

COLONY-MUTATED ANT SYSTEM FOR PIPE NETWORK OPTIMIZATION*

M. H. AFSHAR

Civil Engineering Faculty and Enviro-Hydroinformatic Centre of Excellence, Iran University of Science and Technology, Narmak, Tehran, I. R. of Iran
Email: mhafshar@iust.ac.ir

Abstract– A new ant algorithm, namely Colony-Mutated Ant System (CMAS), circumventing the premature convergence phenomenon is proposed in this paper and applied to pipe network optimization problems. The method uses a simple but effective mechanism, namely Pheromone Replacement Mechanism (PRM), to make sure that the global-best solution path always has the maximum trail intensity. This mechanism introduces enough exploitation into the method and, more importantly, enables one to exactly predict the number of global-best solutions at each iteration of the algorithm without the necessity of calculating the cost of the solutions created. This number is used as a measure for premature convergence of the method at each iteration. The colony is then mutated such that a predefined number of global-best solutions survive the mutation process. Two different mutation mechanisms, namely one-bit and uniform mutation are introduced and used. The probability of mutation is adjusted at each iteration so that the required number of global-best solutions survive the mutation. The method is shown to produce results comparable to Max-Min ant system (MMAS) algorithm, while requiring less free parameter tuning. The application of the method to a benchmark example in the pipe network optimization discipline is presented and the results are compared. The results indicate that the proposed CMAS method shows improved performance with improved convergence characteristics. Furthermore, the method requires less computational effort for tuning purposes due to the fewer number of free parameters compared to the MMAS method.

Keywords– Mutated, ant colony optimization, pipe networks, optimal design

1. INTRODUCTION

Ant algorithms were initially inspired by the observation that ants can find the shortest paths between food sources and their nest even though they are almost blind. Individual ants choose their paths from the nest to the food source in an essentially random fashion [1]. While walking from food sources to the nest and vice versa, however, 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 acts as a form of indirect communication called stigmergy [2], helping 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. It has been shown experimentally [3] that this pheromone trail following behavior can give rise, once employed by a colony of ants, to the emergence of the shortest paths.

Ant Colony Optimization (ACO) is a general framework for developing optimization algorithms based on the collective behaviour of ants in their search for food [4]. The searching behavior of Ant Colony Optimization Algorithms (ACOA) can be characterized by two main features [5], exploration and exploitation. Exploration is the ability of the algorithm to broadly search through the solution space, while

*Received by the editors February 20, 2008; Accepted June 21, 2011.

exploitation is the ability of the algorithm to search thoroughly in the local neighborhood where good solutions have previously been found. Higher exploitation is reflected in the rapid convergence of the algorithm to a suboptimal solution, while higher exploration results in a better solution at higher computational cost due to the slow convergence of the method. By definition, these attributes are in conflict with one another. A trade-off between exploration and exploitation in ant algorithms is, therefore, vital for a logical balance between the optimality of the solution and the efficiency of the method. To encourage exploitation, techniques have been adopted to ensure that information about the best solutions govern the search process. Bullnheimer et al. [6] suggested an elitism strategy where information about the best solution is emphasized in the algorithms' search procedure. Dorigo and Gambardella [7] used a technique to confine the search to the local neighborhood of the best solution. Dorigo et al. [8] used local optimizers to further improve good solutions. The biggest problem that can be caused by such exploitative methods is insufficient exploration and premature convergence to sub-optimal solutions. Different remedies in the form of anti-convergence techniques are suggested for premature convergence phenomena often encountered when using these exploitative methods. Dorigo and Gambardella [7] suggested a method based on discouraging the re-selection of previously selected solutions. The most notable of these methods is the Max-Min Ant System (MMAS) proposed by Stutzle and Hoos [9], in which the pheromone trails are adjusted at each iteration such that no one solution dominates the stochastic selection process. Afshar [10] has recently proposed an alternative form of the ant's stochastic decision policy which overcomes the stagnation phenomena often encountered with the algorithms using the elitist strategy. The proposed method has the advantage of not introducing a free parameter, while still producing comparable results with other anti-stagnation methods.

A new anti-stagnation method is proposed in this paper to be used with the elitist strategy of pheromone updating in ACO algorithms. The method is based on the observation that at the stagnation point, the colony is dominated by one solution which may or may not be the global best solution of the search depending on the pheromone updating procedure used. The proposed method uses a Pheromone Replacement Mechanism (PRM) to ensure that the colony is only dominated by the global-best solution when the stagnation occurs. This mechanism is advantageous as it enables one to exactly calculate the number of global-best solutions created at each iteration. The colony of solutions created at each iteration is mutated such that a predefined number of these solutions survive the mutation process. Two different mutation mechanisms, namely one-bit and uniform mutation are devised and used. The probability of mutation is calculated as a function of the predicted number of global-best solutions constructed at each iteration to introduce enough exploration into the search process. The proposed method is used here in conjunction with the ant system using elitist strategy, and hence the name Colony-Mutated Ant System (CMAS) is used for the resulting algorithm. The application of the proposed method to one of benchmark problems in the pipe network optimization literature is addressed and the results are compared with those of MMAS. The experiments show the proposed method is able to produce comparable results to that of MMAS while introducing fewer free parameters.

2. ANT COLONY OPTIMIZATION ALGORITHM

In the Ant Colony Optimization (ACO) meta-heuristic a colony of artificial ants cooperate in finding good solutions to discrete optimization problems. Application of ACO algorithm to the arbitrary combinatorial optimization problem requires that the problem can be projected on a graph (Dorigo and Gambardella 1997). Consider a graph $G = (D, L, C)$ in which $D = \{d_1, d_2, \dots, d_n\}$ is the set of decision points at which some decisions are to be made, $L = \{l_{ij}\}$ is the set of options $j=1, 2, \dots, J$ at each of the decision points $i=1, 2, \dots, n$ and finally $C = \{c_{ij}\}$ is the set of costs associated with options $L = \{l_{ij}\}$. The components of sets

D and L may be constrained if required. A path on the graph is called a solution (φ) and the minimum cost path on the graph is called the optimal solution (φ^*). The cost of a solution is denoted by $f(\varphi)$ and the cost of the optimal solution by $f(\varphi^*)$.

The basic steps on the ACO algorithms may be defined as follows [1]:

1- m ants are randomly placed on the n decision points of the problem and the amount of pheromone trail on all options are initialized to some proper value at the start of the computation.
 2- A transition rule is used for ant k at each decision point i to decide which option is to be selected. Once the option at the current decision point is selected, the ant moves to the next decision point and a solution is incrementally created by ant k as it moves from one point to the next one. This procedure is repeated until all decision points of the problem are covered and a complete solution is constructed by ant k . The transition rule used in the original ant system is defined as follows [1]:

$$p_{ij}(k,t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{j=1}^J [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta} \quad (1)$$

where $p_{ij}(k,t)$ is the probability that the ant k selects option $l_{ij}(t)$ for the i th decision at iteration t ; $\tau_{ij}(t)$ is the concentration of pheromone on option l_{ij} at iteration t ; $\eta_{ij} = 1/c_{ij}$ is the heuristic value representing the cost of choosing option j at point i , and α and β are two parameters that control the relative weight of the pheromone trail and heuristic value referred to as pheromone and heuristic sensitivity parameter, respectively. The heuristic value η_{ij} is analogous to providing the ants with sight and is sometimes called visibility. This value is calculated once at the start of the algorithm and is not changed during the computation. The role of the parameters α and β can be best described as follows. If $\alpha=0$, the cheapest options are more likely to be selected leading to a classical stochastic greedy algorithm. If on the contrary $\beta=0$, only pheromone amplification is at work, which will lead to the pre-mature convergence of the method to strongly sub-optimal solution [1].

3- The cost $f(\varphi)$ of the trial solution generated is calculated. The generation of a complete trial solution and calculation of the corresponding cost is called a cycle (k).

4- Steps 2 and 3 are repeated for all m ants of the colony at the end of which, m trial solutions are created and their costs are calculated. Generation of m trial solution and the calculation of their corresponding costs is referred to as an iteration (t).

5- The pheromone is updated at the end of each iteration. The general form of the pheromone updating used in the ant system is as follows [1]:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \quad (2)$$

where $\tau_{ij}(t+1)$ is the amount of pheromone trail on option j of the i th decision point, i.e. option l_{ij} , at iteration $t+1$; $\tau_{ij}(t)$ concentration of pheromone on option l_{ij} at iteration t ; $0 \leq \rho \leq 1$ is the coefficient representing the pheromone evaporation and $\Delta\tau_{ij}$ is the change in pheromone concentration associated with option l_{ij} . The amount of pheromone trail $\tau_{ij}(t)$ associated with option l_{ij} is intended to represent the learned desirability of choosing option j when in decision point i . The pheromone trail information is changed during the problem solution to reflect the experience acquired by ants during problem solving. The main role of pheromone evaporation is to avoid stagnation, that is, the situation in which all ants end up doing the same tour. In addition, evaporation reduces the likelihood that high cost solutions will be selected in future cycles.

Different methods are suggested for calculating the pheromone change. In the original ant system suggested by Dorigo et al. [1], all ants deposit pheromone on the options they have selected to produce the solution,

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

in which $\Delta\tau_{ij}^k$ is the pheromone deposited by ant k on option l_{ij} during iteration t . The amount of pheromone change is usually defined as [1]:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{R}{f(\varphi)^k} & \text{if option (i, j) is chosen by ant k,} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $f(\varphi)^k$ is the cost of the solution produced by the ant k and R is a quantity related to the pheromone trail called pheromone reward factor. The amount of pheromone added to each of the options during a cycle is a function of the cost of the trial solution generated. The better the trial solution and hence the lower the cost, the larger the amount of pheromone added to the option. Consequently, solution components that are used by the best ant and form a part of the lower cost solution receive more pheromone and are more likely to be selected by future ants. This choice clearly helps to direct the search towards good solutions.

At the end of each iteration, each ant has generated a trial solution. The pheromone is updated before the next iteration starts. This process is continued until the iteration counter reaches its maximum value defined by the user. A note has to be added regarding the feasibility of the solutions created by ants in constrained optimization problems. If the constraints can be explicitly defined in terms of the options available at a decision point, the ants are forced to create feasible solutions by limiting the available options to those leading to feasible solutions. In TSP, for which the ant algorithms are originally devised and tested on, the feasibility of the solution requires that each point is visited once and only once and that the finishing point is the same as the starting one. This is not, however, possible on optimization problems such as pipe network optimization problems, where the constrained are implicitly defined in terms of the options and, therefore, the feasibility of the solution is only known when the solution is totally created. In these problems, a higher total cost is usually associated with the infeasible solutions via use of a penalty function to discourage the ants to take options which constitute parts of these solutions.

3. ELITIST STRATEGIES

In the ant system described in the previous section, all the ants contribute to the pheromone change calculation defined by Eq. (3). This means that options that have been selected before will have a higher chance of selection in future iterations. This pheromone updating rule is of a highly explorative nature. The exploitation, on the other hand, is only reflected in Eq. (4) where the pheromone change caused by better solutions is calculated to be higher than other solutions. The experience shows, however, that the exploitation introduced into the method by Eq. (4) is not enough to balance the exploration present in the algorithm. This is usually reflected in slower convergence of the method or convergence to the sub-optimal solutions depending on the value of the evaporation factor used. Different methods are suggested to regulate a trade-off between the exploitation of the best solutions (iteration-best and global-best) and further exploration of the solution space. Dorigo and Gambardella [7] presented Any Colony System (ACS), which includes additional rules that probabilistically determine whether an ant is to act in an

exploitative or explorative manner at each decision point. Another mechanism used within ACS is the local updating of the pheromone of an ant's selected options immediately after it has generated its solution, such that the reselection of options within an iteration is discouraged, leading to further exploration of the method. The global updating rule in ACS is similar to that in AS, but in ACS only the path with the global-best solution receives additional pheromone. This updating rule clearly acts as an encouragement for exploitation as only the best solution is reinforced with additional pheromone. To exploit information about the global-best solution, Dorigo et al. [1] proposed the use of an algorithm known as Elitist Ant System (AS_{elite}). The updating rule in AS_{elite} is the same as that of AS, except that in AS_{elite} the global-best ant is also allowed to contribute to the pheromone change σ time at each iteration. The updating rule for AS_{elite} encourages both exploration, as each of the m solutions found by the colony receive a pheromone addition, and exploitation, as the global-best path is reinforced with the greatest amount of pheromone. As $\sigma \rightarrow \infty$, the emphasis on exploitation is greater. Another method further developing the idea of elitism is the elitist-Rank Ant System (AS_{rank}) proposed by Bullnheimer et al. [6], which involves a rank-based updating scheme. At the end of an iteration, σ elitist ants reinforce the current global-best path, as in AS_{elite} , and the ants that found the top $\sigma-1$ solutions within the iteration add pheromone to their paths with a scaling factor related to the rank of their solution. The decision rule for the AS_{rank} is the same as that for AS.

4. MAX-MIN ANT SYSTEM

Max-Min Ant System (MMAS) suggested by Stutzle and Hoos [9] is yet another method which employs the idea of elitism to introduce exploitation into the original ant system. The provision of exploitation is made in MMAS by the addition of pheromone to only the iteration-best ant's path at the end of each iteration. To further exploit good information, MMAS uses the global-best solution to update the pheromone trail at every T_{gb} iterations. The MMAS updating scheme is then given by:

$$\Delta \tau_{ij}(t) = \Delta \tau_{ij}^{ib}(t) + \Delta \tau_{ij}^{gb}(t) I_N \{t / T_{gb}\} \quad (5)$$

where, N is the set of natural numbers and $\Delta \tau_{ij}^{ib}(t)$ and $\Delta \tau_{ij}^{gb}(t)$ are the pheromone addition given by the iteration-best and global-best ants, respectively.

Premature convergence to sub-optimal solutions is an issue that can be experienced by all ACO algorithms, especially those that use an elitist strategy of pheromone updating. To overcome this problem whilst still allowing for exploitation, Stutzle and Hoos [9] proposed the provision of dynamically evolving bounds on the pheromone trail intensities such that the pheromone intensity on all paths is always within a specified range. As a result, all paths will have a non-trivial probability of being selected and thus wider exploration of the search space is encouraged. MMAS uses upper and lower bounds to ensure that pheromone intensities lie within a given range which is calculated based on some analytical reasoning. The upper pheromone bound at iteration t is given by [9]:

$$\tau_{max}(t) = \frac{1}{1 - \rho} \frac{R}{f(\varphi)^{gb}} \quad (6)$$

This expression is equivalent to the asymptotic pheromone limit of an option receiving the pheromone addition of $R/f(\varphi)^{gb}$ and decaying by a factor of $1 - \rho$ at the end of each iteration. The upper bound as defined in Eq. (6) was found to be of lesser importance, while the lower limit played a more decisive role. Stutzle and Hoos [9] introduced the following formula for the calculation of the lower trail strength limit based on some analytical arguments:

$$\tau_{\min} = \frac{\tau_{\max} \cdot (1 - p^{dec})}{(J_{avg} - 1) \cdot p^{dec}}, \quad p^{dec} = (p^{best})^{1/n} \quad (7)$$

where τ_{\min} represents the lower limit for the pheromone trail strength; p^{dec} is the probability that an ant constructs each component of the best solution again; p^{best} is the probability that the best solution is constructed again and J_{avg} is the average number of options available at the decision points of the problem. MMAS as formulated in Stutzle and Hoos [9], also incorporates another mechanism known as pheromone trail smoothing (PTS). This mechanism reduces the relative difference between the pheromone intensities, and encourages further exploration. The PTS operation performed at the end of each iteration is given by

$$\tau_{ij}(t) \rightarrow \tau_{ij}(t) + \delta(\tau_{\max}(t) - \tau_{ij}(t)) \quad (8)$$

where $0 \leq \delta \leq 1$ is the PTS coefficient. If $\delta = 0$ the PTS mechanism has no effect, whereas if $\delta = 1$ all pheromone trails are scaled up to $\tau_{\max}(t)$. In addition to these additional mechanisms, MMAS uses the same decision policy as AS.

5. PIPE NETWORK OPTIMIZATION

Due to the high costs associated with pipe networks, much research over the last decades has been dedicated to the development of methods to minimize the capital costs associated with such infrastructure. Within the last decade, many researchers have shifted the focus of pipe network optimization from traditional techniques based on linear and nonlinear programming to the implementation of heuristic methods derived from nature namely: genetic algorithm (GAs) [11-16], simulated annealing [17], and ant colony optimization (ACO) [10,18-21]. The pipe network optimization problem in its simplest form is defined as selecting the diameter of each pipe of the network so that the resulting network has a minimum cost, while meeting the required conditions. These conditions are often considered as pipe velocities and nodal pressures remaining in a pre-specified range defined by maximum and minimum velocity and pressure values. Here, each pipe is a decision point at which the diameter of the pipe is to be determined. The component of the decision set $\mathbf{D} = \{d_1, d_2, \dots, d_i, \dots, d_n\}$ is, therefore, the existing pipes of the network, where d_i represents the i th pipe of the network. The pipe diameters are usually selected from a set of commercially available diameters $\varphi = \{\varphi_{ij}\}$ which may or may not be the same for all the pipes. Assuming that these diameters are the same for all the pipes, then $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_j)$ would represent the list of available options at each and every decision point of the problem. If uc_j is defined as the per unit length cost of the pipe with diameter φ_j , the cost c_{ij} associated to the option φ_j at decision point d_i can now be calculated as the product of the per unit cost uc_j and the length le_i of the link under consideration. The cost of a trial solution $f(\varphi)$ which may or may not be a feasible solution, is now calculated as the sum of the links cost given by

$$f(\varphi) = \sum_{i=1}^n uc_j \times le_i + \alpha_p CSV \quad (9)$$

$$CSV = \left\{ \sum_{i=1}^n \left(1 - \frac{V_i}{V_{\min}}\right) + \sum_{i=1}^n \left(\frac{V_i}{V_{\max}} - 1\right) + \sum_{in=1}^{nn} \left(1 - \frac{H_{in}}{H_{\min}}\right) + \sum_{in=1}^{nn} \left(\frac{H_{in}}{H_{\max}} - 1\right) \right\} \quad (10)$$

in which n and nn is the number of existing pipes and nodes, respectively; H_{in} is the nodal head; H_{\min} and H_{\max} are minimum and maximum allowable hydraulic head; V_i is the pipe velocity; V_{\min} and V_{\max} are minimum and maximum allowable flow velocity; CSV represents a measure of the head and flow constraint violation of the trial solution and α_p is the penalty parameter with a large enough value to ensure

that any infeasible solution will have a higher total cost than any feasible solution. It should be noted that in calculating the CSV , the summation ranges over those nodes and pipes at which a violation of pressure and velocity constraints occurs, ie; the terms in parenthesis are positive. Here, the penalty parameter is taken as the cost of the most expensive network, ie; a network with all its pipes having the largest possible diameter. For a given network, the nodal pressures and pipe velocities are obtained via the use of a simulation program that explicitly solves the set of hydraulic constraints for nodal heads [22]. This, however, requires the definition of some parameters in the Hazen-Williams equation which states the relation between head loss and flow in each link. Here, a Hazen-Williams formula of the type

$$h_f = \mu L \left(\frac{Q}{C} \right)^\lambda D^{-\gamma} \quad (11)$$

is used, in which L = length of a pipe; Q = flow rate of a pipe; C = Hazen-Williams coefficient, D = internal diameter of a pipe and: $\lambda = 1.852$, $\gamma = 4.871$, $\mu = 10.667$ for Q in cubic meter per hour and D in inch (equivalent to $\mu = 4.727$ for D in feet and Q in cubic feet per second) are Hazen –Williams constants as used in EPANET 2.0.

6. PROPOSED COLONY-MUTATED ANT SYSTEM (CMAS)

MMAS, as defined above, suffers from some shortcomings. Firstly, the argument behind MMAS is based on the strong assumption that around good solutions other good or even better solutions are located. This is definitely the case for TSP, the problem for which the MMAS is proposed as it is shown that reasonably good tours are located in a small region of the search space. This is not necessarily true for other problems such as pipe network optimization problems in which good solutions may be surrounded by costly infeasible solutions. The second is that the trail limits, and in particular the lower limit, used in MMAS will effectively come into play when a best found solution dominates the colony to encourage the ant to create some other solutions using the components of this solution. When an elitist strategy is used for pheromone updating, the trail intensities on all the options available at an arbitrary decision point is nearly zero except for the option corresponding to the best found solution. MMAS calculates the lower bound of the trail intensities for a given value of p^{best} and raises the near-zero value of all options to this value. At this moment, all the options except one will have the same non-zero trail intensity. This will, of course, increase the chance of other options to constitute part of the next iteration solutions, but in a random fashion. The ants will be required to take a random walk in an artificially widened search space around the dominating solution. And finally, the MMAS introduces some additional free parameters such as p^{best} , T_{gb} , and δ in addition to α, β, ρ, Q and m which are used by all ACO algorithms. While some heuristics are derived for the second set of parameters [20], the setting of the second set is subject to trial and error. The value of these parameters should be tuned for the best performance of the algorithm prior to the main application of the method. This, of course, adds to the computational requirement of MMAS compared to that of the original ant system.

To introduce the proposed method, first consider the role of the additional parameters p^{best} , T_{gb} , and δ used in MMAS. Parameters p^{best} and δ are both meant to introduce exploration into the algorithm as defined earlier. The exploration increases with the decreasing value of p^{best} and the increasing value of δ . These parameters are not, however, independent. Assuming that the PTS operation defined by Eq. (8) is followed by the implementation of Eq. (7) using predefined p^{best} , then it is highly probable that for large enough values of PTS parameter, δ , the smoothed pheromone trails calculated by Eq. (8) are higher than the lower bound τ_{min} defined by Eq. (7), leading to the redundancy of this equation. If, on the other hand, the PTS operation is preceded by the implementation of p^{best} , then the PTS mechanism leads to a mere

constant scaling of the calculated minimum pheromone trails τ_{\min} on the options which do not constitute a part of the dominating solution. This effect can be clearly achieved by using a lower value of p^{best} without having to use PTS mechanism. It can therefore be argued that only one of these two mechanisms is needed to introduce the required exploration into MMAS. The parameter p^{best} has the advantage of easier setting as it carries a physical meaning, ie; the probability that the best solution is created by all ants. It is therefore reasonable to disregard the PTS operation by assuming a value of zero for δ and only tune p^{best} for balancing the exploitation and exploration of the MMAS.

Now consider the effect of T_{gb} as used in Eq. (5). This equation states that the global-best path should be reinforced every T_{gb} iteration. For a very large value of this parameter, only iteration-best solutions are used to update the pheromone trail. In this situation it is possible that the search does not converge on a single solution or otherwise converge to a solution different from the global-best solution, depending on the value of evaporation factor ρ used. For the values of ρ close to 1.0, MMAS may fail to converge and for small enough values of ρ the stagnation at sub-optimal solution may occur. In the first case, implementation of Eq. (7) will be redundant since this mechanism comes into effect when stagnation starts to take place. Implementation of Eq. (7) in the second case will lead to a search around a sub-optimal solution which will clearly be inefficient. Small values of T_{gb} with a minimum value of one result in higher exploitation of the global-best solution, which is often reflected in the colony being dominated by the current global-best solution. In other words, the role of the T_{gb} is merely to ensure that the path with maximum pheromone intensity corresponds to the current global-best solution at all stages of the search.

The proposed Colony-Mutated Ant System (CMAS), which uses the same decision policy as that of AS and an elitist pheromone updating rule in which only iteration-best solutions are reinforced at each iteration, introduces a new balancing exploitation and exploration mechanism. As an exploitation mechanism which ensures that the algorithm only converges to the GBS, CMAS uses a simple but effective parameter-free Pheromone Replacement Mechanism (PRM) in which the pheromone intensity of the GBS is replaced with that of the path defined by the maximum pheromone intensity and vice versa whenever a new GBS is located. This will guarantee that the current global-best solution has the maximum pheromone trail and, therefore, has a very high chance of being selected as the iteration-best solution of the iteration to be used in the pheromone updating process.

The proposed CMAS uses an explorative feature to balance the exploitation embedded in the algorithm via use of PRM to replace the lower bound scaling (Eq. (7)) of MMAS. This is achieved using the mutation mechanism commonly used in GAs on the colony created at each iteration once the stagnation is started. Two mutation procedure, one bit-wise and the other uniform mutation, are introduced and used here. In the first one, a one-bit mutation is carried out with a probability P_m defined as

$$P_m = 1 - \left(m \frac{P_{gb}}{M_{gb}} \right) \quad (12)$$

While in the second method, the colony undergo a uniform mutation with a probability P_m defined as

$$P_m = 1 - \left(m \frac{P_{gb}}{M_{gb}} \right)^{1/n} \quad (13)$$

where n denotes the number of decision points of the problem; m is the colony size as defined earlier; P_{gb} is the ratio of the number of global-best solutions surviving the mutation to the colony size defined by the user and M_{gb} is the number of global-best solutions created by the ants at the current iteration. The probability of mutations so calculated ensures that, on average, $m.P_{gb}$ of global-best solutions survive the

mutation process. It can be seen that the mutation mechanism is activated only when $M_{gb} > m \cdot P_{gb}$ in both of the methods. It should be noted that P_{gb} carries a meaning similar to that of P_{best} used in MMAS.

7. TEST EXAMPLE

The test problem considered here concerns the rehabilitation of the New York City water supply network with 21 pipes, 20 demand nodes, and one reservoir as shown in Fig. 1 [11]. The commercially available pipe diameters and their respective costs are listed in Table 1 while the pipe and nodal data of the existing network are shown in Table 2. The last column of the table represents the minimum head requirement at each node corresponding to the minimum pressure of 30 meters. The minimum head requirement is the only constraint of the problem as defined in Dandy et al. [11]. This table is augmented by a virtual zero-diameter cost equal to the half of the cheapest diameter to enable the calculation of local heuristics for all available options. This problem has been used as a case study by many researchers using genetic algorithm [11-16, 23] and most recently by Maier et al. [18]; Zecchin et al [19, 20] and Afshar [10, 21] using ACO algorithms.

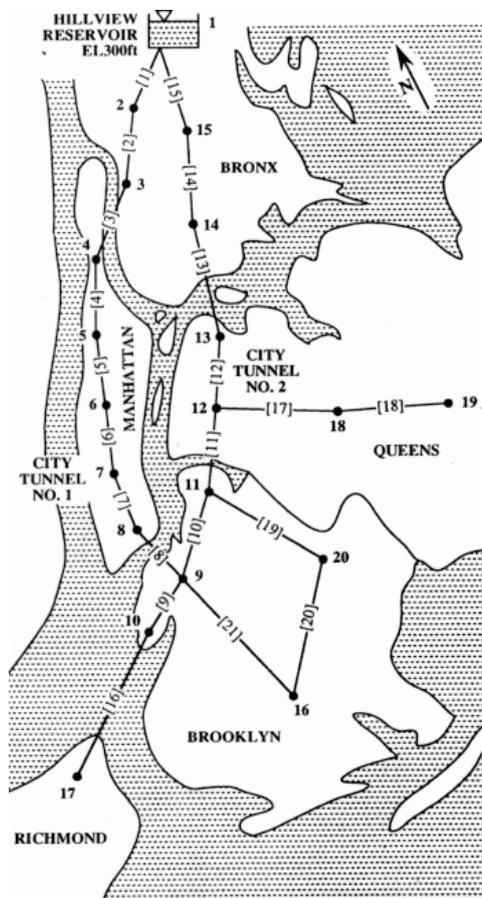


Fig.1. New York tunnel network

Table 1. Pipe cost data for New York network

| | | | | | | | | |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Diameter (inch) | 0 | 36 | 48 | 60 | 72 | 84 | 96 | 108 |
| Cost (\$/ft) | 0 | 93.5 | 134.0 | 176.0 | 221.0 | 267.0 | 316.0 | 365.0 |
| Diameter (inch) | 120 | 132 | 144 | 156 | 168 | 180 | 192 | 204 |
| Cost (\$/ft) | 417.0 | 469.0 | 522.0 | 577.0 | 632.0 | 689.0 | 746.0 | 804.0 |

Table 2. Pipe and nodal data for New York tunnel network.

| Pipe data | | | | | Nodal data | | |
|-----------|------------|----------|---------------|---------------------------|------------|-----------------|------------------|
| Pipe | Start Node | End Node | Length (m) | Existing Diameter (mm) | Node | Demand (l/s) | Min. Head (m) |
| 1 | 1 | 2 | 3535.6 | 4572 | 1 | reservoir | 91.4 |
| 2 | 2 | 3 | 6035.0 | 4572 | 2 | 2616 | 77.72 |
| 3 | 3 | 4 | 2225.0 | 4572 | 3 | 2616 | 77.72 |
| 4 | 4 | 5 | 2529.8 | 4572 | 4 | 2497 | 77.72 |
| 5 | 5 | 6 | 2621.2 | 4572 | 5 | 2497 | 77.72 |
| 6 | 6 | 7 | 5821.6 | 4572 | 6 | 2497 | 77.72 |
| 7 | 7 | 8 | 2926.0 | 3353 | 7 | 2497 | 77.72 |
| 8 | 8 | 9 | 3810.0 | 3353 | 8 | 2497 | 77.72 |
| 9 | 9 | 10 | 2926.0 | 4572 | 9 | 4813 | 77.72 |
| 10 | 11 | 9 | 3413.7 | 5182 | 10 | 28 | 77.72 |
| 11 | 12 | 11 | 4419.6 | 5182 | 11 | 4813 | 77.72 |
| 12 | 13 | 12 | 3718.5 | 5182 | 12 | 3315 | 77.72 |
| 13 | 14 | 13 | 7345.6 | 5182 | 13 | 3315 | 77.72 |
| 14 | 15 | 14 | 6431.2 | 5182 | 14 | 2616 | 77.72 |
| 15 | 1 | 15 | 4724.4 | 5182 | 15 | 2616 | 77.72 |
| 16 | 10 | 17 | 8046.7 | 1829 | 16 | 4813 | 79.25 |
| 17 | 12 | 18 | 9509.7 | 1829 | 17 | 1628 | 83.15 |
| 18 | 18 | 19 | 7315.2 | 1524 | 18 | 3315 | 77.72 |
| 19 | 11 | 20 | 4389.1 | 1524 | 19 | 3315 | 77.72 |
| 20 | 20 | 16 | 11704.3 | 1524 | 20 | 4813 | 77.72 |
| 21 | 9 | 16 | 8046.7 | 1829 | | | |

To pave the way for the introduction and implementation of the proposed CMAS on the test example, a series of experiments are carried out to assess the effect of MMAS parameters on its performance as described previously.

An experiment is first carried out to verify the role of T_{gb} in MMAS described previously. The example problem is solved with different values of $T_{gb}=1, 10$, and ∞ for fixed values of other parameters $\alpha = 1, \beta = 0.25, \rho = 0.98, m = 50$ and $p^{best} = 1.0$. These values are chosen following heuristics suggested by Zecchin et al. [20] and some preliminary runs. Figures 2 to 5 show the variation of the averaged number of global-best solutions (GBS) and maximum pheromone intensity solutions (MPIS) during the search process for different values of T_{gb} obtained from ten runs using different initial colonies. It is clearly seen from Fig. 2 that for a large value of $T_{gb}=\infty$, the number of GBS and MPIS are different during the search process. The difference increases as the solution corresponding to the maximum pheromone intensity dominates the colony. This difference indicates that a pheromone updating rule which only uses iteration-best solutions may lead to domination of a solution different from the global-best solution. It is obvious that implementation of Eq. (7) with $p^{best} < 1$ will be inefficient in this situation. The difference between the number of GBS and MPIS decreases with the decreasing value of T_{gb} as illustrated in Figs. 3 and 4. It can, therefore, be argued that the main effect of reinforcing the global-best path in MMAS is to make sure that the solution corresponding to maximum pheromone intensity is the current global-best solution of the search. In this situation, implementation of Eq. (7) with $p^{best} < 1$ will result in a colony of solutions constructed on and around the global-best solution of the search. This, of course increases the chance of improving the current GBS compared to a situation in which the colony is constructed on and around an inferior solution, a situation which happens for larger values of T_{gb} . It is also instructive to note that small values of T_{gb} (reinforcing the global-best path more often) will result in more exploitation which is reflected in faster stagnation of the search, an effect similar to that expected from evaporation factor. This means that both the evaporation factor ρ and T_{gb} play an exploitative role in MMAS. A successful implementation of the algorithm, therefore, requires a careful tuning of these

parameters to ensure that a) the colony have enough time to explore the search space before domination of MPIS and b) the MPIS is the same as GBS so that the colony is dominated by the current GBS and not any other inferior solution.

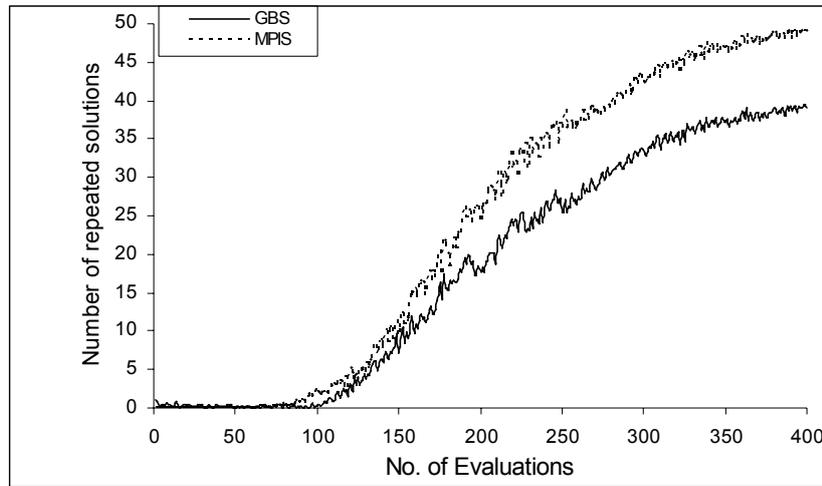


Fig. 2. Variation of the average number of GBS and MPIS with the number of iterations for ten runs ($T_{gb}=\infty$)

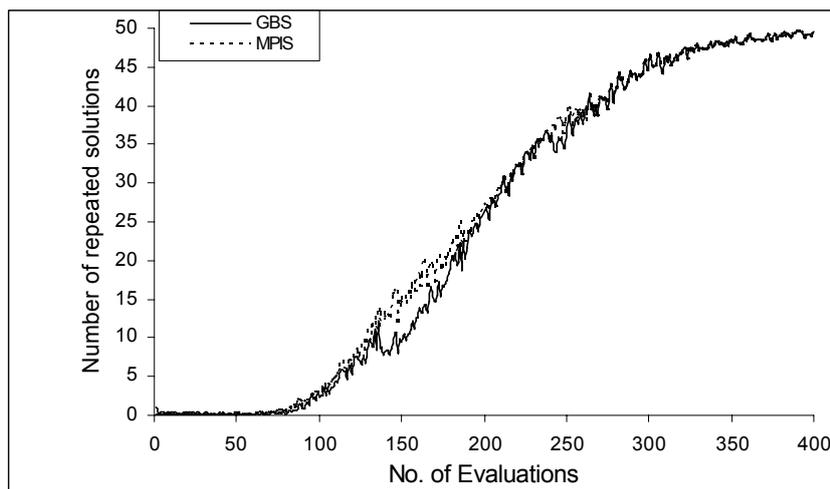


Fig. 3. Variation of the average number of GBS and MPIS with the number of iterations for ten runs ($T_{gb}=10$)

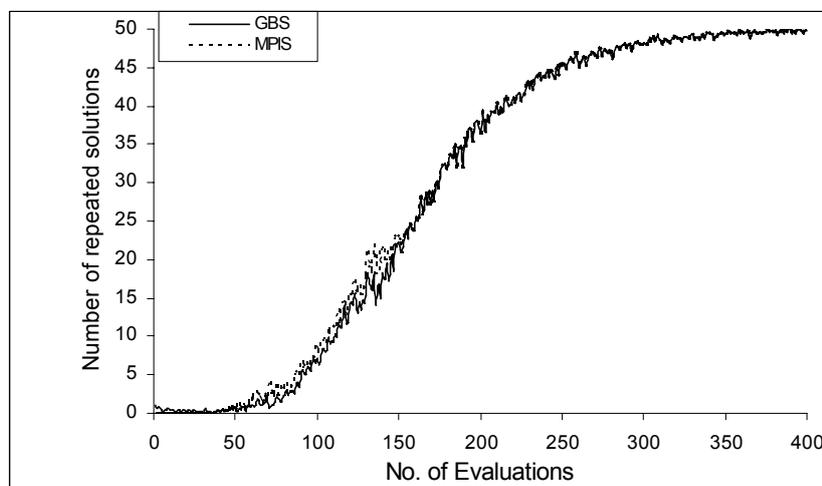


Fig. 4. Variation of the average number of GBS and MPIS with the number of iterations for ten runs ($T_{gb}=1$)

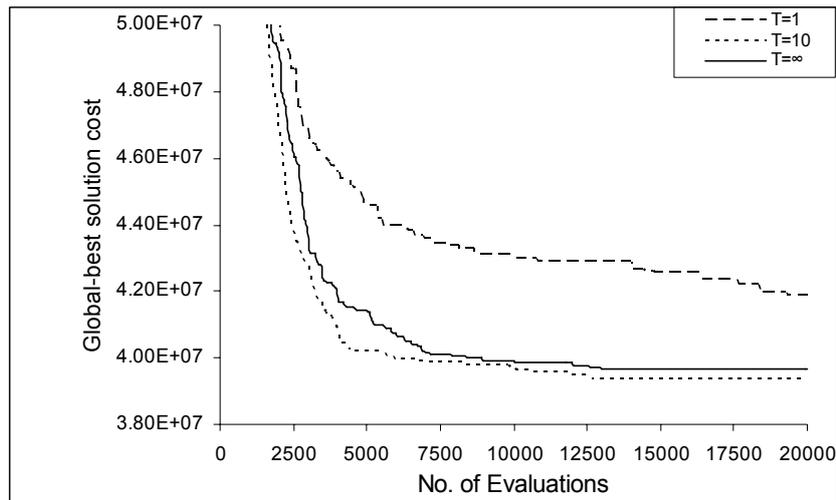


Fig. 5. Variation of the average GBS cost for different values of T_{gb}

It is instructive to see the performances of MMAS for different values of parameter T_{gb} . Fig. 5 shows the variation of the average GBS costs of ten runs using different initial colonies for $T_{gb}=1, 10, \infty$ and $p^{best}=0.05$. The best performance of the MMAS is achieved for $T_{gb}=10$ in terms of convergence characteristics and the quality of the solution. The algorithm shows the worst performance for $T_{gb}=1$ due to higher exploitation, which is not balanced by the exploration introduced via use of $p^{best}=0.05$. MMAS using $T_{gb}=\infty$ shows not only inferior, though close, convergence behavior to MMAS using $T_{gb}=10$, but also a lower success rate of 1 in ten runs in locating the global solution of the problem, with a cost of \$38.63M, compared to the success rate of 3 achieved by the latter. This can be attributed to the fact that in the latter case, the maximum pheromone intensity path does not correspond to the GBS in all the ten runs as shown earlier in Fig. 2. To complete the observations, another experiment is carried out to examine the convergence behavior of the MMAS algorithm for $\rho=1$, $p^{best}=1$ and $T_{gb}=1, 10$ and ∞ . The results, not shown here, indicated that irrespective of the level of exploitation, regarding the value of T_{gb} , the algorithm is not convergent when no evaporation is present ($\rho=1$). For all values of T_{gb} used, the average number of GBS and MPIS was always below 2% of the colony size at all stages of the search. It is obvious that the introduction of further exploration via implementation of Eq. (7) with $p^{best}<1$ will be redundant in this situation. It can, therefore, be argued that in MMAS, the evaporation ($\rho<1$) guarantees the convergence, reinforcement of GBS with a proper value of T_{gb} , ensures that the algorithm converges on the GBS and, finally, the adjustment of the lower pheromone bound with $p^{best}<1$ enlarges the search space around the GBS providing the opportunity for the ants to further improve the current GBS.

An experiment is now carried out to verify the effectiveness of the proposed PRM. Figure 6 shows the average number of GBS and MPIS of ten runs versus the number of iterations for three values of evaporation factor $\rho=1, 0.995$, and 0.99 with other parameters chosen as $\alpha=1, \beta=0.25, m=50$, and $p^{best}=1.0$. It should be noted that each curve in Fig. 6 is representative of both the number of GBS and MPIS as these have been found to be virtually the same. It is interestingly seen that the PRM introduces enough exploitation into the algorithm, even when no evaporation, $\rho=1$, is introduced into the algorithm. The algorithm shows faster stagnation with decreasing values of evaporation factor as expected. The algorithm, however, has enough chance to explore the search space before stagnation starts when no evaporation is used. The proposed PRM seems to be very advantageous as it simulates the effect of both GBS reinforcement and evaporation without introducing any free parameter. It can, therefore, be expected that PRM with little or no evaporation perform better as the resulting search process will have enough time to explore the search space before stagnating at the current global-best solution. This expectation is

indeed fulfilled as shown in Fig. 7 where the average GBS cost is seen to decrease with the increasing value of the evaporation factor. The minimum average solution cost, in fact, is obtained when no evaporation is used. The proposed PRM, therefore, ensures enough exploitation and convergence of the method to the GBS solution irrespective of the amount of evaporation used. The averaged GBS costs and the success rate of the algorithm for the values of evaporation factor $\rho=1, 0.995, \text{ and } 0.99$ were (\$39.61M, 2), (\$39.82M, 1), and (\$39.93M, 1), respectively.

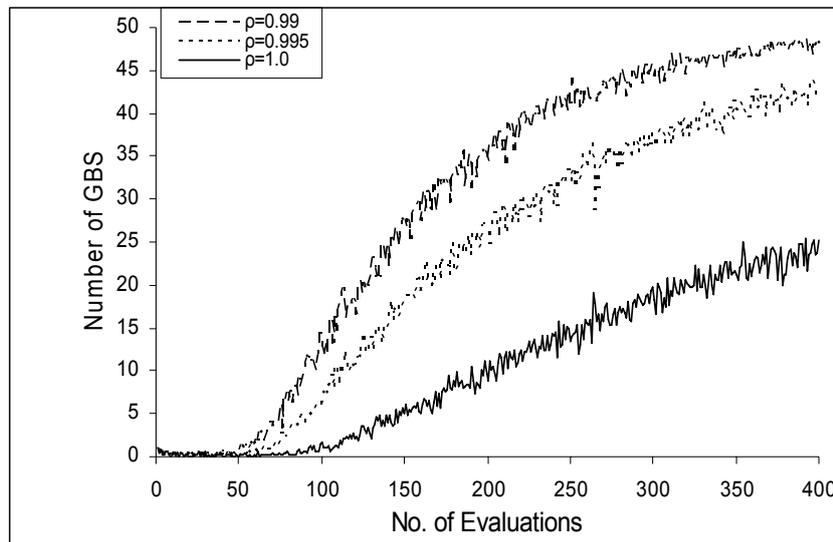


Fig. 6. Variation of the average number of GBS during the search for different values of evaporation factor using PRM

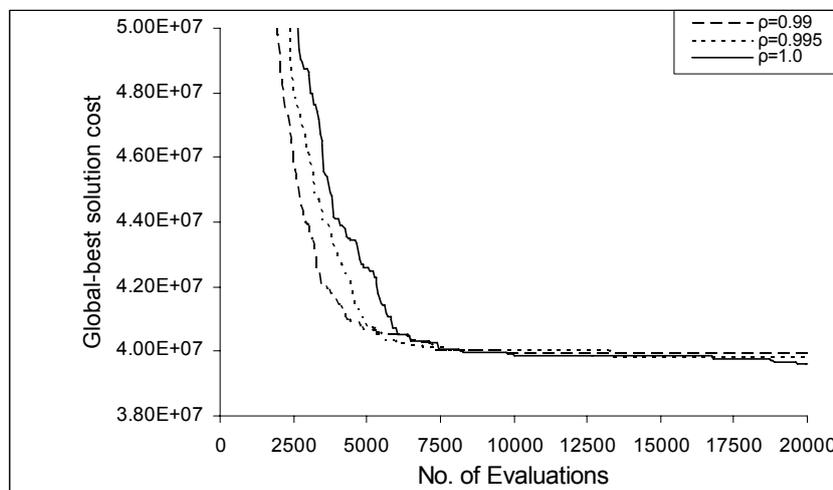


Fig. 7. Variation of the average GBS cost for different values of evaporation factor using PRM

The final experiment is now carried out to verify the efficiency of the proposed mutation mechanisms. Figure 8 compares the variation of the average GBS cost using first and second mutation mechanisms denoted by CMAS1 and CMAS2, respectively, with that of the best performing MMAS. CMAS results were obtained using five parameter values $\alpha = 1, \beta = 0.25, \rho = 1.0, P_{gb} = 0.05, m = 50$, while MMAS required the tuning of six parameters as: $\alpha = 1, \beta = 0.25, \rho = 0.98, T_{gb} = 10, p_{best} = 0.05, m = 50$. Considering the exploitative behavior of CMAS with no evaporation, there is actually no need to tune for the evaporation factor. The number of free parameters of EMAS, therefore, reduces to four compared to six for MMAS.

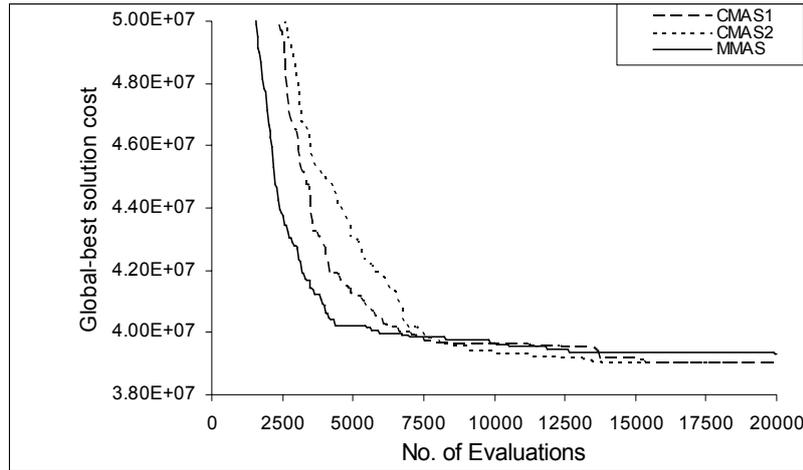


Fig. 8. Variation of the average GBS cost for the best performing MMAS and proposed CMAS

It is seen that the average GBS cost obtained by both CMAS1 (\$39.03M) and CMAS2 (\$39.00M) is considerably superior to that of MMAS (\$39.32M). The CMAS1, however, showed a higher success rate of 4 out of ten in locating the optimum solution of \$38.64M compared to that of 3 out of ten for MMAS, while CMAS2 showed a poorer performance with a success rate of 2 out of ten. It is obvious that the mutations introduced are responsible for improving the average GBS cost and the success rate of the PRM from (\$39.61M,2) to (\$39.03M,4) and (\$39.00M,2) obtained by the CMAS1 and CMAS2, respectively. It is also instructive to compare the number of average global-best solutions for three algorithms as shown in Fig. 9. It is clearly seen that both of the mutation mechanisms used in CMAS were successful to control the number of GBS around $m.P_{gb}=2.5$, while this number is very high for MMAS. This is, in fact, another feature of the proposed CMAS, enabling the method to compete with MMAS using fewer tuning parameters. The proposed CMAS is therefore computationally less demanding than MMAS while producing comparable results.

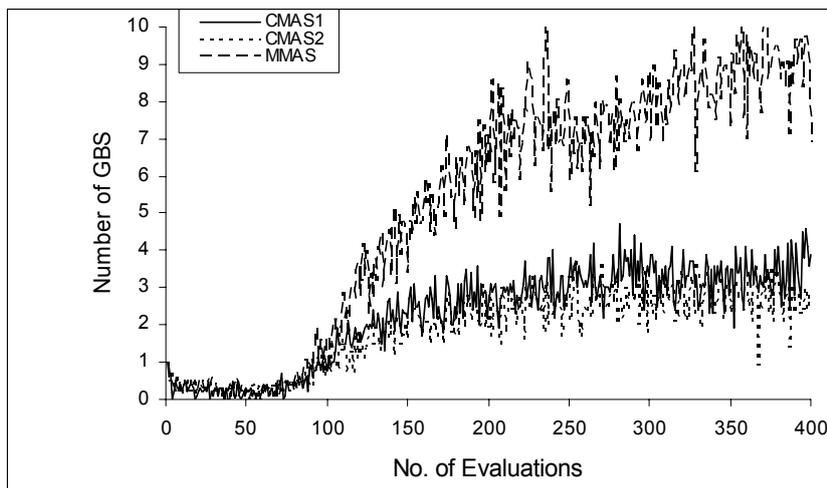


Fig. 9. Variation of average number of GBS for best performing MMAS and proposed CMAS

The experiments carried out on the test example shows that the advantages of the proposed CMAS over the commonly used MMAS is threefold. First, the proposed method has shown overall better performance than the MMAS by producing lower cost solutions in ten runs. In fact, the CMAS1 and CMAS2 were able to produce the best solution with average costs of \$39.03M and \$39.00M, respectively compared to the averaged best solution cost of \$39.32M obtained by MMAS. It should, however, be noted

that all methods have been able to produce the known optimal solution of \$38.64M for the test example [11]. The second advantage of the proposed method over the MMAS lies in its convergence rate, as evident from Fig. 8. In fact, all the ten runs carried out by the CMAS1 and CMAS2 converged within 15350 and 13850 function evaluations, respectively, while the MMAS required the maximum number of function evaluations of 20000 to converge. This emphasizes the efficiency of the proposed CMAS compared to MMAS. And finally, the proposed CMAS methods only require the proper value for the four free parameters of α, β, P_{gb}, m compared to the six free parameters of $\alpha, \beta, \rho, T_{gb}, p_{best}, m$ required by the MMAS. This means the proposed CMAS methods require less computational effort for tuning the free parameters of the method compared to the MMAS.

8. CONCLUDING REMARKS

A new ACO algorithm was presented as an alternative to the Max-Min Ant System. The method exploits automatically balanced exploitative and explorative features. The exploitation of the method is provided by a simple but effective free-parameter procedure in which the global-best solution pheromone intensity is replaced by the current maximum pheromone trail each time the global-best solution is updated. This procedure was shown to introduce enough exploitation into the method ensuring the convergence of the search to the global-best solution, irrespective of the value of the evaporation factor. The method offers the advantage of exactly predicting the number of global-best solutions of the iteration without the necessity of calculating the cost function of the trial solutions. Two mutation mechanisms were then used on the colony at each iteration to introduce balancing exploration into the algorithm. The first mutation mechanism uses a one-bit mutation, while a uniform mutation is used in the second one. The probability of mutations are adjusted at each iteration such that a predefined number of global-best solution, survive the mutation process. The proposed algorithm was tested against a benchmark example in the water distribution network optimization literature and the results compared with that of MMAS. The results show that the proposed algorithm produces solutions comparable to those of MMAS, while introducing fewer free parameters to be tuned.

REFERENCES

1. Dorigo, M., Manielzo, V. & Colomi, A. (1996). The ant system: optimization by a colony of cooperating ants. *IEEE Trans. Syst. Man Cybern.*, Vol. 26, pp. 29-42.
2. Dorigo, M., Bonabeau, E. & Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Generation Comput. Systems*, Vol. 16, pp. 851-87.
3. Deneubourg, J. L., Aron, S., Goss, S. & Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of Insect behavior*, Vol. 3, pp. 159-168.
4. Dorigo, M. & Di Caro, G. (1999). *The ant colony optimization meta heuristic. New ideas in optimization*. D. Come, M. Dorigo, and F. Glover, eds., McGraw-Hill, London, pp. 11-32.
5. Colomi, A., Dorigo, M., Maffioli, F., Maniezzo, V., Righini, G. & Trubian, M. (1996). Heuristics from nature for hard combinatorial optimization problems. *International Transactions in Operational Research*, Vol. 3, No. 1, pp. 1-21.
6. Bullnheimer, B., Hartl, R. F. & Strauss, C. (1999). A new rank based version of the Ant System: A computational study. *Central European Journal for Operation Research and Economics*, Vol. 7, No. 1, pp. 25-38.
7. Dorigo, M. & Gambardella, L. M. (1997). A cooperative learning approach to TSP. *IEEE Transaction on Evolutionary Computation*, Vol. 1, No. 1, pp. 53-66.

8. Dorigo, M. & Di Caro, G. & Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial Life*, Vol. 5, No. 2, pp. 137-172.
9. Stutzle, T. & Hoos, H. H. (2000). MAX-MIN ant system. *Future Generation Comput. Systems*, Vol. 16, pp. 889-914.
10. Afshar, M. H. (2005). A new transition rule for ant colony optimization algorithms: application to pipe network optimization problems. *Engineering Optimization*, Vol. 37, No. 5, pp. 525-540.
11. Dandy, G. C., Simpson, A. R. & Murphy, L. J. (1996). An improved genetic algorithm for pipe network optimization. *Water Resources Research*, Vol. 32, No. 2, pp. 449-458
12. Halhal, D., Walters, G. A. Ouazar, D. & Savic, D. A. (1997). Water network rehabilitation with structured messy genetic algorithm. *J. Water Resour. Plan. Manage.*, Vol. 123, No. 3, pp. 137-146.
13. Savic, D. A. & Waters, G. A. (1997). Genetic algorithms for least-cost design of water distribution networks. *Water Res. Planning and Management, ASCE*, Vol. 123, No. 2, pp. 67-77
14. Walters, G. A., Halhal, D., Savic, D. & Quazar, D. (1999). Improved design of anytown distribution network using structured messy genetic algorithms, *Urban Water*, Vol. 1, No. 1, pp. 23-38.
15. Wu, Z. Y. & Simpson, A. R. (2002). A self-adaptive boundary search genetic algorithm and its application to water distribution systems. *Journal of Water Research*, Vol. 40, No. 2, pp. 191-203.
16. Zheng, Wu, Boulos, P. F., Orr, C. H. & Ro, J. J. (2001). Using genetic algorithms to rehabilitate water distribution systems. *Journal of AWWA*, Vol. 93, No. 11, pp. 74-85.
17. Cuhna, M. C. & Sousa, J. (1999). Water distribution network design optimization: Simulated annealing approach. *Journal of Water Resources Planning and Management, ASCE*, Vol. 125, No. 4, pp. 215-221.
18. Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y. & Tan, C. L. (2003). Ant colony optimization for design of water distribution systems. *J. Water Resour. Plan. Management, ASCE*, Vol. 129, 3, pp. 200-209.
19. Zecchin, A. C., Maier, H. R., Simpson, A. R., Roberts, A., Berrisford, M. J. & Leonard, M. (2003). Max-min ant system applied to water distribution system optimization. *Modsim 2003 - International Congress on Modelling and Simulation*, Modelling and Simulation Society of Australia and New Zealand Inc, Townsville, Australia, Vol. 2, pp.795-800.
20. Zecchin, A. C., Simpson, A. R., Maier, H. R. & Nixon, J. B. (2004). Parametric study for an ant algorithm applied to water distribution system optimization. *IEEE Transaction on Evolutionary Computation*, in press.
21. Afshar, M. H. (2007). Application of Ant Algorithm to Pipe network Optimisation. *Iranian Journal of Science & Technology, Transaction B: Engineering*, Vol. 31, B5, pp. 487-500.
22. Afshar, M. H. (2001). An Element - by - Element Algorithm for the Analysis of Pipe Networks. *Int. J. for Eng. Science*, Vol. 12 No. 3, pp. 87-100.
23. Lippai, I., Heany, J. P. & Laguna, M. (1999). Robust water system design with commercial intelligent search optimizers. *J. Comput. Civ. Eng.*, Vol. 13, No. 3, pp. 135-143.