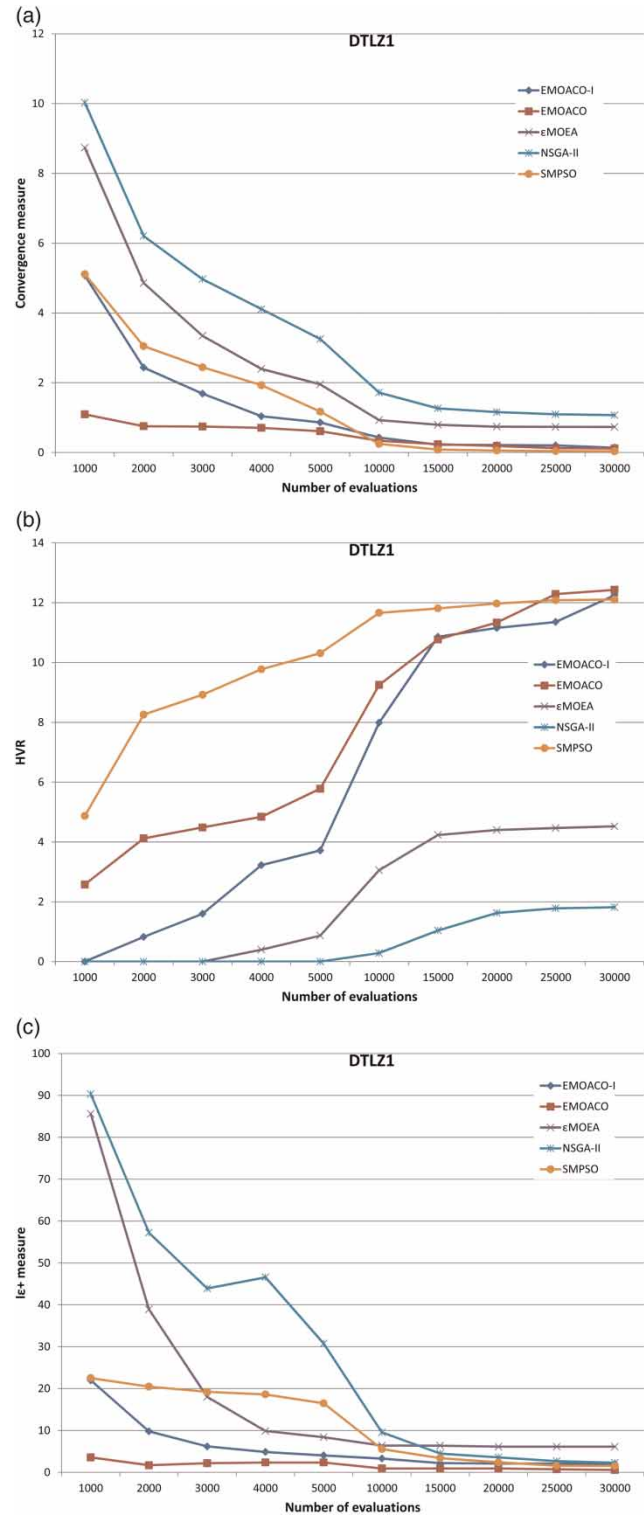


Figure 3 | Comparison of performance for five MOO methods as a function of the number of evaluations for the DTLZ1 problem. (a) Convergence, (b) HVR, (c) $I_{\epsilon+}$.

are computationally expensive. This section is organized as follows. First, the parameters of all five MOO methods are tuned. Then the performance of these methods is compared using the three metrics recommended by Hadka & Reed (2011).

Tuning

The five MOO methods were tuned to obtain ‘good’ parameters to ensure a fair comparison. To ensure consistency across methods, binary coding with 8 bits (equivalent of 256 segments) and the same number of evaluations, that is, 10,000, was used. All methods were run 10 times with different initial random number seeds. p_{best} in Equation (8) was set to 0.05 (Stützle & Hoos 2000). Polynomial mutation and uniform crossover operators were applied to SMPSO and NSGA-II respectively. One-point crossover with bitwise mutation was used in ϵ MOEA with an initial population of 100.

The parameters of the five MOO methods are listed in Table 6. To ensure all five methods were compared in a fair manner, a structured search was used to optimize the performance metrics using a related problem, namely the two-objective Canberra system simulated between 1970 and 1990. For each method, a set of default parameters based on values recommended in the literature was adopted. Then a range of values for each parameter was generated by

perturbing the default values. A combination of tuning parameters was formed by selecting a value for one parameter from the available range while keeping the other parameters at their default values. Table 6 summarizes the adopted parameters for each method. As only a limited number of combinations for each method was explored, there is a distinct possibility that the best set was not identified. Because EMOACO and EMOACO-I had seven parameters for which only 20 combinations were tested, it is more likely that a better set of parameters was found for the non-EMOACO methods. Therefore, the tuning procedure intrinsically favoured better outcomes for the non-EMOACO methods. It is acknowledged that the structured search is premised on the assumption that there is little interaction between parameters. As most of the tuned parameters were at the default literature values, this issue is considered to be of secondary importance.

RESULTS AND DISCUSSION

In this section, the performance of the five MOO methods is evaluated. All methods were applied with the tuned parameters reported in Table 6. Two case studies were undertaken, the first using the Canberra system, and the second using the more complex Sydney system. Because the MOO methods were tuned to the Canberra case study, applying these methods to the Sydney case study will test the capability of these methods under more stringent conditions. The performance measures are reported as the average of 10 runs.

Case study – Canberra water supply system: two objectives

Figure 4(a) shows a plot of the convergence measure for the five MOO methods against a range of function evaluations for the Canberra system minimizing restriction frequency and present worth cost. EMOACO-I unequivocally outperforms the other methods demonstrating very rapid convergence. ϵ MOEA is the best method among non-ACO methods. This graph clearly shows that the ranking of the methods varies as the number of function evaluations changes; for instance, NSGA-II outranks SMPSO and

Table 6 | Summary of parameters used in the five MOO methods

Method	Parameter	Tuned value
EMOACO and EMOACO-I	Number of ants	1
	ρ	0.02
	C	10
	τ_0	20
	K_{max}	5
	P_{adj}	0.05
	q_0	0.1
SMPSO	Swarm size	100
	Archive size	100
	P_{Mutation}	0.005
NSGA-II	$P_{\text{Crossover}}$	0.9
	P_{Mutation}	0.005
	Population	50
ϵ MOEA	$P_{\text{Crossover}}$	1.0
	P_{Mutation}	0.01
	$P_{\text{Inversion}}$	0.005

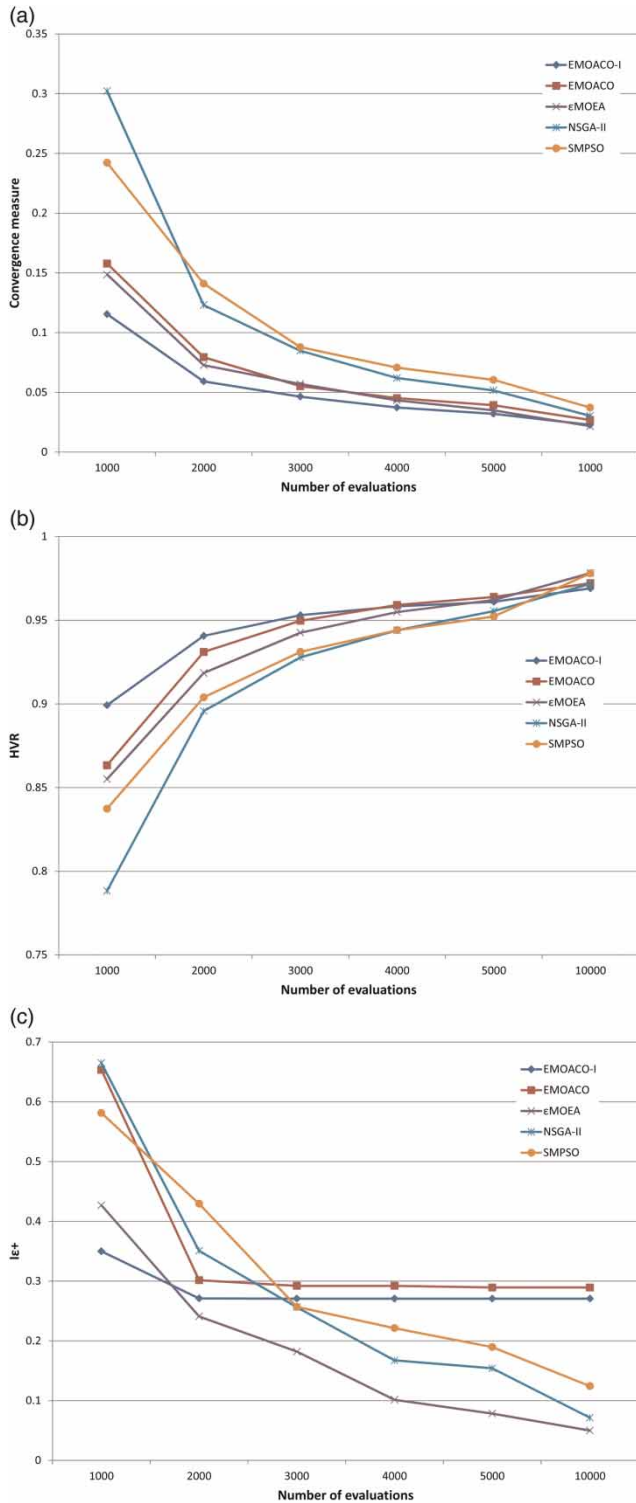


Figure 4 | Comparison of measures for five MOO methods as a function of the number of evaluations for the Canberra case study minimizing two objectives: present worth cost and restriction frequency. (a) Convergence, (b) HVR, (c) $I_{\epsilon+}$.

MOACO-State after 2,000 evaluations. Figure 4(b) presents a similar plot for the HVR metric. Once again, EMOACO-I outperforms the other methods except when the number of evaluations is 5,000. EMOACO and ϵ MOEA are ranked as second best. Again this graph shows changes in rankings as the number of evaluations progress; for example, the rank of SMPSO is four after 5,000 evaluations but climbs to two after 10,000 evaluations. Figure 4(c) shows a plot of the $I_{\epsilon+}$ measure for the five MOO methods. This plot shows even more variation in the ranking highlighting the sensitivity of this measure. Except for 1,000 evaluations, ϵ MOEA is the best method. Interestingly, all non-ACO methods continue to improve as the number of evaluations increase.

Case study – Canberra water supply system: three objectives

The convergence, HVR and $I_{\epsilon+}$ metrics for the five MOO methods optimizing the Canberra headworks system with three objectives are presented in Figure 5. EMOACO-I has the best convergence metrics and ϵ MOEA has the best convergence among benchmark MOO methods. However, in contrast to Figure 2(a), Figure 5(a) shows greater variability between methods – even after 10,000 evaluations there remain significant differences in the convergence measure. The addition of the third objective, to which the MOO parameters were not tuned, has made the optimization more challenging. With regard to the HVR metric, Figure 5(b) shows that EMOACO-I is superior up to 3,000 evaluations but is then marginally overtaken by all methods. It is noted in Figure 5(c) that ϵ MOEA is clearly the best of all methods in terms of $I_{\epsilon+}$ with EMOACO-I unambiguously ranked second.

Case study – Sydney water supply system: two and three objectives

The MOO parameters, presented in Table 3, were tuned to the Canberra system. In the Sydney case study, the performance of the MOO methods is evaluated without any further tuning.

Figure 6 presents the convergence, HVR and $I_{\epsilon+}$ measures for the five MOO methods for the Sydney case study minimizing two objectives, present worth cost and

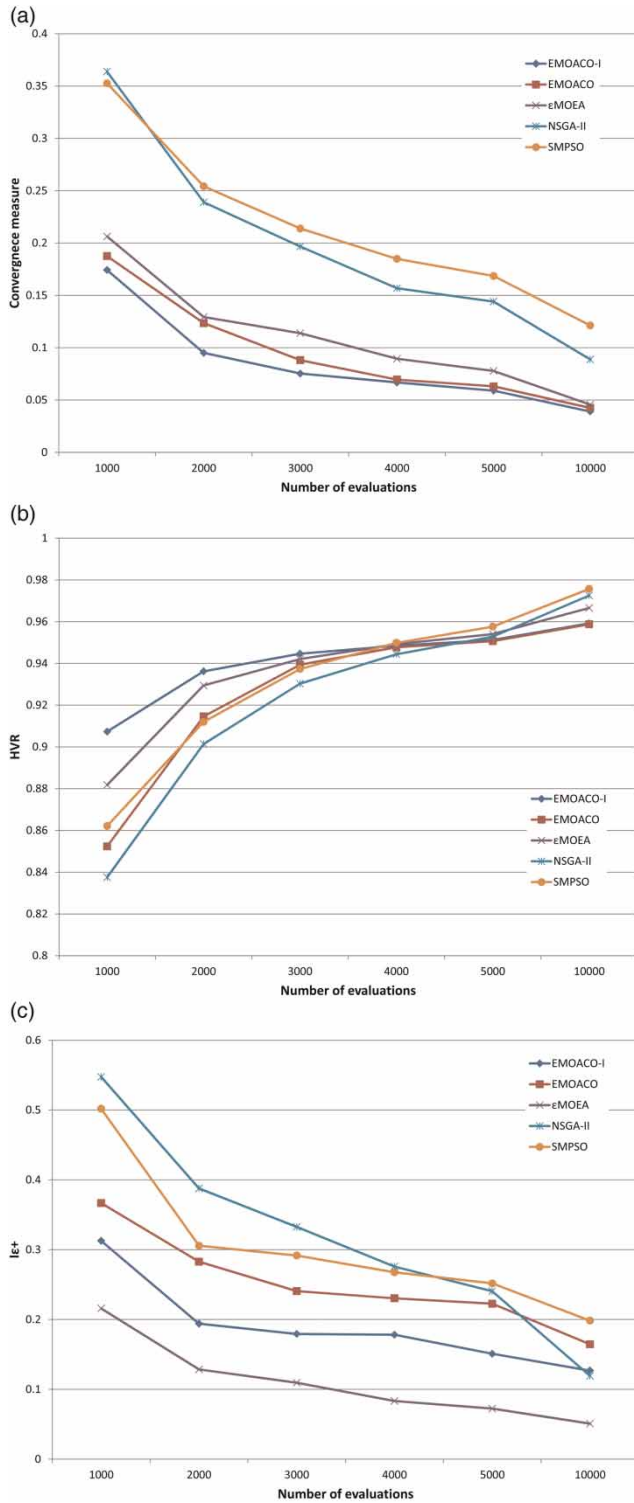


Figure 5 | Comparison of measures for five MOO methods as a function of the number of evaluations for the Canberra case study minimizing three objectives: present worth cost, restriction frequency and time storage less than 20%. (a) Convergence, (b) HVR, (c) $I_{\epsilon+}$.

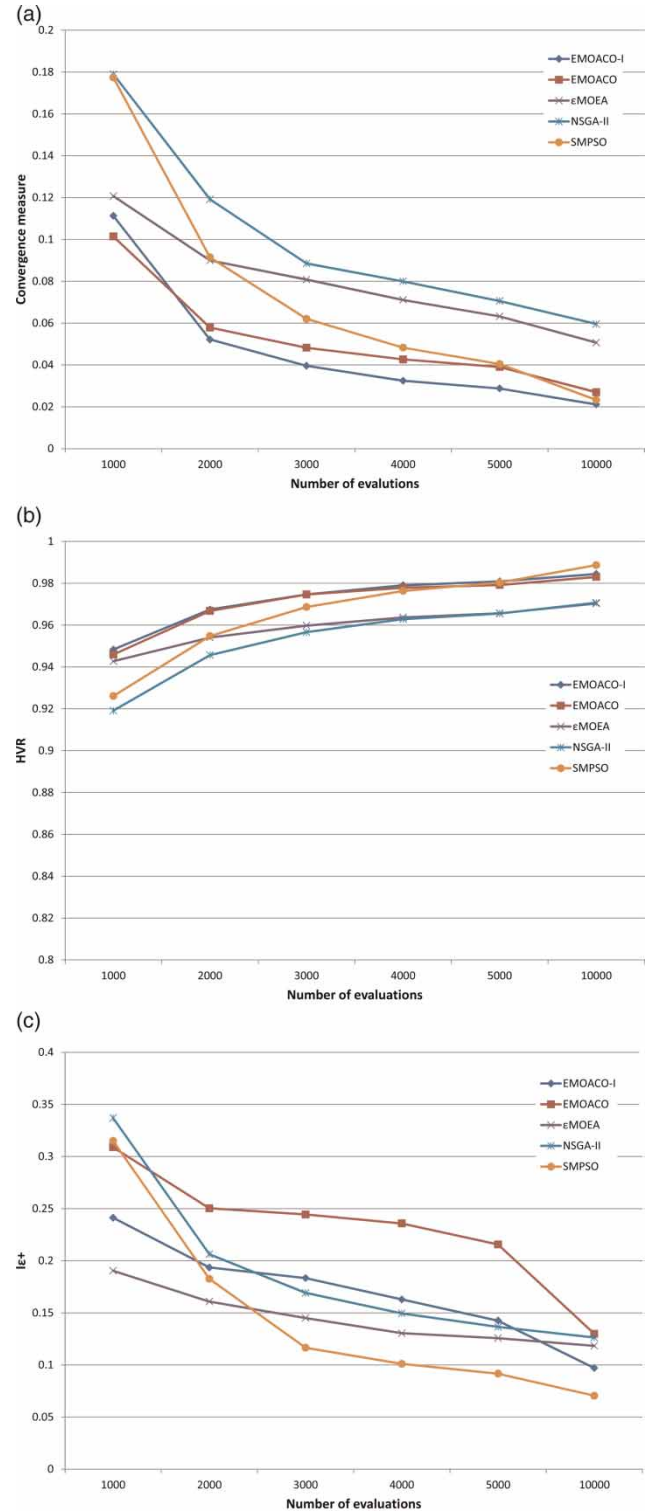


Figure 6 | Comparison of measures for five MOO methods as a function of the number of evaluations for the Sydney case study minimizing two objectives: present worth cost and restriction frequency. (a) Convergence, (b) HVR, (c) $I_{\epsilon+}$.

restriction frequency. EMOACO-I has the best convergence measure except for 1,000 evaluations. With the exception of 1,000 evaluations, SMPSO is the best method among benchmark methods in terms of convergence. EMOACO-I outperforms other methods in terms of the HVR measure. SMPSO is the best method among benchmark methods except for 1,000 evaluations. Although Figure 6(c) indicates good $I_{\epsilon+}$ performance of ϵ MOEA for 1,000 and 2,000 evaluations, SMPSO overtakes ϵ MOEA after 2,000 evaluations. It also shows EMOACO-I is competitive with these methods with respect to $I_{\epsilon+}$.

The Sydney three-objective case study reveals a significant divergence in performance among the methods. Figure 7 presents the convergence, HVR and $I_{\epsilon+}$ measures for the five MOO methods for the Sydney case study. EMOACO and EMOACO-I are significantly superior to the other methods using the convergence measure, but EMOACO-I did not perform as well for the other metrics after 1,000 evaluations. ϵ MOEA is the best of the benchmark methods. For the HVR measure, EMOACO-I is clearly superior up to 2,000 evaluations, after which SMPSO and ϵ MOEA marginally overtake it.

Comparison of all five MOO methods in terms of the three measures for the four case study combinations reveals that no one method unambiguously outperforms the other methods. For instance, while EMOACO-I performed well in terms of convergence and HVR measure for the two-objective Canberra case, it did not perform as well in terms of $I_{\epsilon+}$. Similarly, ϵ MOEA is the best method among benchmark methods for three cases while SMPSO has better convergence performance for the two-objective Sydney case.

It is noted that NSGA-II was ranked last among the benchmark methods in most of the cases and was particularly poor in the three-objective cases with respect to the convergence measure.

CONCLUSIONS

The optimization of water resource systems in the presence of conflicting objectives necessitates the use of multi-objective optimization methods. Modern-day MOO methods based on probabilistic methods require many thousands of

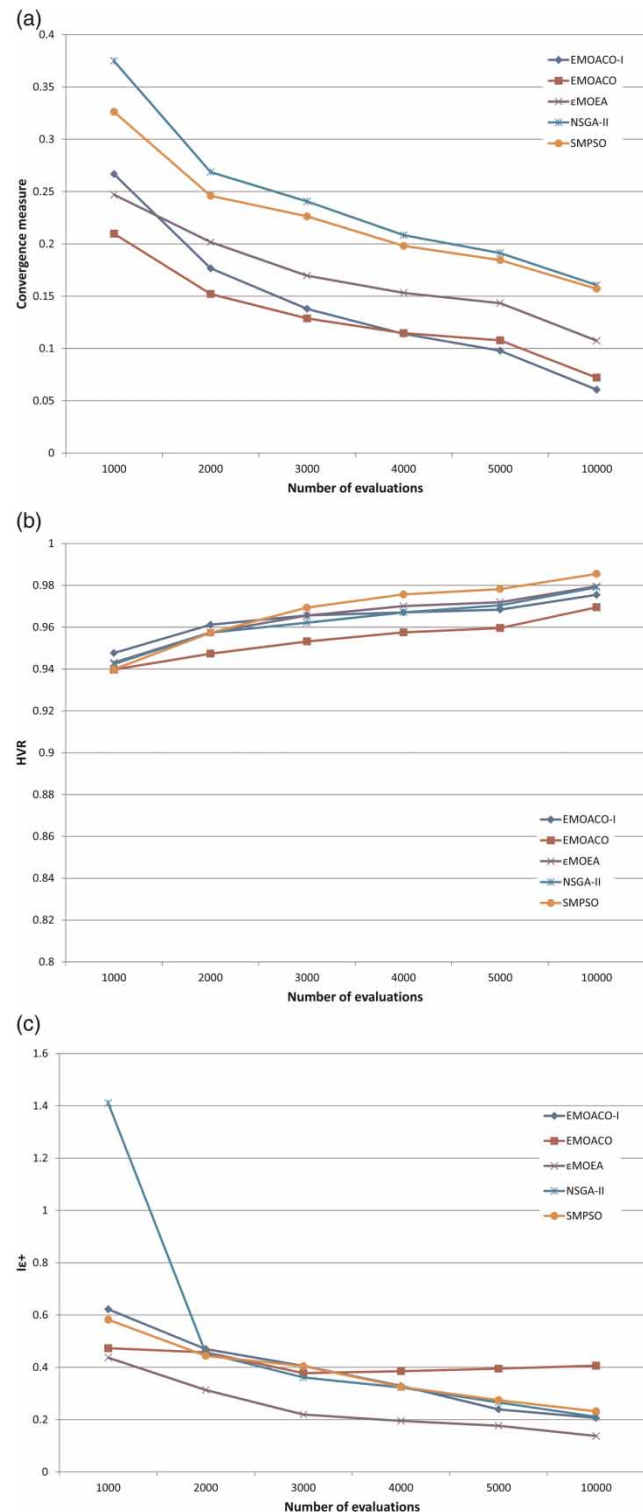


Figure 7 | Comparison of measures for five MOO methods as a function of the number of evaluations for the Sydney case study minimizing three objectives: present worth cost, restriction frequency and environmental stress. (a) Convergence, (b) HVR, (c) $I_{\epsilon+}$.

objective function evaluations. Unfortunately, these evaluations typically require running simulation models, which for complex water resource systems can be computationally very expensive. Therefore there is a strong practical need for MOO methods that converge quickly while maintaining diversity along the Pareto front. Recently, a number of studies, mainly using evolutionary algorithms and PSO methods, have focussed on MOO methods that converge more quickly. This study sought to identify which of these methods is best suited to urban water management applications. It identified three benchmark MOO methods, NSGA-II, ϵ MOEA and SMPSO.

The good performance of ACO in finding optimal solutions in single objective problems motivated investigation of its potential in multi-objective optimization. As most multi-objective implementations of ACO have focussed on combinatorial problems, the approach taken was to adapt existing ACO methods to develop a MOACO algorithm suitable for urban water resource applications.

A review of the MOACO literature identified a number of shortcomings, the main one being their problem-specific implementation. Based on this review, an algorithm called EMOACO was developed incorporating the best features of existing ACO methods whilst avoiding their shortcomings. Important features of EMOACO include the use of a single ant colony with one pheromone matrix, a pheromone updating process independent of the number of objectives and the scale of objective function values, and the elimination of heuristic problem-specific information.

Two concepts borrowed from evolutionary search methods, namely adjacency and random selection, were implemented in the EMOACO framework. Adjacency exploits the proposition that potentially good solutions lie in the neighbourhood of current non-dominated solutions. Random selection allows ants to visit routes with low pheromone. Furthermore, the use of a simple heuristic to reduce the number of decision combinations in the initial phase of EMOACO was added to accelerate initial convergence. This method was called EMOACO-I.

Over the range of function evaluations considered in this study, it was found that in most cases EMOACO-I was the best performing method in terms of convergence and HVR with EMOACO ranked second. With respect to the $I_{\epsilon+}$ metric EMOACO-I was the best in one case and

competitive in the other cases. It was observed that the $I_{\epsilon+}$ metric was the most sensitive of the metrics, primarily because it focuses on outliers on the non-dominated solution set. These findings suggest that the EMOACO methods are competitive in applications involving computationally intensive urban water resource problems.

For the non-MOACO methods, ϵ MOEA and SMPSO had comparable performance with NSGA-II ranked behind them. Overall it was found that NSGA-II was the worst performing method.

Of significance was the finding that no method was superior in terms of all measures and for all case study problems. Of particular interest was the greater variability in the performance of the MOO methods when moving from two to three objective problems and from the Canberra case study, for which the MOO parameters were tuned, to the more complex Sydney case study, for which the MOO parameters were not tuned. This suggests that a strategy based on pooling the results from multiple MOO methods could help guard against the vagaries in performance of individual methods.

REFERENCES

- Afshar, A., Sharifi, F. & Jalali, M. R. 2009 [Non-dominated archiving multi-colony ant algorithm for multi-objective optimization: Application to multi-purpose reservoir operation](#). *Eng. Optim.* **41**, 313–325.
- Alaya, I., Solnon, C. & Ghedira, K. 2007 Ant colony optimization for multi-objective optimization problems. 19th IEEE International Conference on Tools with Artificial Intelligence, 1, IEEE Computer Society Press, Los Alamitos, 450–457.
- Angus, D. 2007a Population-based ant colony optimisation for multi-objective function optimisation. *Lect. Notes Artif. Int.* **4828**, 232–244.
- Angus, D. 2007b Crowding population-based ant colony optimisation for the multi-objective travelling salesman problem. IEEE Symposium on Computational Intelligence in Multicriteria Decision Making. MCDM2007, 333–340.
- Angus, D. & Woodward, C. 2009 [Multiple objective ant colony optimisation](#). *Swarm Intell.* **3**, 69–85.
- Baran, B. & Schaerer, M. 2003 A multiobjective ant colony system for vehicle routing problem with time windows. *Proceedings of the 21st IASTED International Conference*, Innsbruck, Austria.
- Bui, L. T., Whitacre, J. M. & Abbass, H. A. 2008 Performance analysis of elitism in multi-objective ant colony optimization

- algorithms. IEEE World Congress on Computational Intelligence. CEC2008, Hong Kong, 1633–1640.
- Cardoso, P., Jesús, M. & Márquez, A. 2003 MONACO – Multi-Objective Network optimisation based on an ACO. X Encuentros de Geometría Computacional Conference, Seville, Spain.
- Chang, L.-C. & Chang, F.-J. 2009 Multi-objective evolutionary algorithm for operating parallel reservoir system. *J. Hydrol.* **377**, 12–20.
- Coello Coello, C. A. 2006 Evolutionary multi-objective optimization: A historical view of the field. *Comput. Intell. IEEE* **1**, 28–36.
- Coello Coello, C. A., Lamont, G. B. & Veldhuizen, D. A. V. 2007 *Evolutionary Algorithms for Solving Multi-Objective Problem*. Springer, New York.
- Consoli, S., Matarazzo, B. & Pappalardo, N. 2008 Operating rules of an irrigation purposes reservoir using multi-objective optimization. *Water Resour. Manag.* **22**, 551–564.
- Cui, L. & Kuczera, G. 2005 Optimizing water supply headworks operating rules under stochastic inputs: Assessment of genetic algorithm performance. *Water Resour. Res.* **41**, W05016.
- Czyzak, P. & Jaszkiwicz, A. 1998 Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization. *J. Multi-Criteria Decis. Anal.* **7** (1), 34–47.
- Deb, K. 2001 *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York.
- Deb, K. & Jain, S. 2002 Running Performance Metrics for Evolutionary Multi-Objective Optimization. Indian Institute of Technology, KanGAL Report No. 2002004, Kanpur, India.
- Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002a A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197.
- Deb, K., Thiele, L., Laumanns, M. & Zitzler, E. 2002b Scalable multi-objective optimization test problems. The 2002 Congress on Evolutionary Computation. The 2002 Congress on Evolutionary Computation (CEC), 825–830.
- Doerner, K. & Gutjahr, W. J. 2004 Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection. *Ann. Operat. Res.* **131**, 79–99.
- Doerner, K. F., Hartl, R. F. & Reimann, M. 2003 Are COMPETants more competent for problem solving? The case of a multiple objective transportation problem. *Eur. J. Oper. Res.* **11**, 115–141.
- Dorigo, M., Maniezzo, V. & Colormi, A. 1991 *Positive Feedback as a Search Strategy*. Dipartimento di Elettronica, Politecnico di Milano, Milan.
- Dorigo, M., Maniezzo, V. & Colormi, A. 1996 Ant system: optimization by a colony of cooperating agents. *IEEE T. Syst. Man Cy. B.* **26**, 29–41.
- Dorigo, M. & Stützle, T. 2004 *Ant Colony Optimization*. The MIT Press, Cambridge, MA.
- Durillo, J. J. & Nebro, A. J. 2011 jMetal: A Java framework for multi-objective optimization. *Adv. Eng. Softw.* **42**, 760–771.
- Durillo, J. J., Nebro, A. J., Luna, F., Coello Coello, C. A. & Alba, E. 2010 Convergence speed in multi-objective metaheuristics: Efficiency criteria and empirical study. *Int. J. Numer. Meth. Eng.* **84**, 1344–1375.
- Eskandari, H., Geiger, C. & Lamont, G. 2007 FastPGA: a dynamic population sizing approach for solving expensive multiobjective optimization problems. *Evol. Multi-Criterion Optim.* **4403**, 141–155.
- Farmani, R., Godfrey, W. & Savic, D. A. 2006 Evolutionary multi-objective optimization of the design and operation of water distribution network: total cost vs. reliability vs. water quality. *J. Hydroinform.* **8**, 165–179.
- Fonseca, C. M., Paquete, L. & López-Ibañez, M. 2006 An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. IEEE Congress on Computation. CEC 2006. Vancouver, BC Canada, 1157–1163.
- García-Martínez, C. 2004 *An Empirical Analysis of Multiple Objective Ant Colony Optimization Algorithms for the Bi-criteria TSP*. ANTS 2004, LNCS 3172. Springer-Verlag, Berlin/Heidelberg.
- Gravel, M., Price, W. L. & Gagné, C. 2002 Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *Eur. J. Oper. Res.* **143**, 218–229.
- Hadka, D. & Reed, P. 2011 Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evol. Comput.* **20**, 1–31.
- Huang, V. L., Suganthan, P. N. & Liang, J. J. 2006 Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. *Int. J. Intell. Syst.* **21**, 209–226.
- Iredi, S. 2001 Bi-Criterion Optimization with Multi Colony Ant Algorithms. The First International Conference on Evolutionary Multi-Criterion Optimization. Springer-Verlag, Berlin/Heidelberg.
- Jefferson, C. E., Holz, L. M., Hardy, M. J. & Kuczera, G. 2005 Integrated Urban Water Management: Combining Multi-Criterion Optimization and Decision Analysis. ASCE. International Conference on Computing in Civil Engineering, Cancun, Mexico.
- Kennedy, J. & Eberhart, R. 1995 Particle swarm optimization. *Proceeding IEEE International Conference on Neural Networks*. Piscataway, NJ, 4, 1942–1948.
- Kim, T., Heo, J.-H., Bae, D.-H. & Kim, J.-H. 2008 Single-reservoir operating rules for a year using multiobjective genetic algorithm. *J. Hydroinform.* **10**, 163–179.
- Ko, S.-K., Fontane, D. G. & Labadie, J. W. 1992 Multiobjective optimization of reservoir systems operation. *J. Am. Water Resour. Assoc.* **28**, 111–127.
- Kollat, J. B. & Reed, P. M. 2007 A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications. *Adv. Water Resour.* **30**, 408–419.
- Labadie, J. W. 2004 Optimal operation of multireservoir systems: state-of-the-art review. *J. Water Res. Pl.-ASCE* **130**, 93–111.
- Laumanns, M., Thiele, L., Deb, K. & Zitzler, E. 2002 Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.* **10**, 263–282.

- Liang, Q., Johnson, L. E. & Yu, Y.-S. 1996 A comparison of two methods for multiobjective optimization for reservoir operation. *J. Am. Water Resour. Assoc.* **32**, 333–340.
- López-Ibáñez, M., Paquete, L. & Stützle, T. 2004 On the design of ACO for the biobjective quadratic assignment problem. *Lect. Notes Comput. Sci.* **3172**, 214–225.
- Martínez, C. G., Cordon, O. & Herrera, F. 2007 A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *Eur. J. Oper. Res.* **180**, 116–148.
- McMullen, P. R. 2001 An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artif. Intell. Eng.* **15**, 309–317.
- Mortazavi, M., Kuczera, G. & Cui, L. 2012 Multiobjective optimization of urban water resources: Moving toward more practical solutions. *Water Resour. Res.* **48**, W03514.
- Mortazavi-Naeini, M. 2013 *Multi-objective Optimization of Urban Water Resource Systems*. PhD, University of Newcastle.
- Mortazavi-Naeini, M., Cui, L.-J. & Kuczera, G. 2009 *Application of Multiobjective Optimization Methods for Urban Water Management: A Case Study for Canberra Water Supply System*. H200. Newcastle, Australia.
- Nebro, A., Durillo, J., CoelloCoello, C. A., Luna, F. & Alba, E. 2008 A study of convergence speed in multi-objective metaheuristics. *Parallel Problem Solving from Nature – PPSN X* **5199**, 763–772.
- Pierro, F., Khu, S.-T., Savic, D. & Berardi, L. 2009 Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. *Environ. Modell. Softw.* **24**, 202–213.
- Reyes-Sierra, M. & CoelloCoello, C. A. 2006 Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *IJCIR* **2**, 287–308.
- Santana-Quintero, L., Ramírez-Santiago, N., Coello, C., Luque, J. & Hernández-Díaz, A. 2006 A new proposal for multiobjective optimization using particle swarm optimization and rough sets theory. *Parallel Problem Solving from Nature – PPSN IX* **4193**, 483–492.
- Savic, D., Walters, G. & Schwab, M. 1997 Multiobjective genetic algorithms for pump scheduling in water supply. *Evol. Comput.* **1305**, 227–235.
- Scott, A. & Grant, T. 1997 *Impacts of Water Management in the Murray-Darling Basin on the Platypus (Ornithorhynchusanatinus) and the Water Rat (Hydromyschrysgaster)*. CSIRO Land and Water, Canberra.
- Shelokar, P. S., Jayaraman, V. K. & Kulkarni, B. D. 2002 Ant algorithm for single and multiobjective reliability optimization problems. *Qual. Reliab. Eng. Int.* **18**, 497–514.
- Shiau, J.-T. 2009 Optimization of reservoir hedging rules using multiobjective genetic algorithm. *J. Water Res. Pl.-ASCE* **135**, 355–365.
- Srinivas, N. & Deb, K. 1994 Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **2**, 221–248.
- Stützle, T. & Hoos, H. H. 2000 MAX-MIN ant system. *Future Gener. Comp. Sy.* **16**, 889–914.
- Stützle, T., López-Ibáñez, M., Dorigo, M., Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P. & Smith, J. C. 2010 A Concise Overview of Applications of Ant Colony Optimization. *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc., New York.
- T'Kindt, V., Monmarché, N., Tercinet, F. & Lüigt, D. 2002 An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *Eur. J. Oper. Res.* **142**, 250–257.
- Toscano-Pulido, G., Coello Coello, C. & Santana-Quintero, L. 2007 EMOPSO: A multi-objective particle swarm optimizer with emphasis on efficiency. *Evol. Multi-Criterion Optim.* **4403**, 272–285.
- Van Veldhuizen, D. A. & Lamont, G. B. 2000 On measuring multiobjective evolutionary algorithm performance. *Evolutionary Computation* **1**, 204–211.
- Zitzler, E., Deb, K. & Thiele, L. 2000 Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **8**, 173–195.
- Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C. M. & Grunert da Fonseca, V. 2002 *Why Quality Assessment Of Multiobjective Optimizers Is Difficult*. GECCO 2002. Morgan Kaufmann Publishers, Burlington, MA, 666–673.

First received 19 September 2013; accepted in revised form 9 June 2014. Available online 5 July 2014