Extension of the constrained ant colony optimization algorithms for the optimal operation of multi-reservoir systems

R. Moeini and M. H. Afshar

ABSTRACT

This paper extends the application of Constrained Ant Colony Optimization Algorithms (CACOAs) to optimal operation of multi-reservoir systems. Three different formulations of the constrained Ant Colony Optimization (ACO) are outlined here using Max-Min Ant System for the solution of multi-reservoir operation problems. In the first two versions, called Partially Constrained ACO algorithms, the constraints of the multi-reservoir operation problems are satisfied partially. In the third formulation, all the constraints of the underlying problem are implicitly satisfied by the provision of tabu lists to the ants which contain only feasible options. The ants are, therefore, forced to construct feasible solutions and hence the method is referred to as a Fully Constrained ACO algorithm. The proposed constrained ACO algorithms are formulated for both possible cases of taking storage/release volumes as the decision variables of the problem. The proposed methods are used to optimally solve the well-known problems of four- and ten-reservoir operations and the results are presented and compared with those of the conventional unconstrained ACO algorithm and existing methods in the literature. The results indicate the superiority of the proposed methods over conventional ACOs and existing methods to optimally solve large scale multi-reservoir operation problems.

Key words | Ant Colony Optimization algorithm, Max-Min Ant System, multi-reservoir operation, partially and fully constrained

R. Moeini (corresponding author)
School of Civil Engineering,
Iran University of Science and Technology,
P.O. 16765-163,
Narmak, Tehran,
Iran
E-mail: *rmoeini@iust.ac.ir*

M. H. Afshar
School of Civil Engineering & Enviro-hydroinformatic Center of Excellence,
Iran University of Science and Technology,
P.O. 16765-163,
Narmak, Tehran,
Iran

LIST OF NOTATIONS

F	Benefit function
K	Total number of reservoirs
T	Total number of operation periods
$b_k(t)$	Benefit function of reservoir k at period t
$R_k(t)$	Release from reservoir k at period t
$S_k(t)$	Storage at period t in reservoir k
$I_k(t)$	Inflows in period t to reservoir k
$S_k^{\min}$	Minimum storage of reservoir k
$S_k^{\max}$	Maximum storage of reservoir k
$R_k^{\min}$	Minimum release from reservoir k
$R_k^{\max}$	Maximum release from reservoir k
D	Set of decision points
L	Set of options
C	Set of costs associated with options L
η_{ij}	Heuristic value corresponding to option j at decision point i
f	Benefit function
f_p	Penalized benefit function
$CSV_k(t)$	Measure of constraint violation at period t of the reservoir k
α_p	Penalty parameter
$\bar{S}_k^{\min}(t), \bar{S}_k^{\max}(t)$	New bounds calculated for the storage volume of reservoir k at time period t

doi: 10.2166/hydro.2012.081

α The parameter that controls the relative weight of the pheromone trail

β The parameter that controls the relative weight of the heuristic value

P_{best} A specified probability by which each component of the current global-best solution will be selected by ants in the next iteration

ρ Pheromone evaporation factor

INTRODUCTION

Reservoir operation is a complex problem that often involves many decision variables and multiple objectives as well as considerable risk and uncertainty (Oliveira & Loucks 1997). Applying optimization techniques for reservoir operation is not a new idea. Several techniques for optimal reservoir operation have been developed and some of their limitations have been discussed in detail (Yeh 1985). Various techniques have been applied in an attempt to improve the efficiency of reservoir operation. These techniques include Linear Programming (LP), Nonlinear Programming (NLP), Dynamic Programming (DP), and more recently heuristic methods such as Genetic Algorithms (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Shuffled Complex Evolution (SCE), etc.

In reservoir operation, LP is well known as the most favored optimization technique with many advantages. It is easy to understand and does not require any initial solution. A number of examples of applying LP to reservoir operation are provided by Yeh (1985). Some recent applications of LP for reservoir operation problems are illustrated by Kuczera (1989), Srinivasan *et al*. (1999), Tu *et al*. (2003) and most recently by Yoo (2009) who applied LP for hydropower operation.

Most of the real problems, however, are not linear and, therefore, need to be linearized so that LP methods can be used. Application of NLP techniques to multi-reservoir operation is less common due to their limitation of slow rate of convergence, requiring a large amount of computational storage and time compared to other methods (Yeh 1985). More importantly, NLP methods are prone to getting stuck in local optima when the problem is non-convex. However, Peng & Buras (2000) and Cai *et al*. (2002) used NLP in the optimization of multi-reservoir operation systems.

Multi-reservoir systems offer various problems for the optimization methods to handle. One of the main challenges in multi-reservoir operation is the nonlinearities and nonconvexities in the field of hydropower, so the linear and NLP methods cannot be directly used. However, these present no problem to the most commonly used method of DP for reservoir operation. In DP methods, the sequential decision problem is divided into a sequence of separate, but interrelated, single-decision sub-problems. In this way, large and complex problems can be theoretically solved by combining the solutions of smaller problems (sub-problems) to obtain the solution of the entire problem (Mays & Tung 2002). Applications of DP in the water resources area was discussed by Yakowitz (1982) and Yeh (1985) while more recent applications of DP and its variants in reservoir operations were presented by Perera & Conder (1998), Kumar & Baliarsingh (2003), and Mousavi & Karamouz (2003).

Each of these techniques has its own merits and limitations. To overcome those limitations, during the last two decades, heuristic algorithms have been developed for solving reservoir operation problems because of their flexibility and effectiveness for optimizing complex systems.

GA is the most commonly used heuristic algorithms in water resources problems. Esat & Hall (1994) applied a GA to the four-reservoir problem and demonstrated the advantages of GAs over standard DP techniques in terms of computational requirements. Fahmy *et al*. (1994) also applied a GA to a reservoir system, and compared the performance of the GA approach with that of DP. Oliveira & Loucks (1997) proposed an approach to determine reservoir operating rules using GA. Two types of GAs, real-coded and binary-coded, were applied to the optimization of a flood control reservoir model by Chang & Chen (1998). Wardlaw & Sharif (1999) explored the potentials of alternative GA formulations in application to real time reservoir operation. Chen (2003) successfully applied real-coded GA in combination with a simulation model to optimize 10-day operation rule curves of major reservoir systems in Taiwan. Later, a comparison between binary-coded and real-coded GA was explored in optimization of the reservoir operation rule curves (Chang *et al*. 2005). Chen & Chang

(2007) applied a real-coded multi-population GA to multi-reservoir operation.

Recently, other heuristic algorithms have been applied to reservoir operation. Being at its early stages of development, a Honey Bees Mating Optimization (HBMO) heuristic algorithm was applied to a single reservoir operation problem (Bozorg Haddad *et al*. 2006). Jalali (2005), Jalali & Afshar (2005) and Jalali *et al*. (2007) used improved and multi-colony ACO algorithms to solve the multi-reservoir operation problem. Kumar & Reddy (2007) proposed an Elitist-Mutated Particle Swarm Optimization (EMPSO) algorithm for multipurpose reservoir systems and compared its performance to standard PSO by solving the hypothetical four-reservoir system and an existing reservoir system, namely the Bhadra reservoir system, in Karnataka State, India. Bozorg Haddad *et al*. (2008) used a HBMO algorithm for design and operation of hydropower multi-reservoirs. More recently, Madadgar & Afshar (2009) applied an Improved Continuous Ant algorithm for optimization of a single hydropower reservoir.

Heuristic search methods have been shown to be more effective than alternative methods such as NLPs and in particular DPs in locating near optimal solutions. The efficiency of these methods, however, may also deteriorate as the scale of the problem increases. Straightforward application of these methods to large scale problems has been shown to lead to sub-optimal and in some cases to infeasible solutions (Afshar 2007; Afshar & Moeini 2008). Afshar (2007) proposed a constrained version of the ACO algorithm to increase the efficiency of the standard ACO algorithm for its application to highly constrained optimization problems and tested its performance on the sewer network design problem. The method was later used for the single reservoir operation problem by Afshar & Moeini (2008) emphasizing the efficiency of the method for large scale problems.

This paper proposes a family of constrained ACO algorithms for solution of multi-reservoir operation problems. These methods are, in fact, a natural extension of the methods proposed by Afshar & Moeini (2008) for a single reservoir. This extension leads to three constrained versions of the ACO algorithms which are developed and used for the optimization of multi-reservoir operation problems. The proposed methods exploit the unique feature of ant based algorithms, namely the incremental solution building mechanism, to explicitly enforce the constraints of the reservoir operation problems. In the first version named Partially Constrained Ant Colony Optimization Algorithm I (PCACOA1), a tabu list is constructed for each ant at each decision point of the problem to simultaneously satisfy both release and storage volume constraints. This method is shown to be successful in constraint satisfaction except for very rare cases. In the second method named Partially Constrained Ant Colony Optimization Algorithm II (PCACOA2), the storage volume bounds of the upstream reservoirs are modified prior to the main search such that infeasible operation of these reservoirs is not constructed by ants of PCACOA1. This is achieved by sweeping the periods of the operation in a reverse order and modifying the storage volume box constraints of the upstream reservoirs so that violations of corresponding release and storage constraints are always avoided. And finally, in the third algorithm referred to as Fully Constrained Ant Colony Optimization Algorithm (FCACOA), the same process of PCACOA2 is also applied to downstream reservoirs each time a partial solution represented by the upstream reservoirs operation policy is constructed by PCACOA1 and PCACOA2. This method is shown to lead to a search algorithm totally in the feasible region of the search space. The proposed methods are outlined here using Max-Min Ant System (MMAS) for the solution of multi-reservoir operation problems with release/storage volumes taken as the decision variables of the problem. The proposed methods are used to optimally solve the well-known four- and ten-reservoir operation problems and the results are presented and compared with those of the conventional unconstrained ACO algorithm and existing methods. The results indicate the superiority of the proposed methods for the optimal solution of large scale multi-reservoir operation problems to those of the conventional ACO algorithm and other methods used for these problems.

MULTI-RESERVOIR OPERATION PROBLEM

The multi-reservoir operation problem is a complex problem that often involves many decision variables, many constraints and multiple objectives of highly nonlinear and non-convex nature. While nonlinear, non-convex and multi-objective features of the reservoir operation problems can be properly handled by heuristic search methods, the efficiency and effectiveness of these methods can be adversely affected by the scale of these problems. Improving the efficiency and performance of these methods to large scale multi-reservoir operation problems is still an active area of research. A multi-reservoir system contains both parallel and series connections with the condition of the outflow of upstream reservoirs serving as the inflow of downstream reservoirs which adds to the complexity of the problem.

With a view to handling the scale of the multi-reservoir operation problems, a single objective multi-reservoir operation problem of linear type is considered here so that the effectiveness of the proposed methods can be tested by comparing the results with those obtained using the LP method and other available results. The objective of the problem to be considered is to maximize benefits from the system over the operation period that can be written as:

$$F = \sum_{k=1}^{K} \sum_{t=1}^{T} b_k(t) \times R_k(t) \quad (1)$$

where, F is the benefit function; K is the total number of reservoirs; T is the total number of operation periods; $b_k(t)$ is the benefit function of reservoir k at period t; and $R_k(t)$ is the release from reservoir k at period t.

The fundamental continuity constraint and constraints on reservoir storage and on release over each operating period t are defined as:

$$S_k(t+1) = S_k(t) + I_k(t) - R_k(t) \quad (2)$$

$$S_k^{\min} \le S_k(t) \le S_k^{\max} \quad (3)$$

$$R_k^{\min} \le R_k(t) \le R_k^{\max} \quad (4)$$

where, $S_k(t)$ is the storage at time period t in reservoir k; $I_k(t)$ is the inflows in time period t to reservoir k; $R_k(t)$ is the release in time period t from reservoir k; $S_k^{\min}$ is the minimum storage of reservoir k; $S_k^{\max}$ is the maximum storage of reservoir k; $R_k^{\min}$ is the minimum release from reservoir k; and $R_k^{\max}$ is the maximum release from reservoir k. The above constraints apply in all periods $t = 1,\ldots, T$.

FORMULATION OF THE MULTI-RESERVOIR OPERATION USING ACOAS

Application of an ACO algorithm to any combinatorial optimization problem is best described by projecting the problem on a graph (Dorigo & Gambardella 1997). Consider a graph $\boldsymbol{G} = (\boldsymbol{D}, \boldsymbol{L}, \boldsymbol{C})$ in which $D = \{d_1, d_2, \ldots\ldots, d_n\}$ is the set of decision points at which some decisions are to be made, $L = \{l_{ij}\}$is the set of options j $(j = 1,2,\ldots\ldots,J_i)$ at each of the decision points i $(i = 1,2,\ldots\ldots,n)$ and finally $\mathbf{C} = \{c_{ij}\}$ is the set of costs associated with options $L = \{l_{ij}\}$. A solution φ, termed a path in the ACO algorithm, is formed by the selection of an option at each decision point.

Formulation of the optimal operation of reservoirs as an optimization problem requires the selection of decision variables. Basically two different sets of decision variables can be sought in reservoir operation problems namely release or storage volumes. However, the problem graph is very much dependent on the decision variables selected for the problem. In this paper both release and storage volumes are separately selected as decision variables of the problems and proper formulations are defined.

Once the decision variables of the problem are selected, the problem graph can be easily defined for the application of standard ACO algorithms referred to here as Unconstrained Ant Colony Optimization Algorithm (UACOA). Assuming that the initial storage volume of all reservoirs is known, each period of the operation for each reservoir is considered as the decision point of the problem leading to a total number of K*T decision points. With the storage/release volumes taken as decision variables of the operation problem, these decision points will then correspond to end-of-the-period storage/within-the-period release volumes, respectively. The options available at each decision point are then represented by the storage/release volumes defined in the range $[S_k^{\min}, S_k^{\max}]/[R_k^{\min}, R_k^{\max}]$. These options $\boldsymbol{L}\{\mathbf{l_{ij}}\}$ can then be represented by a set of look-up tables constructed for each decision point in which l_{ij} represents the jth component of the look-up table constructed for decision point i of the graph. The graph representation of the problem for the application of UACOA is shown in Figure 1, while Figure 2 shows the base graph of UACOA for an arbitrary ant. Here the solid circles represent the available options, storage/release volumes depending on the type of decision variables selected, at each decision point; dashed paths represent potential solutions and the solid path represents a trial solution on the graph constructed by an arbitrary ant. A known mapping of the type $i = \mathrm{Map}(k,t)$ is assumed when using these graphs for the application of UACOA. This mapping can be arbitrary in the sense that the decision point i can be arbitrarily associated to the period t of reservoir k. Furthermore, this mapping can be different from one ant to the other and in particular can change from iteration to iteration without affecting the application of UACOA defined earlier. This is because the ants' decision at an arbitrary decision point is made independent from the decisions already made at previous decision points. Furthermore, the base graph of the problem is seen to be the same as the base graph of an arbitrary ant in this formulation. This point is made here to emphasize the fact that the search spaces of all ants are the same and equal to the search space of the whole problem. With the formulation just outlined, each ant is now required to incrementally build a solution, operation policy, by selecting an option at each decision point before moving to the next decision point.

A further note should also be added here regarding the heuristic information η_{ij} which reflects the local benefit of choosing option j at decision point i. When release volumes are taken as decision variables of the problem, this benefit

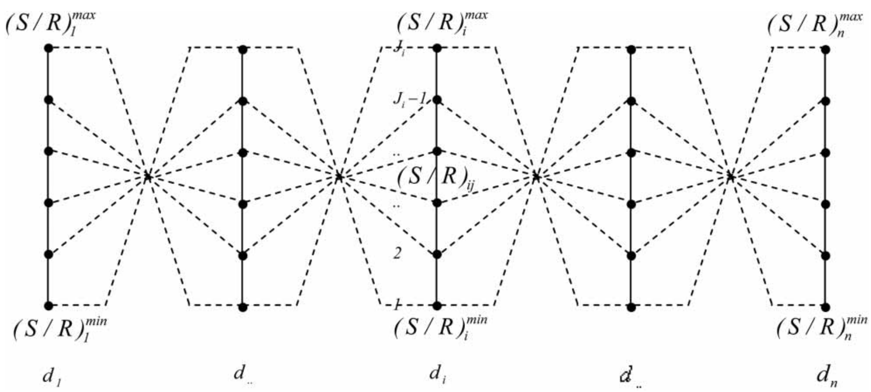

Figure 1 | Base graph of UACOA with storage/release volume as decision variable.

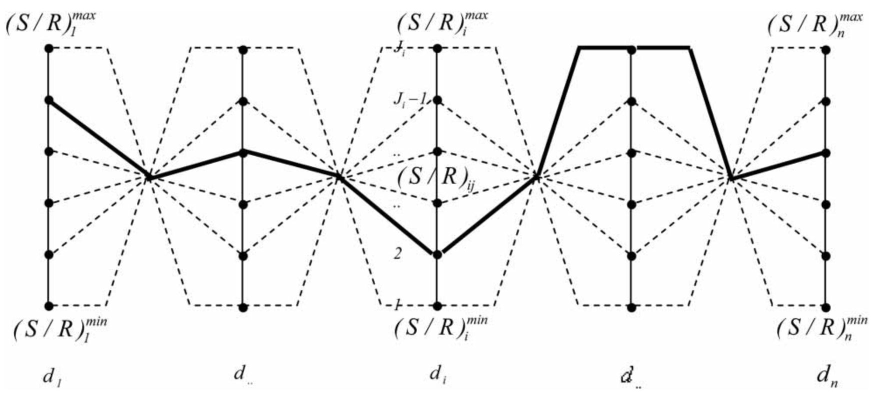

Figure 2 | Base graph of UACOA for an arbitrary ant with storage/release volume as decision variable.

can be defined as:

$$\eta_{ij} = b_k(t) * R_k(t) \tag{5}$$

where, η_{ij} is the heuristic value corresponding to option j at decision point i, $b_k(t)$ is the benefit function of reservoir k at period t corresponding to the ith decision point defined by the mapping outlined earlier; $R_k(t)$ is the release from reservoir k at period t represented by l_{ij}, jth option of the ith decision point, chosen by ant. However, with the storage volumes taken as decision variables of the problem, definition of heuristic information is not possible since it requires the value of the release volume which is not known. Therefore, the heuristic values are indiscriminately set equal to one in this case for all the options at all decision points of the problem.

This formulation, however, may lead to trial solutions that violate some of the problem constraints. To be specific, the resulting operation policies defined by the constructed storage/release volumes may violate some or all of the constraints defined by Equations (4)/(3). To discourage the ants from making decisions (i.e. select storage/releases volumes) which may constitute an infeasible solution, a lower benefit is associated with the solutions that violate the corresponding constraints of the problem. This may be achieved via the use of a penalty method in which the total benefit of the problem is considered as the difference between the problem's benefit and a penalty value as:

$$f_{\mathrm{p}} = f - \alpha_{\mathrm{p}} \times \sum_{k=1}^{K} \sum_{t=1}^{T} \mathrm{CSV}_k(t) \ \text{ if } f_{\mathrm{p}} \geq 0.0 \tag{6}$$

$$f_p = 0.0 \ \text{otherwise}$$

with

$$\mathrm{CSV}_k(t) = \left(1 - \frac{R_k(t)}{R_k^{\min}}\right) + \left(\frac{R_k(t)}{R_k^{\max}} - 1\right) + \left(1 - \frac{S_k(t)}{S_k^{\min}}\right) + \left(\frac{S_k(t)}{S_k^{\max}} - 1\right)$$

where, f_{p} is the penalized benefit function; f is the benefit function F defined in Equation (1); $\mathrm{CSV}_k(t)$ is a normalized measure of violation from constraints (3) and (4) at period t of the reservoir k; and α_{p} represents the penalty parameter with a large enough value such that any infeasible solution has a value smaller than any feasible solution. The proper value of the penalty parameter is decided via a trial and error process before the main calculation. The total cost is assumed non-negative, as indicated by Equation (6), to avoid negative pheromone laying by the ants. The negative or zero value of each term in the parenthesis used for the definition of the $\mathrm{CSV}_k(t)$ means that the corresponding constraint is satisfied, therefore, only the positive values of the term in the parenthesis are used for the calculation of $\mathrm{CSV}_k(t)$.

PROPOSED CONSTRAINED ACO ALGORITHMS FOR MULTI-RESERVOIR OPERATION

ACO algorithms enjoy a unique feature, namely incremental solution building capability, which is not observed in other evolutionary search methods such as GA, currently used for the optimization of engineering problems. This capability is reflected in the process of solution building by ants in which each ant is required to choose an option out of the options available at a decision point of the problem. This is very useful in solving optimization problems of sequential nature such as reservoir operation problems considered here. The constraints of these problems are of explicit nature and, therefore, can be explicitly enforced by limiting the ant's available options to feasible ones via a tabu list. Thereby, the ants can be forced to move in the feasible search space and the use of penalty method as used in Equation (6) for constraint satisfaction can be avoided. The advantages of this process are twofold. The search space size of the problem could be greatly reduced depending on the characteristics of the problem and its constraints. This may in turn lead to better solutions and more importantly to improved convergence characteristics of the algorithm. This idea has already been used in the context of storm sewer network optimization by Afshar (2007) and single reservoir operation by Afshar & Moeini (2008) which is now extended for multi-reservoir operation in this work.

In the proposed Constrained Ant Colony Optimization Algorithms (CACOAs), an attempt is made to satisfy the constraints of the problem as much as possible. The constraints of the multi-reservoir operation problems as defined in Equations (3) and (4) for the sample problem, however, are more complex than the sewer network and single reservoir operation problems. This is particularly true for the storage/release volume constraints of the downstream reservoirs which are affected not only by their operation but also by the operation of the upstream reservoirs. The attempt made in this paper to overcome this complexity has led to development of three constrained ACO algorithms, each of which is responsible for handling some of the problems encountered in the satisfaction of problem constraints.

In the first version, only local operation of each reservoir is designed such that the corresponding constraints are satisfied. However, the method fails in some cases and hence is referred to as PCACOA1. In the second version, the whole operation of each reservoir over the operation period is designed separately such that the corresponding constraints are satisfied. This method works perfectly for the upstream reservoirs but could fail in some rare cases for the downstream reservoirs and is therefore named PCACOA2. In the third and final version, the shortcoming of PCACOA2 is removed by considering the effect of the upstream reservoirs on the downstream reservoirs' operation policy for constraint satisfaction. This method leads to a search algorithm totally in the feasible region of the search space and, hence, called FCACOA.

PARTIALLY CONSTRAINED ANT COLONY OPTIMIZATION ALGORITHM I (PCACOA1)

Consider an ant at an arbitrary decision point i (corresponding to operation period t of the kth reservoir) with a known value for the corresponding decision variable, i.e. storage/release volume. The continuity equation can now be used to calculate a new set of bounds for the decision variable, storage/release volume, of the next $(i+1)$th decision point corresponding to the period $t+1$ of reservoir k, such that the remaining constraints (4)/(3) are fully satisfied by the resulting release/storage volume. These bounds are then used to construct a tabu list for the ant at decision point $i+1$ containing only feasible options of the look-up table $\boldsymbol{L_i} = \{l_{ij}\};\ j = 1,2,\ldots,J_i$ constructed before.

To clarify this, assume a known value of $S_k(t)$ for storage volume of reservoir k at the beginning of the period t. With the release taken as the decision variable, the continuity Equation (2) can be used to substitute $S_k(t+1)$ into storage constraint (3) written at $t+1$ to yield the following constraints for the release of the reservoir k at period t,

$$S_k(t) + I_k(t) - S_k^{\max} \leq R_k(t) \leq S_K(t) + I_k(t) - S_k^{\min} \tag{7}$$

Combining Equation (7) with the original box constraints for the release volume, Equation (4), leads to the following constraints to be met by the release of the reservoir k at period t so that the resulting storage volume $S_k(t+1)$ is feasible.

$$\text{Max}\,(R_k^{\min}, S_k(t) + I_k(t) - S_k^{\max}) \leq R_k(t) \leq \text{Min}\,(R_k^{\max}, S_k(t) + I_k(t) - S_k^{\min}) \tag{8}$$

An analogous procedure can be used to arrive at the following constraints for the storage volume at the end of the period when storage volume is selected as the decision variable.

$$\text{Max}\,(S_k^{\min}, S_k(t) + I_k(t) - R_k^{\max}) \leq S_k(t+1) \leq \text{Min}\,(S_k^{\max}, S_k(t) + I_k(t) - R_k^{\min}) \tag{9}$$

A tabu list can now be constructed for the ant currently at decision point i corresponding to the period t of reservoir k, which contains only those elements of the corresponding look-up table $\boldsymbol{L_i} = \{l_{ij}\};\ j = 1,2,\ldots,J_i$ satisfying constraint (8) or (9) depending on the decision variables selected. This procedure can be repeated in turn for the next decision point until a complete solution is constructed. Note that the solutions so created will automatically satisfy constraints (3) and (4) except for some rare cases to be addressed later.

The graph representation of the problem for the application of PCACOA1 with explicit enforcement of the constraints (3) and (4) is shown in Figure 3 for a typical

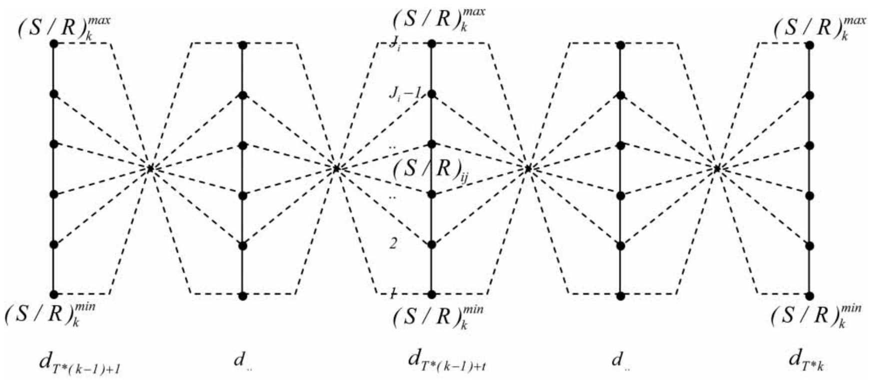

Figure 3 | Base graph of PCACOA1 for reservoir *k* with storage/release volume as decision variable.

reservoir. The graph representing the total search space of the problem is seen to be generally the same as the graph of the problem for the application of UACOA defined before. This, however, is not true for the graph of PCACOA1 for an arbitrary ant, shown in Figure 4, which is totally different from the graph of UACOA for an arbitrary ant (Figure 2). Here the circles at each decision point represent the available options at each decision point, the triangles represent the new bounds of the decision variable calculated using Equation (8) or (9), the solid circles denote feasible options at each decision point for the ant under consideration, dashed paths are the potential feasible solutions and finally the solid path represents a trial feasible solution on the graph. Note that the feasible options represented by the tabu list at an arbitrary decision point are now generally different for each ant, as implied in Figure 4, depending on the partial path covered by the ant to reach the current decision point. A mapping of the type $i = (k-1)*T + t$ is used here to associate the decision point i to the period t of the kth reservoir. This point will be addressed again later in the paper.

It can be clearly seen from these figures that the resulting search space, which is feasible in most cases, is much smaller than the original search space. It is, therefore, expected that the resulting algorithm will perform better than the original unconstrained algorithm. However, a situation might arise in which PCACOA1 fails to produce a feasible solution depending on the partial solution created by the ant and more importantly the magnitude of the water inflow $I_k(t)$ at period t to reservoir k. This happens when the new range defined by Equation (8) or (9) for the corresponding decision variable is empty. This problem is resolved here by offering the ant a single option, upper or lower bound of the decision variable, to choose which will constitute an infeasible solution.

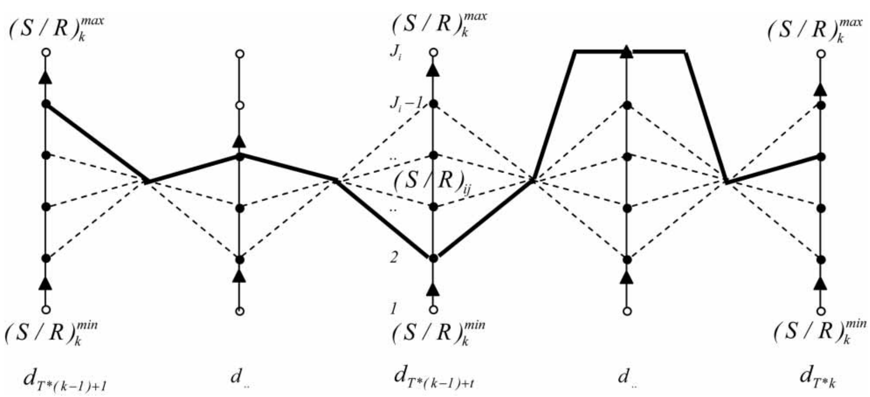

Figure 4 | Base graph of an arbitrary ant in PCACOA1 for reservoir *k* with storage/release volume as decision variable.

PARTIALLY CONSTRAINED ANT COLONY OPTIMIZATION ALGORITHM II (PCACOA2)

The above problem can be resolved for the upstream reservoirs by a simple but effective modification to PCACOA1 leading to an algorithm which will never produce an infeasible operation policy for the upstream reservoirs referred to as PCACOA2. For this, the periods of operation of upstream reservoirs are swept in reverse order and a set of new bounds is calculated for the storage volume at the beginning of the period such that the PCACOA1 is not given any chance of producing infeasible solutions for the upstream reservoirs. To clarify this, consider the storage volume constraint for reservoir k at a period $t+1$:

$$S_k^{\min} \le S_k(t+1) \le S_k^{\max} \tag{10}$$

Substituting $S_k(t+1)$ from Equation (2) into Equation (10) leads to the following constraints for the storage volume $S_k(t)$at the beginning of the period:

$$R_k(t) - I_k(t) + S_k^{\min} \le S_k(t) \le R_k(t) - I_k(t) + S_k^{\max} \tag{11}$$

For this constraint to be valid for any value of release from reservoir k in the range $[R_k^{\min}, R_k^{\max}]$, the following equation should hold.

$$R_k^{\min} - I_k(t) + S_k^{\min} \le S_k(t) \le R_k^{\max} - I_k(t) + S_k^{\max} \tag{12}$$

Combining Equation (12) with the original constraints of Equation (3) leads to the following constraints for the storage volume at the beginning of the period.

$$\bar{S}_k^{\min}(t) \le S_k(t) \le \bar{S}_k^{\max}(t) \tag{13}$$

with

$$\bar{S}_k^{\min}(t) = \mathrm{Max}\,(S_k^{\min}, R_k^{\min} - I_k(t) + S_k^{\min})$$

$$\bar{S}_k^{\max}(t) = \mathrm{Min}\,(S_k^{\max}, R_k^{\max} - I_k(t) + S_k^{\max})$$

where $\bar{S}_k^{\min}(t)$ and $\bar{S}_k^{\max}(t)$ are the new bounds calculated for the storage volume of upstream reservoirs that are used in Equation (3) instead of $S_k^{\min}$ and $S_k^{\max}$. Note that these bounds are calculated once in the beginning of the computation and are then used in search process carried out by PCACOA1 defined earlier. These bounds are generally different from the original bounds and may also differ from one period to another as indicated by the notation used. This procedure when used with PCACOA1 leads to PCACOA2 which will not produce any infeasible operation for upstream reservoirs during the search process. Note that this procedure is also valid when the release volumes are selected as the decision variables of the problem.

Figure 5 shows the base graph of the PCACOA2 for the upstream reservoir k for the case where storage volumes are taken as decision variables. Here the solid squares represent the bounds computed for storage volumes at each decision point using Equation (13), the circles at each decision

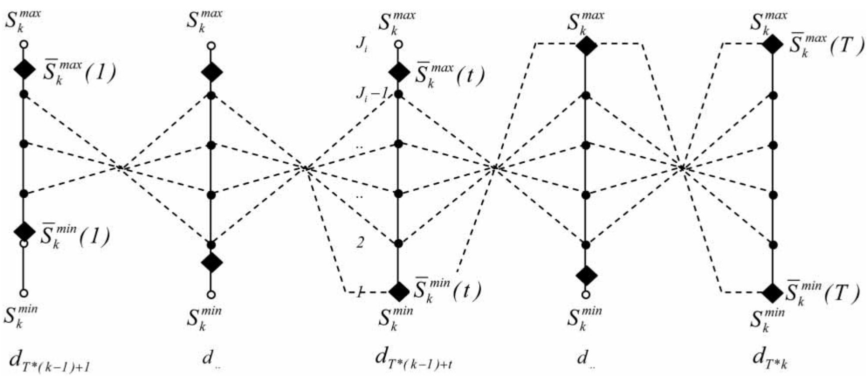

Figure 5 | Base graph of PCACOA2 for upstream reservoir *k* with storage volume as decision variable.

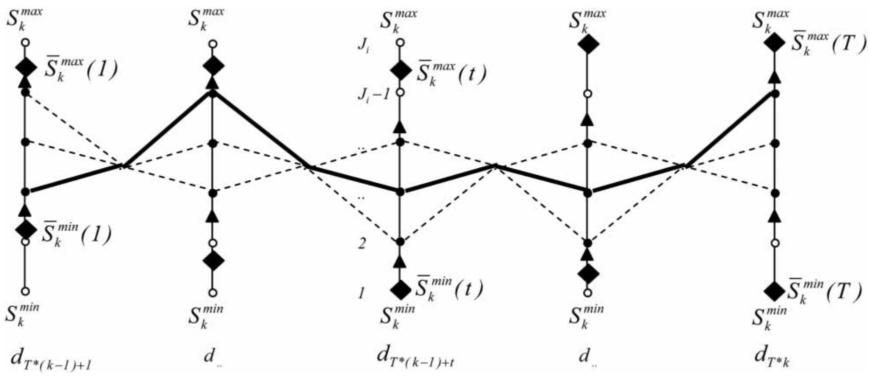

Figure 6 | Base graph of an arbitrary ant in PCACOA2 for upstream reservoir *k* with storage volume as decision variable.

point represent the available options of the storage volumes, and the solid circles denote the option which will not constitute an infeasible solution while the hollow circles are infeasible options at each decision point. Figure 6 shows the base graph of the PCACOA2 for an arbitrary ant on the upstream reservoir *k*. Here the triangles represent the bounds calculated using Equation (9) during solution construction, solid circles represent the feasible options, storage volumes, at each decision point, dashed paths are the potential feasible solutions and finally the solid paths represent a trial, always feasible, solution on the graph. Note that the graphs of the problem for the application of PCACOA2 for the case where release volumes are selected as decision variables are not shown here.

FULLY CONSTRAINED ANT COLONY OPTIMIZATION ALGORITHM (FCACOA)

The proposed PCACOA2, though more efficient than PCACOA1 in constraint satisfaction, might fail to create feasible operation for the downstream reservoirs as implied before. This is due to the fact that PCACOA2 might face a situation in which the range calculated by Equation (8)/(9) for the release/storage volume of the downstream reservoirs is empty leading to infeasible operations of the downstream reservoirs. The procedure defined via Equation (13) for storage bound modification cannot be directly applied to downstream reservoirs since the value of inflow $I_k(t)$ to these reservoirs is not known before the operation of all upstream reservoirs is known. A remedy to this problem can be found by using Equation (13) to modify the storage bounds of each downstream reservoir once a partial solution represented by the full operation of all corresponding upstream reservoirs is constructed. For such a partial solution, the inflow to the reservoir under consideration can be determined and used in Equation (13) to calculate the modified storage bounds of the downstream reservoirs in turn. This procedure when used with PCACOA2 will lead to FCACOA which only creates feasible solutions for any multi-reservoir system.

A final note has to be made here regarding the sequence of the decision points used to define the multi-reservoir operation graph. This sequence should be known *a priori* for each ant irrespective of the formulation used. In UACOA, this sequence can be arbitrarily associated with the operation period *t* of the reservoir *k* since the local operation policies are constructed independently. For the constrained algorithms, however, the sequence of the decision points covered by each ant should be arranged such that:

1. The value of the decision variable (storage/release volume depending on the decision variables selected) at an arbitrary period *t* is known before operation policy of period $(t+1)$ is determined. This is a property required for the application of PCACOA1 defined by Equation (8)/(9).
2. The full operation of all reservoirs upstream to an arbitrary reservoir *k* should be known before attempting to construct the operation policy of the reservoir under consideration. This is a property required for the application of FCACOA defined by Equation (13).

This is achieved here by defining the set of decision points as $\boldsymbol{D} = [\boldsymbol{d_1}, \boldsymbol{d_2}, . , \boldsymbol{d_k}, . , . , \boldsymbol{d_K}]$, where d_k represents the set of decision points corresponding to the operation periods of reservoir k defined as $\boldsymbol{d_k} = [d_1, d_2, ., d_t, ., ., d_T]_k$ with $d_{t,k}$ denoting the decision point corresponding to operation period t of the kth reservoir. With this projection, the ants are now required to cover the decision points in turn from the first point $d_{1,1}$ to the last one d_{T*K}.

NUMERICAL EXPERIMENTS

In this section the well-known four- and ten-reservoir operation problems are considered as the test examples to verify the versatility and efficiency of the proposed methods. The hypothetical four-reservoir system first proposed by Larson (1968) is shown in Figure 7. This problem has been used as a benchmark example to test many combinatorial optimization algorithms such as DP and heuristic search methods due to the availability of its global solution. The four-reservoir problem was first formulated and solved with LP and Dynamic Programming and Successive Approximation (DPSA) by Larson (1968). Heidari *et al.* (1971) applied Discrete Differential Dynamic Programming (DDDP), and Kumar & Baliarsinch (2003) applied Folded Dynamic Programming (FDP) to evaluate the performance of their respective proposed techniques. Esat & Hall (1994) and Wardlaw & Sharif (1999) applied GA with real coding for the optimal solution of this problem. Recently, Kumar & Reddy (2007) used this problem to test a modified PSO algorithm suggested for multipurpose operation of reservoirs.

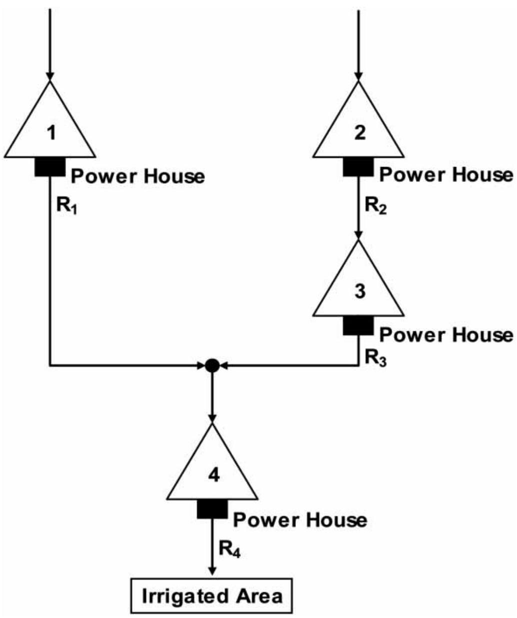

Figure 7 | Four-reservoir problem.

For the four-reservoir problem, there are inflows to the first and second reservoirs only, and these are two and three units, respectively, in all time periods. The initial storages in all reservoirs are five units. The benefit functions $b_k(t)$of reservoir k at period t tabulated by Larson (1968) and by Heidari *et al.* (1971) are shown in Table 1. The following values are used for the bounds of the storages and releases: $S_1^{\min}, S_2^{\min}, S_3^{\min} = 0.0$; $S_1^{\max}, S_2^{\max}, S_3^{\max} = 10$; $S_4^{\max} = 15$; $R_1^{\min}, R_2^{\min}, R_3^{\min}, R_4^{\min} = 0.0$; $R_1^{\max} = 3$; $R_2^{\max}, R_3^{\max} = 4$; $R_4^{\max} = 7$.

The original four-reservoir problem was devised for 12 operation periods ($T = 12$) with a time interval of 2 h. However, the problem is solved here for different operation periods ranging from 12 to 96, so that the performance of the proposed methods can be tested for large scale reservoir operation problems by comparing the results to those obtained by Murray & Yakowitz (1979) and Wardlaw & Sharif (1999).

A set of preliminary runs is first conducted to find the proper values of MMAS parameters as shown in Table 2 for the four-reservoir problem. In Table 2, α is the parameter that controls the relative weight of the pheromone trail; β is the parameter that controls the relative weight of the heuristic value; P_{best} is a specified probability by which each component of the current global-best solution will be selected by ants in the next iteration; and ρ is the coefficient representing the pheromone evaporation. To consider the scale of the problems, a colony size of 200 with a maximum number of 3,000 iterations amounting to a maximum number of 600,000 function evaluations is used for operation of the four-reservoir problem over 12, 24, 36, and 48 time periods. For larger scale problems of four-reservoir operation over 60, 72, 84, and 96 operation time periods, a colony size of 200 with a maximum number of 4,000, 5,000, 6,000, and 7,000 iterations amounting to a maximum number of 800,000, 1,000,000, 1,200,000, 1,400,000 function evaluations is used, respectively. It should be noted

Table 1 | Benefit functions of each reservoir at each period for four- and ten-reservoir problems

	Benefit functions												
		Period											
Problem	**Reservoir**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Four	1	1.4	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1	1.1
	2	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1	1.1	1.4
	3	1.1	1.4	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1
	4	2.5	2.7	3.1	3.6	4.1	3.8	4	4.2	4.4	3.6	2.9	2.6
Ten	1	1.4	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1	1.1
	2	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1	1.1	1.4
	3	1.1	1.4	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1
	4	1.4	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1	1.1
	5	1.14	1.25	1.34	1.45	1.56	1.67	1.5	1.4	1.3	1.2	1.1	1
	6	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1	1.1	1.4
	7	2.5	2.7	3.1	3.6	4.1	3.8	4	4.2	4.4	3.6	2.9	2.6
	8	1.14	1.25	1.34	1.45	1.56	1.67	1.5	1.4	1.3	1.2	1.1	1
	9	1.1	1.4	1.8	2.2	1.8	2	2.2	2.5	1.8	1.2	1	1
	10	2.1	3.5	2.8	3.7	3.6	4	3.9	2.9	3.2	2.8	3	2.7

Table 2 | Values of MMAS parameters for four-reservoir problem

Decision variable	α	β	ρ	p_{best}
Release	1	0.3	0.9	0.2
Storage	1	0.0	0.9	0.2

that this problem is solved on a 2.8 MHZ Core 2 due Pentium 4 PC.

Table 3 shows the results of 10 runs carried out for the four-reservoir operation problem with releases taken as decision variables using parameters of Table 2 over 12, 24, 36, 48, 60, 72, 84 and 96 operation time periods using UACOA and proposed constrained algorithms. It is clearly seen that all the results, including minimum, maximum and average objective function values, obtained with FCACOA over all periods are better than those produced by UACOA, PCACOA1 and PCACOA2. The superiority of the results obtained by the fully constrained version is more evident for the longer periods illustrating the efficiency of the FCACOA for large scale problems. It is also interesting to note that the scattering of the solutions obtained decreases as one moves from PCACOA1 to PCACOA2 and from PCACOA2 to FCACOA.

Table 4 shows the results obtained for the four-reservoir operation problem with storage volumes taken as decision variables. It is seen here that the unconstrained algorithm could create final feasible solutions only for the shortest operation of the four-reservoir problem over 12 time periods, while the solutions produced by the FCACOA are all feasible. It is again seen that FCACOA performs better than the partially constrained algorithms. It is more interesting to note that the number of final feasible solutions created by PCACOA1 and PCACOA2 decreases with increasing number of operation periods with the number of feasible solutions of PCACOA2 always greater than that of PCACOA1. Comparison of the results of Tables 3 and 4 indicates that the performance of all methods with release volumes taken as decision variables is better than those when storage volumes are taken as decision variables which could be attributed to the following reasons. First, the search space size of the problem is greater when storage volumes are taken as decision variables of the problem. Secondly, and more importantly, definition of heuristic information is not possible when storage volumes are selected as decision variables. This emphasizes the importance of the heuristic information on the performance of the ACO algorithms for reservoir operation problems.

This problem has also been solved by other researchers. Table 5 compares the best results of the four-reservoir problem produced by the proposed FCACOA with some other available results. It is clearly seen that the proposed FCACOA method shows superior performance to those of

Table 3 | Maximum, minimum and average objective function values over 10 runs for four-reservoir problem using UACOA, PCACOA1, PCACOA2 and FCACOA with release as decision variable

		Objective function values					
Model	Periods	Maximum	Minimum	Average	Scaled standard deviation	No. of runs with final feasible solution	CPU time for each run (seconds)
UACOA	12	401.2	400.9	401.05	0.0003	10	59
	24	809.7	723.4	798.75	0.0336	10	118
	36	1,214.2	1,208.4	1,212.12	0.0015	10	177
	48	1,617.9	1,611.02	1,614.9	0.0013	10	236
	60	2,022.4	2,014.7	2,018.74	0.0012	10	394
	72	2,427.8	2,406.1	2,418.78	0.0027	10	590
	84	2,833.4	2,811.1	2,825.18	0.0022	10	826
	96	3,235.5	3,193.6	3,218.02	0.0041	10	1,100
PCACOA1	12	401.3	401.1	401.23	0.00018	10	62
	24	810.3	809.3	809.94	0.0005	10	124
	36	1,219.2	1,217.5	1,218.11	0.0005	10	186
	48	1,626.6	1,624.5	1,625.58	0.0006	10	248
	60	2,035.2	2,030.6	2,033.97	0.0009	10	413
	72	2,444.1	2,440.2	2,441.78	0.0004	10	620
	84	2,853.4	2,846.5	2,850.43	0.0007	10	868
	96	3,262.0	3,254.29	3,258.29	0.0008	10	1,157
PCACOA2	12	401.3	401.1	401.23	0.00016	10	61
	24	810.5	809.5	809.96	0.0004	10	122
	36	1,219.2	1,217.1	1,218.4	0.0005	10	183
	48	1,627.0	1,624.5	1,625.8	0.0005	10	244
	60	2,035.6	2,033.7	2,034.7	0.0004	10	407
	72	2,444.6	2,440.2	2,442.58	0.0004	10	610
	84	2,853.4	2,848.2	2,850.88	0.0004	10	854
	96	3,262.0	3,256.3	3,258.34	0.0004	10	1,139
FCACOA	12	401.3	401.2	401.25	0.00013	10	62
	24	810.6	80.9.8	810.23	0.0002	10	124
	36	1,219.4	1,217.9	1,218.73	0.0004	10	186
	48	1,627.3	1,625.2	1,626.47	0.0004	10	248
	60	2,036.9	2,033.7	2,035.5	0.0004	10	413
	72	2,446.7	2,440.6	2,443.29	0.0003	10	620
	84	2,854.1	2,850.1	2,852.16	0.0004	10	868
	96	3,262.4	3,256.3	3,260.05	0.0004	10	1,157

GA of Wardlaw & Sharif (1999) and the Improved Ant Colony System of Jalali (2005) for all the operation periods considered. It is interesting to note that the proposed FCACOA is able to locate the optimal solution for two cases, of 12 and 24 operation periods, while other methods were able to do so only for the shortest, 12 operation period, case. The fact that the proposed FCACOA could not compete with DDDP for the larger operation periods can be described by the fact that DDDP solutions are globally optimum since the considered problem is of discrete nature.

And finally Table 6 compares the computational effort required by the proposed FCACOA and those of GA of Wardlaw & Sharif (1999) represented by the number of function evaluations required to reach the same local solution for the four-reservoir operation problem. Comparison of the result shows that the proposed FCACOA is more efficient than GA for all operation periods considered. This problem has also been solved by Kumar & Reddy (2007) using the EMPSO algorithm for the shortest case of 12 operation periods requiring 325,400 function evaluations to get the optimal solution of 401.3 which is obtained in just 64,000 function evaluations using the proposed FCACOA.

Comparison of the final release and storage volumes produced by LP (Larson 1968) and the FCACOA for the

Table 4 | Maximum, minimum and average objective function values over 10 runs for four-reservoir problem using UACOA, PCACOA1, PCACOA2 and FCACOA with storage as decision variable

Model	Time periods	Objective function values: Maximum	Minimum	Average	Scaled standard deviation	No. of runs with final feasible solution	CPU time for each run (seconds)
UACOA	12	391.6	363.5	383.84	0.0267	10	104
	24	776.9	in	in	–	5	208
	36	1,162.1	in	in	–	4	312
	48	in	in	in	–	0	416
	60	in	in	in	–	0	693
	72	in	in	in	–	0	1,040
	84	in	in	in	–	0	1,456
	96	in	in	in	–	0	1,941
PCACOA1	12	395.0	385.0	390.89	0.0094	10	65
	24	794.5	in	in	–	9	130
	36	1,198.4	in	in	–	7	195
	48	1,587.4	in	in	–	8	260
	60	1,984.4	in	in	–	7	433
	72	2,380.6	in	in	–	7	650
	84	2,774.5	in	in	–	6	910
	96	3,170.9	in	in	–	5	1,213
PCACOA2	12	395.8	392.3	393.52	0.0027	10	65
	24	794.9	790.1	792.18	0.0016	10	130
	36	1,218.89	1,217.1	1,218.13	0.0005	10	195
	48	1,588.3	1,560.7	1,577.73	0.0057	10	260
	60	1,985.4	1,957.6	1,975.66	0.0040	10	433
	72	2,382.1	in	in	–	7	650
	84	2,778.8	2,765.5	2,772.0	0.0014	10	910
	96	3,173.9	in	in	–	8	1,213
FCACOA	12	400.3	392.5	396.42	0.0007	10	58
	24	800.0	790.5	795.7	0.0004	10	116
	36	1,219.4	1,217.9	1,218.73	0.0004	10	174
	48	1,590.0	1,577.9	1,584.55	0.0024	10	232
	60	1,987.7	1,975.2	1,980.7	0.0022	10	387
	72	2,382.4	2,369.8	2,377.4	0.0018	10	580
	84	2,783.4	2,270.7	2,776.01	0.0013	10	812
	96	3,183.5	3,165.5	3,171.01	0.0012	10	1,083

Table 5 | Comparison the results obtained with different methods for four-reservoir operation problem

Operation periods	DDDP (Wardlaw & Sharif 1999)	GA (Wardlaw & Sharif 1999)	Improved ACO (Jalali 2005)	Present work (FCACOA)
12	401.3	401.3	401.3	401.3
24	810.6	808.9	810.2	810.6
36	1,220.1	1,218.6	1,217.1	1,219.4
48	1,629.6	1,626.5	1,617.6	1,627.3
60	2,039.1	2,036.9	–	2,036.9
72	2,448.6	2,446.0	–	2,446.7
84	2,858.1	2,847.5	–	2,854.1
96	3,267.6	3,259.8	–	3,262.4

four-reservoir problem over 12 operation periods is presented in Figure 8. This problem is known to have more than one global optimum solution; therefore, it is interesting to note that the optimal operation created by the proposed FCACOA differs from that obtained by LP.

The hypothetical ten-reservoir system considered as the second test problem is shown in Figure 9. The ten-reservoir problem was first formulated and solved with constrained Differential Dynamic Programming (DDP) by Murray & Yakowitz (1979). Wardlaw & Sharif (1999) applied GA with real coding for the optimal solution of this problem. The objective function of Equation (1) is to be maximized over 12 operation periods. There are inflows to the upstream

Table 6 | Comparison of the number of function evaluations required by GA and proposed FCACOA to reach local solution for four-reservoir operation problem

		Number of function evaluations to obtain local solution	
Operation periods	Local solution	Real coded GA (Wardlaw & Sharif 1999)	Present Work (FCACOA)
12	401.3	100,000	64,000
24	808.9	180,000	92,000
36	1,218.6	380,000	300,000
48	1,626.5	420,000	340,000
60	2,036.9	800,000	800,000
72	2,446.0	820,000	720,000
84	2,847.5	880,000	660,000
96	3,259.8	1,100,000	1,040,000

reservoirs only (reservoir 1, 2, 3, 5, 6 and 8) in all time periods. The initial storage and target storage at the end of the operation period are considered for all reservoirs. The benefit functions $b_k(t)$ of reservoir k at period t are shown in Table 1. Other details of the problem are tabulated by Murray & Yakowitz (1979).

A set of preliminary runs is first conducted to find the proper values of MMAS parameters as shown in Table 7 for the ten-reservoir operation problem. For the ten-reservoir problem, a colony size of 500 with a maximum number of 3,000 iterations amounting to a maximum number of 1,500,000 function evaluations is used. It should be noted that this problem is also solved on a 2.8 MHZ Core 2 due Pentium 4 PC.

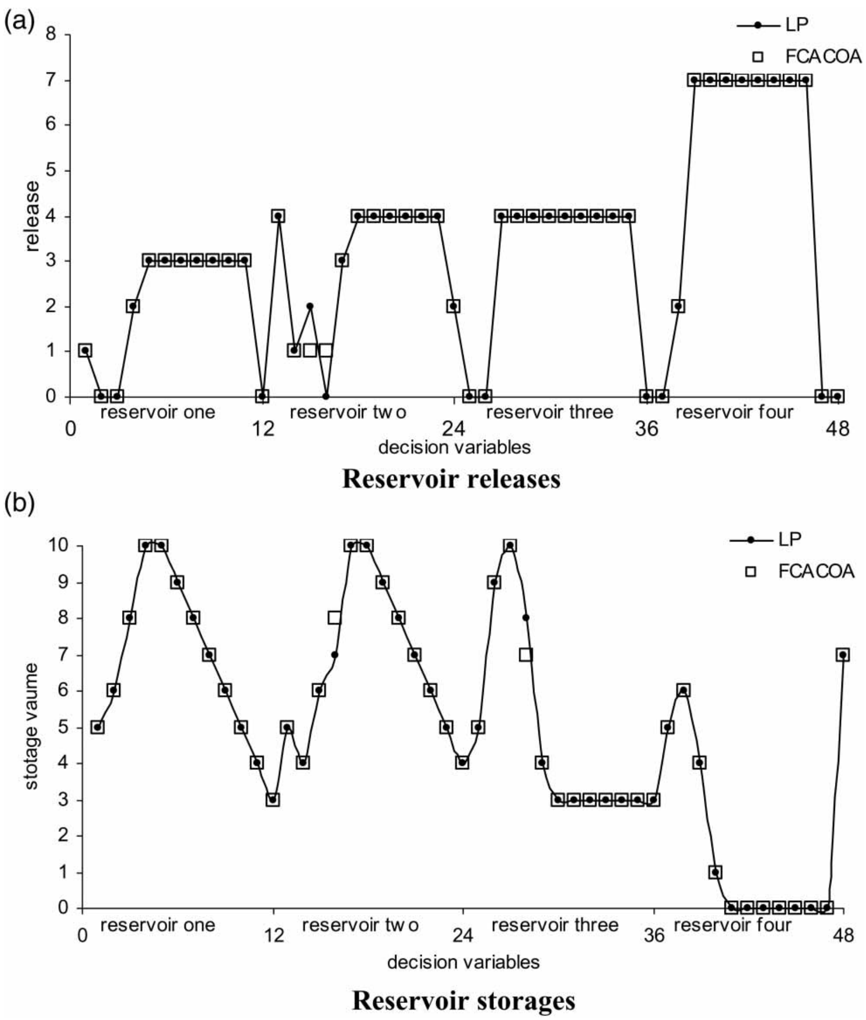

Figure 8 | Comparison between the best result of the proposed FCACOA with releases as decision variables and result of LP model over 12 operation periods. (a) Reservoir releases. (b) Reservoir storages.

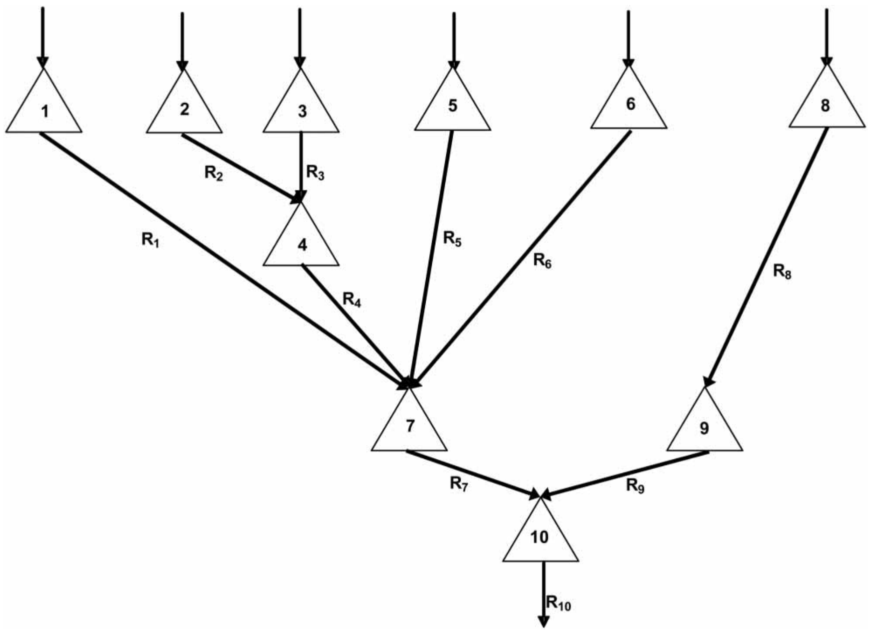

Figure 9 | Ten-reservoir problem.

Table 7 | Values of MMAS parameters for ten-reservoir problem

Decision variable	α	β	ρ	p_{best}
Release	1	0.0	0.85	0.15
Storage	1	0.0	0.88	0.15

Table 8 shows the results of 10 runs carried out for the ten-reservoir problem with storage/release volumes taken as decision variables using the parameters of Table 7 over 12 operation periods using UACOA and proposed constrained algorithms. It is clearly seen that all the results, including minimum, maximum and average objective function values, obtained with FCACOA are better than those produced by UACOA, PCACOA1 and PCACOA2 for either of the decision variables. Once again, the results of Table 8 indicate that the performance of all methods with release volumes taken as decision variables is better than those when storage volumes are taken as decision variables.

Table 9 compares the best results of the ten-reservoir problem produced by proposed FCACOA with some other available results. The objective function value of the global optimum solution for this problem is equal to 1,194.44 obtained using LP (Jalali *et al*. 2007). The objective function value of 1,190.652 was obtained using constrained DDP by Murray & Yakowitz (1979) for this problem. Wardlaw & Sharif (1999) used a real coding GA for this problem and required 1,250,000 function evaluations to get an objective function value of 1,190.25. This problem has also been solved by Jalali (2005) using the Improved Ant Colony System and the Improved Ant Colony System with discrete refinement (DR) mechanism leading to the objective function value of 1,153.64 and 1,174.69, respectively, requiring 6,000,000 function evaluations for both proposed algorithms. Jalali & Afshar (2005) solved this problem using ACO with pheromone re-initiation (PRI), partial path replacement (PPR), and DR mechanism, getting the best objective function value of 1,178.8. The best result of 1,190.26 was obtained here using FCACOA with release volumes taken as decision variables. It is clearly seen that the proposed FCACOA method shows superior performance to those of Improved Ant Colony System of Jalali (2005) and methods of Jalali & Afshar (2005) and marginally improved performance to those of GA of Wardlaw & Sharif (1999).

A note should be made regarding the apparently different level of performance of the proposed methods and in particular the FCACOA for the four- and ten-reservoir problems in comparison to GA of Wardlaw & Sharif (1999).

Table 8 | Maximum, minimum and average objective function values over 10 runs for ten-reservoir problem using UACOA, PCACOA1, PCACOA2 and FCACOA with storage/release as decision variable

Model	Decision variable	Objective function values: Maximum	Minimum	Average	Scaled standard deviation	No. of runs with final feasible solution	CPU time for each run (seconds)
UACOA	Release	1,184.19	1,173.07	1,180.63	0.0034	10	570
	Storage	1,040.50	in	in	–	1	1,160
PCACOA1	Release	1,186.75	1,175.62	1,181.14	0.0025	10	540
	Storage	1,170.94	1,153.85	1,161.90	0.0052	10	660
PCACOA2	Release	1,188.24	1,177.66	1,183.88	0.0031	10	510
	Storage	1,173.44	1,144.97	1,161.30	0.0069	10	610
FCACOA	Release	1,190.26	1,178.87	1,184.71	0.0028	10	510
	Storage	1,174.09	1,166.47	1,168.83	0.0020	10	600

Table 9 | Comparison of the results obtained with different methods for ten-reservoir operation problem

Operation periods	LP (Jalali *et al.* 2007)	Constrained DDP (Murray & Yakowitz 1979)	GA (Wardlaw & Sharif 1999)	Improved ACO (Jalali 2005)	Improved ACO with DR (Jalali 2005)	Present work (FCACOA)
12	1,194.44	1,190.652	1,190.25	1,153.64	1,174.69	1,190.26

While the results produced by the proposed FCACOA are better than all available results, except for DP based methods, including those of Wardlaw & Sharif (1999) for both test cases, the improvement made in the results for the ten-reservoir problem seems to be marginal (as seen from Table 9). This is due to the fact that ACOA is essentially of a discrete nature requiring discretization of the search space when applied to continuous problems leading to poor performance compared to discrete problems while the GA of Wardlaw & Sharif (1999) is a real-coded GA suitable for continuous problems. Superiority of the proposed FCACOA to GA of Wardlaw & Sharif (1999) is more evident from the results produced for the discrete problem of the four-reservoir system as indicated in Table 5.

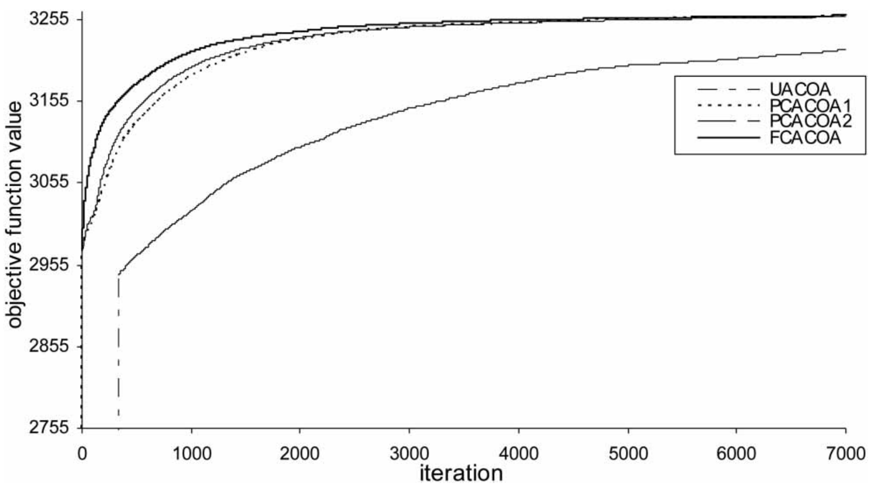

Figure 10 | Average objective function value versus iterations for four-reservoir problem over 96 operation periods using UACOA, PCACOA1, PCACOA 2 and FCACOA with release as decision variable.

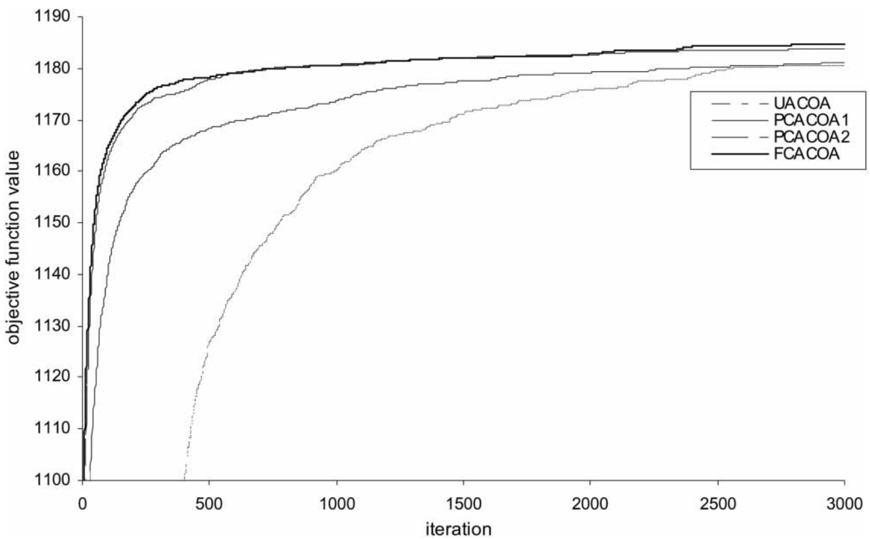

Figure 11 | Average objective function value versus iterations for ten-reservoir problem over 12 operation periods using UACOA, PCACOA1, PCACOA2 and FCACOA with release as decision variable.

Finally, the claim that the search space created by FCACOA is totally feasible is supported by the two convergence curves shown in Figures 10 and 11 for average objective function value obtained over 96 operation periods for the four-reservoir operation problem and over 12 operation periods for the ten-reservoir operation problem, respectively, with release taken as decision variable. It is seen that the initial colony of FCACOA has an average objective function value way over that of PCACOA1, PCACOA2 and UACOA, due to the fact that all the solutions created by FCACOA are feasible and, therefore, are not penalized.

CONCLUDING REMARKS

Incremental solution building capability of the ACO algorithm was exploited in this paper to develop three constrained versions of the ACO algorithm for the efficient solution of multi-reservoir operation problems. In the first version, the ants were forced to locally move in the feasible region of the search space by providing each ant with a tabu list consisting of those options which constitute a local feasible solution. The tabu list was constructed using the decision made at previous decision points such that the continuity equation and the box constraints of release and storage volumes were simultaneously satisfied. In the second algorithm, storage values of the upstream reservoirs constituting infeasible operations were recognized and excluded from the search space before the main search started. These two algorithms, however, were shown to fail in some rare cases. A third algorithm was then proposed and shown to be capable of only searching the feasible region of the search space. Proposed methods were applied to the problems of four- and ten-reservoir operation problem and the results were presented and compared with those of the original unconstrained algorithm and existing results in the literature. Results indicated that the proposed constrained algorithms were more effective and efficient than the conventional unconstrained ACO algorithm in solving multi-reservoir operation problems. It was also shown that the fully constrained algorithm consistently gave better quality solutions than existing heuristic search methods, including GA, with less computational effort.

REFERENCES

Afshar, M. H. 2007 Partially constrained ant colony optimization algorithm for the solution of constrained optimisation problems: application to storm water network design. *Advance in Water Resources* **30** (4), 954–965.

Afshar, M. H. & Moeini, R. 2008 Partially and fully constrained ant algorithms for the optimal solution of large scale reservoir operation problems. *Journal of Water Resources Management* **22** (1), 1835–1857.

Bozorg Haddad, O., Afshar, A. & Marino, M. A. 2006 Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization. *Water Resources Management* **20** (5), 661–680.

Bozorg Haddad, O., Afshar, A. & Mariño, M. A. 2008 Design-operation of multi-hydropower reservoirs: HBMO approach. *Water Resources Management* **22** (12), 1709–1722.

Cai, X., Mckinney, D. C. & Larson, L. S. 2002 Piece-by-piece approach to solving large nonlinear water resources management models. *ASCE Journal of Water Resources Planning and Management* **127** (6), 363–368.

Chang, F. J. & Chen, L. 1998 Real-coded genetic algorithm for rule-based flood control reservoir management. *Water Resources Management* **12** (3), 185–198.

Chang, F. J., Chen, L. & Chang, L. C. 2005 Optimizing the reservoir operating rule curves by genetic algorithms. *Hydrological Processes* **19** (11), 2277–2289.

Chen, L. 2003 Real time genetic algorithm optimization of long term reservoir operation. *Journal of the American Water Resources Association* **39** (5), 1157–1165.

Chen, L. & Chang, F. J. 2007 Applying a real-coded multi-population genetic algorithm to multi-reservoir operation. *Hydrological Processes* **21** (5), 688–698.

Dorigo, M. & Gambardella, L. M. 1997 Ant colonies for the traveling salesman problem. *Biosystems* **43**, 73–81.

Esat, V. & Hall, M. J. 1994 Water resource system optimization using genetic algorithms. *Hydro informatics'94, pro., 1st Int. Conf. on Hydro informatics*, Balkerma, Rotterdam, The Netherlands, pp. 225–231.

Fahmy, H. S., King, J. P., Wentzle, M. W. & Seton, J. A. 1994 Economic optimization of river management using genetic algorithms. Int. summer Meeting, AM. Soc. Agric. Engrs, paper no.943034, St. Joseph, Michigan.

Heidari, M., Chow, V. T., Kokotovic, P. V. & Meredith, D. D. 1971 Discrete differential dynamic programming approach to water resources systems optimization. *Water Resource Research* **7** (2), 273–282.

Jalali, M. R. 2005 *Optimal Design and Operation of Hydro Systems by Ant Colony Algorithms: New Heuristic Approach*. PhD Thesis, Department of Civil Engineering, Iran University of Science and Technology.

Jalali, M. R. & Afshar, A. 2005 *Semi-continuous ACO Algorithms (technical report)*. Hydroinformatics Center, Civil Engineering Department, Iran University of Science and Technology, Tehran, Iran.

Jalali, M. R., Afshar, A. & Marino, M. A. 2007 Multi-colony ant algorithm for continuous multi-reservoir operation optimization problems. *Journal of Water Resources Research* **21** (9), 1429–1447.

Kuczera, G. 1989 Fast multi reservoir multi-period linear programming models. *Water Resources Research* **25** (2), 169–176.

Kumar, D. N. & Baliarsingh, F. 2003 Folded dynamic programming for optimal operation of multi-reservoir system. *Water Resources Management* **17** (5), 337–353.

Kumar, D. N. & Reddy, J. 2007 Multipurpose reservoir operation using particle swarm optimization. *Journal of Water Resources Planning and Management* **133** (3), 192–201.

Larson, R. E. 1968 *State Increment Dynamic Programming*. American Elsevier Publishing Company Inc., New York.

Madadgar, S. & Afshar, A. 2009 An improved continuous ant algorithm for optimization of water resources problems. *Water Resources Management* **23** (10), 2119–2139.

Mays, L. W. & Tung, Y. K. 2002 *Hydrosystems Engineering and Management*. Water Resource Publications, Littleton, Colorado.

Mousavi, J. & Karamouz, M. 2003 Computational improvement for dynamic programming models by diagnosing infeasible storage combinations. *Advances in Water Resources* **26** (8), 851–859.

Murray, D. M. & Yakowitz, S. 1979 Constrained differential dynamic programming and its application to multireservoir control. *Water Resources Research* **15** (5), 1017–1027.

Oliveira, R. & Loucks, D. 1997 Operation rules for multi reservoir systems. *Water Resources Research* **33** (4), 839–852.

Peng, C. H. & Buras, N. 2000 Dynamic operation of a surface water resources system. *Water Resources Research* **36** (9), 2701–2709.

Perera, B. J. C. & Conder, G. P. 1998 Computational improvement for stochastic dynamic programming models for urban water supply reservoirs. *Journal of American Water Resources Association* **34** (2), 267–278.

Srinivasan, K., Neelakantan, T. R., Narayan, P. S. & Kumar, C. N. 1999 Mixed-integer model for reservoir performance optimization. *Journal of Water Resources Planning and Management* **125** (5), 298–301.

Tu, M. Y., Hsu, N. S. & Yeh, W. W. G. 2003 Optimization of reservoir management and operation with hedging rules. *Journal of Water Resources Planning and Management* **129** (2), 86–97.

Wardlaw, R. & Sharif, M. 1999 Evaluation of genetic algorithms for optimal reservoir system operation. *Journal of Water Resources Planning and Management* **125** (1), 25–33.

Yakowitz, S. 1982 Dynamic programming application in water resources. *Water Resources Research* **18** (4), 673–696.

Yeh, W. G. 1985 Reservoir management and operations models: a state-of-the-art review. *Water Resources Research* **21** (12), 1797–1818.

Yoo, J. H. 2009 Maximization of hydropower generation through the application of a linear programming model. *Journal of Hydrology* **376** (1), 182–187.

First received 25 June 2011; accepted in revised form 14 June 2012. Available online 18 September 2012