Advances in Engineering Software 41 (2010) 188–195

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

A parameter free Continuous Ant Colony Optimization Algorithm for the optimal design of storm sewer networks: Constrained and unconstrained approach

M.H. Afshar

Civil Engineering, Iran University of Science and Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 21 December 2007
Received in revised form 24 July 2009
Accepted 28 August 2009
Available online 22 October 2009

Keywords:
Constrained Continuous Ant Colony Optimization Algorithm
Explicit constraints
Optimal design
Storm water network

ABSTRACT

This paper describes the application of the newly introduced Continuous Ant Colony Optimization Algorithm (CACOA) to optimal design of sewer networks. Two alternative approaches to implement the algorithm is presented and applied to a storm sewer network in which the nodal elevations of the network are considered as the decision variables of the optimization problem. In the first and unconstrained approach, a Gaussian probability density function is used to represent the pheromone concentration over the allowable range of each decision variable. The pheromone concentration function is used by each ant to randomly sample the nodal elevations of the trial networks. This method, however, will lead to solutions which may be infeasible regarding some or all of the constraints of the problem and in particular the minimum slope constraint. In the second and constrained approach, known value of the elevation at downstream node of a pipe is used to define new bounds on the elevation of the upstream node satisfying the explicit constraints on the pipe slopes. Two alternative formulations of the constrained algorithm are used to solve a test example and the results are presented and compared with those of unconstrained approach. The methods are shown to be very effective in locating the optimal solution and efficient in terms of the convergence characteristics of the resulting algorithms. The proposed algorithms are also found to be relatively insensitive to the initial colony and size of the colony used compared to the original algorithm.

1. Introduction

Artificial life emerged through the interaction of biology and computer sciences. Artificial life not only contributes to understanding the secret of life, but also provides innovative concepts and important approaches to practical applications of artificial intelligence. Some bio-inspired mathematical models in artificial life such as emergent colonization in the artificial ecology and ant algorithms have provided the idea of distributed algorithms which have been successfully applied to the parallel optimization and design. The underlying idea is that some complex tasks may be performed by distributed activities over massively parallel systems composed of computationally simple elements [2].

Ant algorithms were inspired by the observation of real ant colonies. Ants are social insects, that is, insects that live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. Social insects have captured the attention of many scientists because of the high structuration level their colonies can achieve, especially when compared to the relative simplicity of the colony's individuals. An important and interesting behavior of ant colonies is their foraging behavior, and, in particular, how ants can find shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called pheromone, forming in this way a pheromone trail. Ants can smell pheromone and, when choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. The pheromone trail allows the ants to find their way back to the food source (or to the nest). Also, it can be used by other ants to find the location of the food sources found by their nest mates [12].

It has been shown experimentally that this pheromone trail following behavior can give rise, once employed by a colony of ants, to the emergence of shortest paths. That is, when more paths are available from the nest to a food source, a colony of ants may be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back. This particular behavior of ant colonies has inspired the Ant Colony Optimization (ACO) algorithm, in which a set of artificial ants cooperate to find solutions to a given optimization problem by depositing pheromone trails throughout the search space [11]. ACO has proven to be an efficient and versatile tool for solving various combinatorial optimization problems. Several versions of ACO have been proposed, but they all follow the same basic ideas:

E-mail address: mhafshar@iust.ac.ir

0965-9978/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2009.09.009

– Search performed by a population of individuals, i.e. simple independent agents.
– Incremental construction of solutions.
– Probabilistic choice of solution components based on stigmergic information.
– No direct communication between the individuals.

Instances of ACO have been applied extensively to a variety of discrete combinatorial optimization problems like the Traveling Salesman Problem, the Quadratic Assignment Problem, the Network Communication Routing Problem, Vehicle Routing Problem, Job-Shop Scheduling, Sequential Ordering, Graph Coloring, Time Tabling, Shape Optimization, and so on.

Application of ACOAs, however, to water resources problems is of recent origin. Abbaspour et al. [1] used the ACO algorithm for estimating the unsaturated soil hydraulic parameters. Zecchin et al. [31] compared the performance of original ant system with that of Max–Min Ant System (MMAS), a modified version of the ant system proposed by Stutzle and Hoos [28] for optimization of water distribution networks. Simpson et al. [27] discussed the selection of parameters used in the ACO algorithm for pipe network optimization problems. More recently, Maier et al. [20] compared the performance of the ACO algorithm with that of GAs for the optimization of water distribution networks. Afshar [3] proposed a new transition rule for ACO algorithms using elitist strategies and applied the method to pipe network optimization problems. That method was shown to overcome the premature convergence problem encountered by elitist ACO algorithms while improving the convergence characteristics of the algorithms compared to alternative methods such as MMAS. Afshar [4] used an adaptive refinement strategy to improve the efficiency of ACOAs for the solution of continuous optimization problems and successfully applied the method to storm sewer network optimization problems using MMAS. Afshar and Marino [6] exploited the incremental solution building capability of ACOAs to efficiently solve the layout optimization of tree networks where a tree growing algorithm was used to keep the ants in the feasible region of the search space.

1.1. Continuous Ant Colony Optimization (CACO)

Many optimization problems can be formulated as continuous optimization problems. These problems are characterized by the fact that the decision variables have continuous domains, in contrast to the discrete domains of the variables in a combinatorial optimization problem. Since the emergence of ant algorithms as an optimization tool, some attempts were also made to use them for tackling continuous optimization problems. However, at the first sight, applying the ACO meta-heuristic to continuous domain was not straightforward. Hence, the methods proposed often drew inspiration from ACO, but did not follow exactly the same methodology. Up to now, only a few ant approaches for continuous optimization have been proposed in the literature. The first method called Continuous Ant Colony Optimization (CACO) was proposed by Bilchev and Parmee [8], later used by some others [29]. CACO uses the ant colony framework to perform local searches, whereas global search is handled by a genetic algorithm. Indeed, the global ants perform a simple evaluation of some regions defined in the search space, in order to update the regions fitness. The creation of some new regions is handled by a process very similar to a genetic algorithm, using common operators that are assimilated by the authors to some real ant colonies behavior like random walk (playing the part of crossovers and mutations). The local level is handled by ants that explore more systematically the regions with a simple descending behavior, in order to move regions closer to the optimum. The algorithm sends some local ants to the regions; these ants lay down some pheromonal spots when they find an improvement of the objective function and the spots are attractive for all the ants of the colony. This process is close to the original metaphor, but unfortunately, the use of two different processes inside the CACO algorithm leads to a requirement of delicate setting of parameters [10,14,19].

Other methods include the Asynchronous Parallel Implementation (API) algorithm inspired by primitive ants behavior and using a tandem-running which involves two ants and leads to gather the individuals on a same hunting site [23], and Continuous Interacting Ant Colony (CIAC), proposed by Dreo and Siarry [10]. Although both CACO and CIAC claim to draw inspiration from the ACO meta-heuristic, they do not follow it closely. All the algorithms add some additional mechanisms (e.g. direct communication – CIAC and API – or nest – CACO) that do not exist in regular ACO. They also disregard some other mechanisms that are otherwise characteristic of ACO (e.g. stigmergy – API – or incremental construction of solutions – all of them). CACO and CIAC are dedicated strictly to continuous optimization, while API may also be used for discrete problems.

Here a new version of the continuous ACO is used which is totally based on basic ideas of discrete ACO algorithms, namely probabilistic decision making based on stigmergic information, no direct communication and more importantly incremental solution building behavior which is subsequently used to develop improved versions of the algorithm [25].

It is proposed to find the global minimum of a positive non-zero multi-variable function within the given bounds for each variable.

$$\begin{aligned} \text{Minimize} \quad & f(x_1, x_2, \ldots, x_n) \quad \Re^n \to \Re \\ & x_i^D \leqslant x_i \leqslant x_i^U \qquad i = 1, 2, \ldots, n \end{aligned} \tag{1}$$

The first step to develop a continuous ant algorithm optimization is to define a continuous pheromone model. Although pheromone distribution has been first modeled over discrete sets, like the edge of a traveling salesman problem, in the case of real ants, pheromone deposition occurs over a continuous space. For this, consider a food source surrounded by a colony of ants. Ants aggregation around the food source causes the pheromone intensity to be higher at the food source position. Naturally, the pheromone intensity of a sample point will decrease with increasing distance from the food source. To model this variation, a Gaussian Probability Distribution Function (PDF) can be used as follows:

$$\tau(x) = \frac{1}{2\sigma\sqrt{\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2}$$

where τ is the pheromone intensity, μ the mean and σ the standard deviation of the distribution. Here μ is taken as the position of the best solution found within the interval (x_i^D, x_i^U) from the beginning of the search. Then σ can be considered as an index of ants aggregation around the current minimum. To initialize the algorithm, μ is randomly chosen within the interval (x_i^D, x_i^U), using a uniform PDF, and σ is taken equal to some integer factor of the length of the search domain, to uniformly locate the ants within the interval. Here a factor of unity is used in all the computations.

In discrete ACO algorithms, ant decision table is influenced by both pheromone intensity and heuristic information. In the absence of heuristic information, the ant decision table will be a normalized version of the pheromone intensity distribution. The pheromone model defined in Eq. (2) can, therefore, be considered as the ant decision table if the heuristic information is disregarded in CACO algorithm. This means that ants can use a Gaussian random generator as the state transition rule to choose the next point to move. The resulting transition rule selects the points around the current global best solution with higher probability reflecting the exploitative nature of the algorithm. The algorithm, however, permits all points of the search space, either close or far from the current global best solution, to be chosen with a probability depending on the ant aggregation index σ, and thus contains exploration.

In the regular ACO, pheromone updating is done via positive and negative strengthening of the pheromone at each iteration, so that it can guide the ants towards better solutions. This process traditionally consists of two actions:

– Reinforcing the probability of the choices that lead to good solution (positive update) and
– Decreasing probability of other choices by forgetting bad solutions (negative update).

In CACO algorithm, the pheromone update is a process of modifying the probability distribution used by the ants at each iteration so that the exploitative behavior of ants increases as the search proceeds. At the first iteration, there is not any knowledge about the minimum point and the ants choose their destinations only by exploration. At each iteration of the algorithm, the ants construct the solutions to the problem. Solutions are evaluated, and the global best solution is found. The new component PDF takes its mean from the value of the respective global best solution variable [30]. The standard deviation of the normal PDF is modified adaptively to increase the exploitation of the knowledge acquired about the minimum. To satisfy simultaneously the fitness and aggregation criteria, a concept of weighted standard deviation is defined as follows:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{m} \frac{1}{f_j - f_{\min}} [x_j - x_{\min}]^2}{\sum_{j=1}^{m} \frac{1}{f_j - f_{\min}}}} \tag{3}$$

where m is the number of ants, x_j the solution created by the jth ant, f_j the fitness value of the solution created by jth ant, $x_{\min}$ the current global best solution of the colony and $f_{\min}$ is the fitness value of the global best solution. This approach means that the center of region to be discovered during the subsequent iteration is the global best point and the narrowness of its width is dependent on the aggregation of the other competitors around the global best one. At each iteration the closer the solutions get to the best one, the smaller σ is assigned to the next iteration. It can be noted that since the PDF is normalized within the defined intervals (x_i^D, x_i^U), the height of distribution function increase with respect to the previous iteration as its narrowness decreases. So this strategy concurrently simulates pheromone accumulation over the promising regions and pheromone evaporation from the others, which are two major characteristics of ant colony pheromone updating rule described previously.

The algorithm can be stopped if the aggregation index σ is lower than some predefined value specified by the user, or the maximum number of iterations defined by the user is reached.

1.2. Application of continuous ant algorithm to storm water network optimization

Optimal design of storm sewer networks has been dominated by dynamic programming (DP) due to the serial nature of these problems. Robinson and Labadie [26], Kulkarni and Khanna [17], and Li and Matthew [18] all used DP for optimal design of wastewater and/or storm water networks. DP methods are applicable to small and medium scale problems but can not be used for the solution of large scale real-world problems due to the so-called curse of dimensionality. Some researchers have, therefore, attempted to use mathematical programming and heuristic methods for the solution of storm water network designs. Elimam et al. [13] used a combination of LP and a heuristic approach to design large-scale storm water networks. Miles and Heaney [22] used a heuristic method and Heaney et al. [16] employed a GA on spreadsheet templates to solve these problems. Afshar et al. [7] used a GA optimizer along with the TRANSPORT module of SWMM4.4H. to build a hydrograph based model for the optimal design of storm sewer networks. Afshar [4] used an adaptive mechanism to improve the efficiency of the discrete ACOA when solving continuous optimization problems and applied it to the optimal design of storm water networks. Afshar [5] incorporated the concept used in this work within the framework of discrete ACOA. A cellular automata approach was used by Guo et al. [15] for optimal design of storm sewer network. The method, however, was only able to adjust the pipe diameters assuming fixed slopes for the pipes. The experiments, however, showed that the cellular automata approach had great capabilities to be used for storm sewer network design.

Storm sewer network design is a multidisciplinary task involving environment, ecology, control, management and even social issues, but it is normally carried out based on several practical considerations [15]. These are based on practical experiences and sewer theories. Some of these constraints which are normally used in optimal design of sewer networks are listed below:

1. A minimum velocity constraint to prevent build-up of sediments.
2. A maximum velocity constraint to prevent erosion of pipes by sediments.
3. A minimum pipe slope to avoid negative slopes caused by inaccurate construction or settlement, and to maintain the minimum velocity constraint.
4. A minimum cover constraint to protect buried pipes from surface damage.
5. A minimum pipe size to prevent blockage of the pipes depending on the type of the sewer.
6. A minimum size for the pipe leaving a manhole as the maximum size of the pipes entering the manhole to avoid physical blockage.
7. No surcharge or pressurized flow should occur to avoid leakage of pollution through joints.
8. Selection of pipe sizes from a set of commercially available diameters.

Although various options are available to design engineers, it is found that the Rational Method was still the most popular sewer design method because of its simplicity and its general ability to meet needs [24]. It should, however, be noted that there is usually a time lag before new technologies is accepted for practice by design engineers [9]. Therefore, great motivation and ambition for a more reliable, efficient, effective and easily handled design methodology always exists to push forward studies in this area [15].

The problem of storm water network design for peak flows with fixed layout may be formulated as:

$$\text{Min. Cost} = \sum_{l=1}^{N} C(d_l, \bar{Z}_l) \tag{4}$$

$$\text{subject to}: \quad \frac{1}{n} A_l R_l^{2/3} S_l^{1/2} = Q_l^*, \quad \forall_l, \tag{5}$$

$$V_l \leqslant V_{\max}, \quad \forall_l, \tag{6}$$

$$V_l \geqslant V_{\min}, \quad \forall_l, \tag{7}$$

$$\left(\frac{y}{d}\right)_l \leqslant \alpha, \quad \forall_l \tag{8}$$

$$S_l > S_{\min}, \quad \forall_l, \tag{9}$$

$$E_l \geqslant E_{\min}, \quad \forall_l, \tag{10}$$

$$E_l \leqslant E_{\max}, \quad \forall_l, \tag{11}$$

$$d_l \in \mathbf{d}, \quad \forall_l \tag{12}$$

where A_l = wetted cross section area of link l; R_l = hydraulic radius of the link l; n = Manning constant; d_l = pipe diameter in link l; $\bar{Z}_l$ = average excavation depth for link l; Q^* = design discharge;

V_l = velocity in link l; y_l = flow depth in link l; S_l = slope of the link l; $V_{\min}$ and $V_{\max}$ = minimum and maximum velocity, respectively; α = maximum allowable ratio of the water depth, y, upon the pipe diameter, d; $E_{\min}$ and $E_{\max}$ = minimum and maximum average pipe cover, respectively; $S_{\min}$ = minimum permitted slope (more than zero in general); and N = total number of links in the network; and $\mathbf{d}$ = discrete set of commercially available pipe diameters.

In the absence of pumps and drops, this problem can be solved with different sets of decision variables such as pipe slopes, pipe diameters or nodal elevations of the network. Here the nodal elevations are used as the decision variables of the problem. For every trial solution, represented by elevations of the network nodes, the pipe slopes are first calculated assuming that each nodal elevation defines the crown elevation of pipes meeting at that node. For a trial network which satisfies the minimum slope constraint defined by Eq. (9), discrete pipe diameters are calculated to satisfy constraints (5) and (8) in the following manner. For each link in turn and starting from smallest diameter available, a normal depth is calculated for the design flow automatically satisfying constraint (5). The smallest diameter for which the constraint (8) is satisfied is taken as the diameter of the considered pipe. In case when the constraint (8) is not satisfied for the largest diameter available, the diameter of the pipe is taken as the largest possible diameter to make sure that the constraint (12) is always satisfied. The resulting solution will, therefore, automatically satisfy constraints (5) and (12) in all cases and constraint (8) in most of the cases. The resulting network is then analyzed using the known slopes and diameters to calculated the flow depth, y_l, and flow velocity, V_l, in each pipe. The resulting solution should now satisfy the remaining constraints of (6)–(8) assuming that the nodal elevations satisfy the minimum slope constraints of (9). Note that constraints (10) and (11) are easily enforced as box constraints in this formulation.

Application of ACO algorithms, as combinatorial optimization search methods, to any problem requires that the problem is defined in terms of a graph. This, though not necessary, could prove useful when applying CACOAs, too. With the nodal elevations chosen as the decision variables, the problem can be easily represented in terms of a graph $\boldsymbol{G}$ for the standard application of the algorithm referred to as Unconstrained Continuous Ant Colony Optimization Algorithm (UCACOA) for the reason to be addressed later. Each node of the network is now a decision point of the graph at which a decision has to be made regarding the corresponding nodal elevation.

In UCACOA, the available options at decision point i are represented by infinite number of elevations in the range $[E_{\min}, E_{\max}]$ at that decision point and the ants are, therefore, free to choose any of these options if required. In this formulation, the ant's decision at an arbitrary decision point is made independent of the decisions made at other decision points, i.e.; the nodal elevations are chosen independently. The sequence of decision points used to construct the trial solutions will, therefore, be immaterial in this representation and in particular can coincide with the sequence of the nodes used to define the network. The graph representation of the problem for the application of UCACOA is shown in Fig. 1, where the circles represent the bounds of the available options, elevations, at each decision point. This formulation, however, may lead to trial solutions that violate some or all of the remaining constraints (6)–(9). To discourage the ants to make decisions (i.e. select elevations) which constitute an infeasible solution, a higher cost is associated to the network that violates these constraints of the problem. This is achieved via the use of a penalty method in which the total cost of the network is considered as the sum of the network cost and a penalty cost as:

$$f(\phi) = \sum_{l=1}^{N} C(d_l, \bar{Z}_l) + \sum_{j=6,7,8,9} \alpha_j \sum_{l=1}^{N} g_{lj}^2 \tag{13}$$

in which g_{lj} is the jth constraint violation at link l and α_j represents the penalty parameter corresponding to constraint j with a large value to ensure that infeasible solutions will have a cost greater than any feasible solution.

The use of penalty method, however, has some drawbacks. The first and mostly noticed drawback is the introduction of a free parameter into the resulting problem which should be properly tuned prior to any application of the optimization algorithm. In addition, the use of the penalty method for the satisfaction of constraints could adversely affect the convergence characteristics and computational costs of the evolutionary search algorithms by enlarging the search space of the original constrained problem. In some cases the search methods even fail to find a feasible solution if the constraints are too strict, i.e.; the feasible search space is too narrow. The larger search space of the unconstrained problem often requires larger number of samples of the search space to enable the search algorithms to get a more detailed vision of the variation of the function to be optimized. This is reflected by the requirement to use larger populations in the population-based search algorithms such as genetic and ant colony optimization algorithms.

1.3. Constrained Continuous Ant Colony Optimization Algorithms

ACO algorithms enjoy a unique feature, namely incremental solution building capability, which is not observed in other evolutionary search methods 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 available options present at a decision point of the problem. This is very useful in solving optimization problems with constraints of explicit nature. These constraints, if possible, can be explicitly enforced by limiting the ant's available options to feasible ones. This idea has already been used in discrete ACO algorithms for the solution of Traveling Salesman Problem (TSP). 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. The use of a penalty parameter and, hence, tuning the penalty parameter for the best performance of the method will not be required.

In the CCACOA the aforementioned concept is used to satisfy some of the constraints of the problem, preferably the most important of these constraints. First, the constraint defined by Eq. (9) which ensures that the slope of the pipes are equal or greater than a minimum slope, is chosen. This constraint has an important effect on the convergence of the algorithm as verified later in this paper. Satisfaction of this constraint is easily achieved by some minor modification to the graph representation of the problem shown in Fig. 1. First, a different sequence of decision points is used here. Starting from the downstream end node of the network, a sequence of the decision points is created such that the decision point corresponding to the upstream node of each link of the network has a number greater than that of downstream node. This can be easily achieved by creating a tree on the network starting from the downstream node as the root. All the ants now have to start from the root and move on the tree to cover all the decision points of the graph. This is clearly different from that used in the unconstrained algorithm where ants could start from an arbitrary decision point and move to the next decision point in an arbitrary fashion.

Second, a new range of options, elevations, is constructed for each ant at decision point i satisfying constraint (9). Satisfaction of this constraint at an arbitrary decision point i requires that the slope of the link with decision point i as its upstream node is equal or more than minimum slope. The required range can, therefore, be

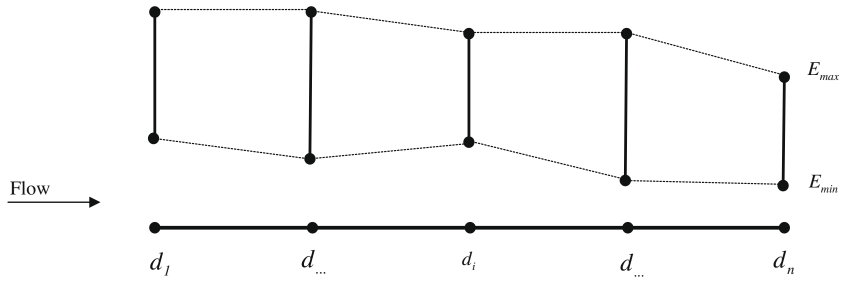

Fig. 1. Graph representation of the Storm water Network Design for Unconstrained Ant Colony Optimization Algorithm.

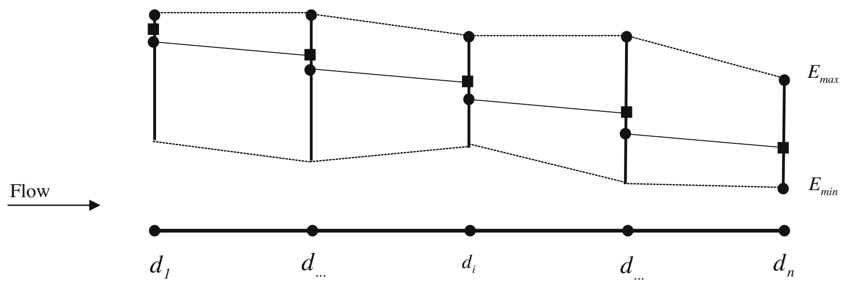

Fig. 2. Graph representation of the Storm water Network Design for Partially Constrained Ant Colony Optimization Algorithm.

easily constructed using the already determined elevation at the previous decision point which corresponds to the downstream node of the link, and the minimum slope defined in Eq. (9) by modifying the lower bound of the original range defined by $[E_{\min}, E_{\max}]$. The modified graph representation of the problem for the application of CCACO algorithm with explicit enforcement of the constraint (9) referred to as CCACOA1 is shown in Fig. 2 where the solid circles define the modified allowable range for the elevation at each decision point. The solid lines represent the minimum slope constraint boundary and the solid squares represent the option, elevation, chosen at corresponding decision point.

The new range so calculated at an arbitrary decision point *i* contains only those elevations which are above the minimum slope constraint boundary of the link with the decision point *i* as its upstream node. It can be clearly seen from Fig. 2 that the resulting search space, which is feasible with regard to the constraint defined by Eq. (9), is much smaller than the original search space. It is, therefore, expected that the resulting algorithm will perform better than the original unconstrained algorithm.

As noted above, constrained formulation of the storm water network design reduces the original unconstrained search space to a smaller search space constrained by Eq. (9). The method, however, still requires the satisfaction of constraints defined by Eqs. (6)–(8) via the penalty method. It is still possible to extend this mechanism to satisfy some other constraints of the problem and further reduce the search space size of the problem. For this, consider the constraint defined by Eq. (8). This constraint may be violated at some links of an arbitrary trial solution, defined by known elevations and consequently known slopes, when the largest available diameter is not enough to pass the corresponding design discharges at a ratio of the water depth upon the pipe diameter equal to α (see Eq. (8)). For a pipe with known diameter and known design discharge, there is a minimum slope for which the constraint (8) is satisfied. This minimum slope decreases with increasing pipe diameter. It is therefore possible to calculate for each link with known design discharge, a minimum slope, S_l^{α}, which satisfies constraint (8) for any diameter of the set **d**. This is in fact the slope at which the largest diameter can pass the design discharge at a ratio of the water depth upon the pipe diameter equal to α. Now constraint (8) can be replaced by the requirement that

$$g_{l4} \equiv S_l \geqslant S_l^{\alpha}, \quad \forall_l \tag{14}$$

This constraint can be easily enforced during the solution construction by ants as defined before for the constraint (9). Satisfaction of this constraint ensures that all the solutions created are feasible with respect to constraint (8). A second constrained ant colony optimization algorithm referred to as CCACOA2 can now be devised by explicitly enforcing both of the constraints (9) and (14). This is easily achieved if the maximum of these minimum slopes are used in the feasible space definition of the constrained Continuous Ant Colony Optimization Algorithm of Fig. 2. It should be noted, however, that the model so constructed still requires the satisfaction of velocity constraints defined by Eqs. (6) and (7) via a penalty method.

2. Model application

The performance of the proposed constrained ant colony optimization algorithms is investigated in this section by applying the model to solve a benchmark problem in the literature. The example to be considered is a problem originally designed by Mays and Wenzel [21] and solved by various investigators. The test problem includes 20 links and 21 nodes as shown in Fig. 3. Table 1 presents the characteristic data of the test problem. Other problem data including pipe roughness, pipe and excavation cost can be found in the paper by Mays and Wenzel [21]. This problem is constrained to have a maximum velocity of 12 fps (3.6 m/s), a minimum velocity of 2 fps (0.6 m/s), and a minimum cover of 8 ft (2.4 m). The value of α and $S_{\min}$ are assumed to be 0.82 and zero, respectively. Mays and Wenzel [21] first used this problem to test a Discrete Differential Dynamic Programming (DDDP) model they proposed. The problem was later solved by Robinson and Labadie [26] with a different version of the DP model. Miles and Heaney [22] also approached this problem in a spread sheet template. This problem was also solved by Afshar [4] with a discrete ACOA using

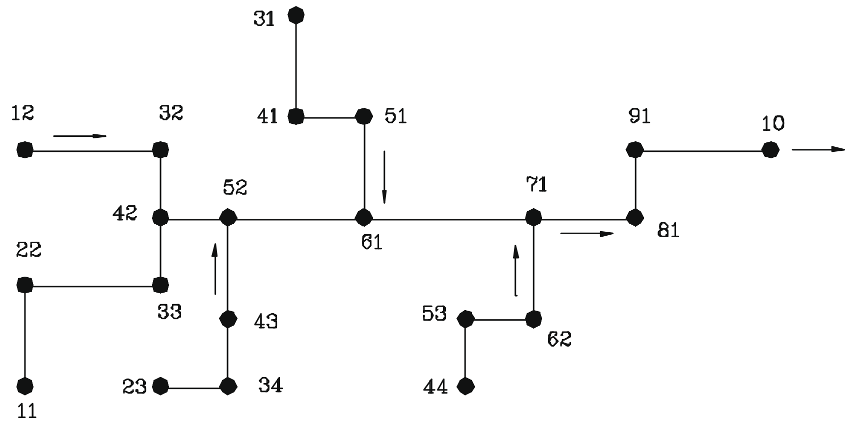

Fig. 3. Layout of the example network [5].

Table 1
Data of the benchmark example.

Link	Ground elevation (m)		Length (m)	Design discharge (cm)
	Upstream	Downstream		
1122	152.40	150.88	106.68	0.1132
2233	150.88	148.49	121.92	0.1982
3342	148.49	146.30	106.68	0.2548
1232	149.35	147.83	121.92	0.1132
3242	147.83	146.30	131.08	0.2265
4252	146.30	143.26	167.68	0.6229
2334	149.35	147.83	147.64	0.2265
3443	147.83	144.78	137.16	0.3398
4352	144.78	143.26	106.68	0.4530
5261	143.26	141.73	152.40	1.2459
3141	147.83	144.78	152.40	0.2548
4151	144.78	143.26	106.68	0.4530
5161	143.26	141.73	106.68	0.5663
6171	141.73	138.65	172.21	2.0104
4453	142.65	141.43	121.92	0.1132
5362	141.43	140.21	91.44	0.1699
6271	140.21	138.65	105.23	0.2548
7181	138.65	137.46	121.92	2.4635
8191	137.46	136.55	152.40	2.5201
9110	136.55	135.64	186.54	2.6617

an adaptive mechanism to improve the efficiency of the discrete ACOA when solving continuous optimization problems. More recently, Afshar [5] attempted this problem using discrete ACOA incorporating the concept used in this work.

One of the interesting features of the proposed continuous ant algorithm is that it only has one free parameter to tune. The problem is, therefore, solved using different colony sizes of 25, 50, 100 and 200 to assess the effect of the colony size on the performance of the algorithms. All these experiments are carried using a maximum number of 20,000 function evaluations. Figs. 4 and 5 compares the variation of the minimum and average final solution costs of 10 runs obtained by UCACOA, CCACOA1 and CCACOA2 using different colony sizes while Fig. 6 shows the variations of the standard deviations of the solutions with the colony size. It is clearly seen from Fig. 4 that the constrained algorithms have been able to locate the lowest cost solution independent of the colony size while the performance of the unconstrained algorithm is very much dependant on the colony size used. Fig. 5 shows that the average solution costs obtained with UCACOA has an oscillatory nature in terms of the colony size while averages corresponding to CCACOAs smoothly decreases with increasing size of the colony. These are in line with the results shown in Fig. 6 for the standard variation of the solutions obtained during 10 runs. In all of the cases CCACO2 algorithm in which two of the problem constrained

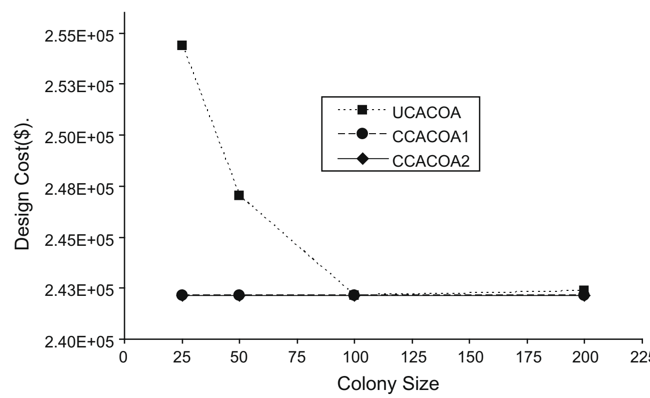

Fig. 4. Variation of the minimum solution cost with colony size for the proposed algorithms.

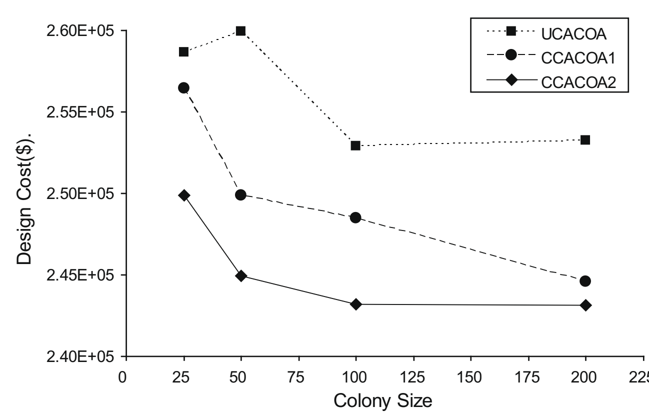

Fig. 5. Variation of the average solution cost with colony size for the proposed algorithms.

are explicitly satisfied during the solution construction performs better than CCACOA1 where only the minimum slope constraint is explicitly enforced.

Table 2 compares the Maximum, Minimum and average solution costs obtained by the proposed algorithms along with the standard deviations of the solution costs and the success rates of these algorithms. It is assumed here that all the solutions with a cost within 0.1% of the best solution cost of 242,119 could be considered as optimal. It is seen that the success rate generally increases as the number of constraints satisfied during the solution

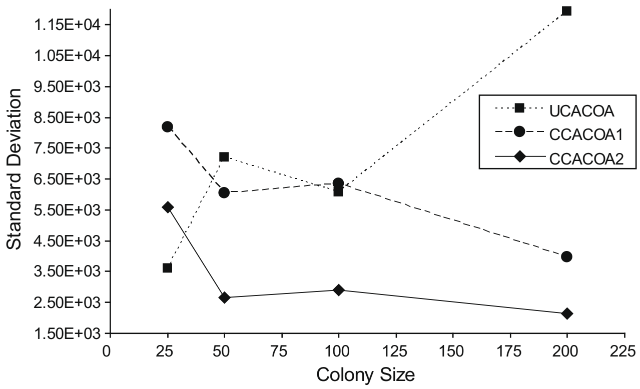

Fig. 6. Variation of the standard deviation of the final solution costs with colony size for the proposed algorithms.

Table 2
Comparison of the performance of the proposed methods for the example network for different Colony Sizes (CS).

Model	Total cost (units)			Standard deviation	Success rate
	Maximum	Minimum	Average		
UCACOA					
CS = 25	267,459	254,349	258,680	3610.32	0
CS = 50	273,924	247,034	259,942	7201.44	0
CS = 100	262,755	242,125	252,924	6075.73	2
CS = 200	281,445	242,389	253,231	11,928.14	0
CCACOA1					
CS = 25	274,096	242,123	256,468	8180.45	1
CS = 50	257,475	242,123	249,909	6037.42	3
CS = 100	259,010	242,123	248,476	6352.87	6
CS = 200	254,219	242,123	244,575	3973.81	6
CCACOA2					
CS = 25	260,782	242,123	249,887	5597.69	1
CS = 50	247,902	242,123	244,945	2673.15	5
CS = 100	247,898	242,119	243,207	2891.33	6
CS = 200	248,804	242,119	243,136	2138.15	8

Table 3
Optimal total cost obtained by different models for the example network.

Model	Best solution		Average best solution	
	Cost (units)	NFE	Cost (units)	NFE
Mays and Wenzel [21]	265,775	–	–	–
Robinson and Labadie [26]	275,218	–	–	–
Miles and Heaney [22]	245,874	–	–	–
Afshar [4]	241,496	29,900	241,513	27,625
Afshar [5]	242,539	13,900	247,327	12,758
Proposed UCACOA				
CS = 25	254,349	14,475		
CS = 50	247,034	12,550	246,474	16,581
CS = 100	242,125	19,300		
CS = 200	242,389	20,000		
Proposed UCACOA1				
CS = 25	242,123	11,900		
CS = 50	242,123	10,850	242,123	15,337
CS = 100	242,123	18,600		
CS = 200	242,123	20,000		
Proposed UCACOA2				
CS = 25	242,123	6100		
CS = 50	242,123	16,800	242,121	14,925
CS = 100	242,119	16,800		
CS = 200	242,119	20,000		

NFE: Number of Function Evaluations.

construction increases. The average success rate of UCACOA is seen to be 0.5 out of 10 while these are 4 and 5 for CCACOA1 and CCACOA2, respectively. These results clearly show the efficiency of the proposed constrained algorithm for the solution of underlying continuous optimization problem.

Table 4
Characteristics of the best solution obtained by the CCACOA2 (CS = 100).

Link	Crown elevation (m)		Diameter (mm)	Velocity (m/s)
	Upstream	Downstream		
1122	150.000	148.476	304.8	1.88
2233	148.476	146.088	381.0	2.48
3342	146.088	143.904	381.0	2.61
1232	146.952	145.412	304.8	1.76
3242	145.412	143.904	457.2	2.11
4252	143.904	140.677	533.4	3.18
2334	146.952	145.428	457.2	1.99
3443	145.428	142.380	457.2	2.98
4352	142.380	140.677	533.4	2.82
5261	140.677	138.444	762.0	3.49
3141	145.428	142.380	381.0	2.59
4151	142.380	140.856	533.4	2.69
5161	140.856	138.444	533.4	3.38
6171	138.444	136.248	914.4	3.6
4453	140.246	138.707	304.8	1.77
5362	138.707	137.808	381.0	1.81
6271	137.808	136.248	457.2	2.38
7181	136.248	135.064	1066.8	3.55
8191	135.064	133.873	1066.8	3.22
9110	133.873	132.277	1066.8	3.40

Table 3 compares the best solution costs obtained by the proposed algorithms with those produced by other researchers. The table also compares the computational effort represented by the Number of Function Evaluations (NFE) required by different algorithms to solve this example. It is first seen that 50% of solutions obtained with the proposed unconstrained algorithm is better than the solution of Miles and Heaney [22] while all of the solutions produced by the constrained algorithms outweigh the solution of Miles and Heaney [22]. In addition, all the solutions obtained using constrained algorithms are better than the best solution obtained by Afshar [5] using discrete constrained ACOAs. The solution of Afshar [4] obtained using adaptive refinement mechanism introduced into discrete ACOA, however, remains to be the best-known solution to this problem. It should, however, be noted that both methods of Afshar [4], Afshar [5] requires additional computational effort for tuning purposes due to discrete nature of used algorithm while the proposed method can avoid such tuning procedure. Furthermore, the method of Afshar [4] is seen from Table 3 to require on average two times the computational effort required by the proposed constrained methods, CCACOA1 and CCACOA2, to get the solution of nearly the same quality. The proposed methods are also shown to be virtually insensitive to the only parameter of the method namely swarm size and can, therefore, considered to be a truly parameter free optimization algorithm for continuous problems. Tables 4 shows the characteristics of the best solution obtained by the constrained methods CCACOA1 and CCACOA2.

3. Concluding remarks

Sequential nature of solution construction in Continuous Ant Colony Optimization Algorithms is used in this paper to develop two constrained Continuous Ant Colony Optimization Algorithms for the solution of storm water network design problems. In the first algorithm referred to as CCACOA1, the minimum slope constraints, zero in the example considered, of the problem are explicitly satisfied by modifying the range of the options available to each ant at each decision point which ensures that trial solutions with negative slopes are not constructed during the search process. In the second algorithm referred to as CCACOA2, the algorithm was extended to include the constraints regarding the maximum ratio of flow depth to the diameter. The applicability and efficiency of the proposed methods was tested against a benchmark example and the results compared with the original unconstrained Continuous Ant Colony Optimization Algorithm and other available results. The results show considerable improvements in the performance of the Continuous Ant Colony Optimization Algorithm regarding both convergence characteristics and the quality of the final solutions. The proposed methods, and in particular CCACOA2, was also shown to be virtually insensitive to the colony size used.

References

[1] Abbaspour KC, Schlin R, Van Genuchten MT. Estimating unsaturated soil hydraulic parameters using Ant Colony Optimization. Adv Water Resour 2001;24(8):827–933.
[2] Adami C. Introduction to artificial life. Santa Clara (CA): TELOS, The Electronic Library of Science; 1998. p. 1–15.
[3] Afshar MH. A new transition rule for ant colony optimisation algorithms: application to pipe network optimisation problems. Eng Optim 2005;37(5):525–40.
[4] Afshar MH. Improving the efficiency of ant algorithms using adaptive refinement: application to storm water network design. Adv Water Resour 2006;29:1371–82.
[5] Afshar MH. Partially Constrained Ant Colony Optimization Algorithm for the solution of constrained optimization problems: application to storm water network design. Adv Water Resour 2007;30:954–65.
[6] Afshar MH, Marino MA. Application of an ant algorithm for layout optimization of tree networks. Eng Optim 2006;38(3):353–69.
[7] Afshar MH, Afshar A, Mariño MA, Darbandi AAS. Hydrograph-based storm sewer design optimization by genetic algorithm. Can J Civ Eng 2006;33:319–25.
[8] Bilchev G, Parmee IC. The ant colony metaphor for searching continuous design spaces. In: Fogarty TC, editor. Proceedings of the AISB workshop on evolutionary computation, LNCS, vol. 993. Berlin, Germany: Springer-Verlag; 1995. p. 25–39.
[9] Burian SJ, Nix SJ, Durrans SR, Pitt RE, Fan CY, Field R. Historical development of wet-weather flow management. J Water Resour Plan Manage 1999;125(1):3–13.
[10] Dreo J, Siarry P. A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continuous functions. In: Proceedings of the third international workshop on ant algorithms. London, UK: Springer-Verlag; 2002. p. 216–21.
[11] Dorigo M, Di Caro G. The ant colony optimization meta heuristic. In: Come D, Dorigo M, Glover F, editors. New ideas in optimization. London: McGraw-HiII; 1999. p. 11–32.
[12] Dorigo M, Manielzo V, Colomi A. The ant system: optimization by a colony of cooperating ants. IEEE Trans Syst Man Cybem 1996;26:29–42.
[13] Elimam AA, Charalambous C, Ghobrial FH. Optimum design of large sewer networks. J Environ Eng, ASCE 1989;115(6):1171–89.
[14] Jun LY, Jun WT. An adaptive ant colony system algorithm for continuous-space optimization problems. J Zhejiang Univ Sci 2003;4(1):40–6.
[15] Guo Y, Walters GA, Khu ST, Keedwell E. A novel cellular automata based approach to storm sewer design. Eng Optim 2007;39(3):345–64.
[16] Heaney JP, Wright LT, Sample D, Field R, Fan C-Y. Innovative methods for the optimization of gravity storm sewer design. In: Proceedings of the 8th international conference on urban storm drainage, Sydney, Australia; 1999. p. 1896–903.
[17] Kulkarni VS, Khanna P. Pumped wastewater collection systems optimization. J Environ Eng, ASCE 1985;111(5):589–601.
[18] Li G, Matthew RGS. New approach for optimization of urban drainage systems. J Environ Eng, ASCE 1990;116(5):927–44.
[19] Ling C, Jie S, Ling O, Hongjian C. A method for solving optimization problems in continuous space using ant colony algorithm. Lect Notes Comput Sci 2002;2463:288–9.
[20] Maier HR, Simpson AR, Zecchin AC, Foong WK, Phang KY, Seah HY, et al. Ant Colony Optimization for design of water distribution systems. J Water Resour Plan Manage, ASCE 2003;129(3):200–9.
[21] Mays LW, Wenzel HG. Optimal design of multi-level branching sewer systems. Water Resour Res 1976;12(5):913–7.
[22] Miles SW, Heaney JP. Better than optimal method for designing drainage systems. J Water Resour Plan Manage, ASCE 1988;114(5):477–99.
[23] Monmarche N, Venturini G, Slimane M. On how Pachycondyla apicalis ants suggest a new search algorithm. Future Gener Comput Syst 2000;16:937–46.
[24] Pitt R, Lilburn M, Durrans SR, Burian S, Nix S, Voorhees J, et al. Guidance manual for integrated wet weather flow (WWF) collection and treatment systems for newly urbanized areas (new WWF systems). Project report, EPA/600/X-99/XXX. Cincinnati (United States): US Environmental Protection Agency; 1999.
[25] Pourtakdoust SH, Nobahari H. An extension of ant colony system to continuous optimization problems. In: ANTS Workshop; 2004. p. 294–301.
[26] Robinson DK, Labadie JW. Optimal design of urban storm water drainage system. In: International symposium on urban hydrology, hydraulics, and sediment control. Lexington (KY): University of Kentucky; 1981. p. 145–56.
[27] Simpson AR, Maier HR, Foong WK, Phang KY, Seah HY, Tan CL. Selection of parameters for Ant Colony Optimization applied to the optimal design of water distribution systems. In: Proceedings of the international congress on modelling and simulation, Canberra, Australia; 2001. p. 1931–6.
[28] Stutzle T, Hoos HH. MAX-MIN ant system. Future Gener Comput Syst 2001;16:889–914.
[29] Wodrich M, Bilchev G. Cooperative distributed search: the ant's way. Control Cybern 1997(3):413–46.
[30] Yuan B, Gallagher, M. Playing in continuous spaces: some analysis and extension of population-based incremental learning. In: Sarker R, et al., editors. Proceedings of congress of evolutionary computation (CEC); 2003. p. 443–50.
[31] Zecchin AC, Maier HR, Simpson AR, Roberts A, Berrisford MJ, Leonard M. Max–Min ant system applied to water distribution system optimization. In: Modsim 2003 – international congress on modeling and simulation, vol. 2. Townsville, Australia: Modeling and Simulation Society of Australia and New Zealand Inc.; 2003. p. 795–800.