Elitist Continuous Ant Colony Optimization Algorithm: Application to Reservoir Operation Problems

Mohammad Hadi Afshar[1], H. Ketabchi[2], E. Rasa[3]

Department of Civil Engineering, Iran University of Science and Technology, Tehran, Iran.
[1]Assistant Professor, Email: mhafshar@iust.ac.ir
[2]Graduated Student, Email: hketabchi@civileng.iust.ac.ir
[3]Graduated Student, Email: erasa@civileng.iust.ac.ir,

Abstract: *In this paper, a new Continuous Ant Colony Optimization (CACO) algorithm is proposed for optimal reservoir operation. The paper presents a new method of determining and setting a complete set of control parameters for any given problem, saving the user from a tedious trial and error based approach to determine them. The paper also proposes an elitist strategy for CACO algorithm where best solution of each iteration is directly copied to the next iteration to improve performance of the method. The performance of the CACO algorithm is demonstrated against some benchmark test functions and compared with some other popular heuristic algorithms. The results indicated good performance of the proposed method for global minimization of continuous test functions. The method was also used to find the optimal operation of the Dez reservoir in southern Iran, a problem in the reservoir operation discipline. A normalized squared deviation of the releases from the required demands is considered as the fitness function and the results are presented and compared with the solution obtained by Non Linear Programming (NLP) and Discrete Ant Colony Optimization (DACO) models. It is observed that the results obtained from CACO algorithm are superior to those obtained from NLP and DACO models.*

Keywords: *Continuous Ant Colony Optimization (CACO) Algorithm, Global Minimization, Reservoir Operation*

Introduction

Artificial life emerged through the interaction of biology and computer sciences. Artificial life not only contributes to the understanding the secret of life, but also provides innovative concepts and important approaches to practical applications of artificial intelligence, engineering, computer science and so on. Some bio-inspired mathematical models in artificial life such as emergent colonization in the artificial ecology and ant colony system (ACS) have provided the ideas of distributed algorithms which have been successfully applied to the parallel optimization and design such as reservoir operation problems. The underlying idea is that some complex tasks may be performed by distributed activities over massively parallel systems composed of computationally simple elements [1].

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 [1].

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. The shortest path around such an obstacle will be probabilistically chosen just as frequently as a longer path however the pheromone trail will be more quickly reconstituted along the shorter path, as there are more ants moving this way per time unit. Since the ants are more inclined to choose a path with higher pheromone levels, the ants rapidly converge on the stronger pheromone trail, and thus divert more and more ants along the shorter path. This particular behavior of ant colonies has inspired the Ant Colony Optimization 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 [3, 4].

They have been eventually formalized into the framework of the Ant Colony Optimization (ACO) meta-heuristic [4]. 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:

- 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. [5]

Instances of ACO have been applied extensively to a variety of discrete combinatorial optimization problems like the Traveling Salesman Problem (TSP) [6, 7], the Quadratic Assignment Problem (QAP) [8, 9], the Network Communication Routing Problem [10], Vehicle Routing Problem [11], Job-Shop scheduling, Sequential Ordering [12, 13], graph Coloring [12], time tabling, shape optimization, and so on [14, 15]. Since the emergence of ant algorithms as an optimization tool, some attempts were also made to use them for tackling continuous optimization problems, especially in engineering. 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. Jalali et al. [15] proposed Discrete Ant Colony Optimization (DACO) algorithms for reservoir operation. Through a collection of cooperative agents called ants, the near optimum solution to the reservoir operation can be effectively achieved. To apply ACO algorithms, the problem is approached by considering a finite horizon with a time series of inflow, classifying the reservoir volume to several intervals, and deciding for releases at each period with respect to a predefined optimality criterion. Maier et al. [16] developed a formulation which enables ACO algorithms to be used for the optimal design of water distribution systems. This

formulation was applied to two benchmark water distribution system optimization problems and the results are compared with those obtained using genetic algorithms (GAs). The findings indicated that ACO algorithms are an attractive alternative to GAs for the optimal design of water distribution systems, as they outperformed GAs for the two case studies considered both in terms of computational efficiency and their ability to find near global optimal solutions. 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 [17], and also later used by some others [8, 18]. 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 (GA), 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 on 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 delicate setting of parameters [19, 20, 21, and 22].

Other methods include the Asynchronous Parallel Implementation (API) algorithm (API algorithm is 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.) by Monmarche [23], and Continuous Interacting Ant Colony (CIAC), proposed by Dreo and Siarry [19, 24]. 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.

The main purpose of this paper is to construct an effective Continuous Ant Algorithm for continuous optimization problems. For this reason, a way to extend a generic ACO to continuous domains without the need to make any major conceptual changes, based on a pure pheromone has been developed. The proposed algorithm has been used to find the global minimum of some continuous benchmark test functions. Efficiency of the proposed CACO algorithm is compared with some other available popular combinatorial meta-heuristic algorithms reported in the literature. The results indicated good performance of the method for global minimization of continuous test functions. The method was also used to find the optimal operation of the Dez reservoir in southern Iran, a problem in the reservoir operation discipline. A normalized squared deviation of the releases from the required demands is considered as the fitness function and the results are presented and compared with the

solution obtained by Non Linear Programming (NLP) and Discrete Ant Colony Optimization (DACO) models [15]. It is observed that the results obtained from CACO algorithm are superior to those obtained from NLP and DACO models.

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

$$
\text{Minimize } f(x_1, x_2, ..., x_n) \quad \Re^n \to \Re
$$

$$
\text{Subject to : } \quad x_i^L \leq x_i \leq x_i^U \qquad i = 1, 2, ..., n
$$

$$(1)$$

Where f is a multi-variable function, x_i, $i=1,2,...,n$ is the ith variable and x_i^L and x_i^U are the lower and upper bounds for the ith variable, respectively. The first step to develop a continuous ant algorithm optimization is to define a continuous ant decision table. The logical adaptation would be to move from using the discrete probability distribution to a continuous one, the Probability Density Function (PDF). The Gausian PDF used in this paper has the main advantage of easily generating random numbers, but it also has some disadvantages. A single normal PDF is not able to describe a situation where two disjoint areas of the search space are promising, as it only has one maximum. This normal distribution is

defined as follows:

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

$$(2)$$

Where μ is the best solution found within the interval (x_i^L, x_i^U) from the previous iteration and σ is an index of the ant aggregation around the current minimum. To initialize the algorithm, μ is randomly chosen within the bounds, using a uniform PDF and σ is taken equal to length of the search bound, to uniformly locate the ants within it. The construction of solutions in CACO on continuous domain is done in principle in the same manner as in the case of regular ACO. At each iteration, each of the ants chooses a component of the solution using the Gausian probability distribution. In the CACO, pheromone update is a process of modifying the probability distribution used by the ants during 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).

At each iteration of the algorithm, the ants construct the solutions to the problem. Different probabilities generated by a random generator are used when choosing each component of the decision variables of the solutions. Solutions are evaluated, and the iteration best solution is chosen. The new component PDF takes its mean from the value of the respective iteration best solution variable [25]. The standard deviation of the normal PDF used for the update is modified adaptively based on the index of current iteration and the solutions found in this

iteration. To satisfy simultaneously the fitness and aggregation criteria, a concept of weighted standard deviation is defined as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{m} \frac{1}{f_i - f_{min}}[x_i - x_{min}]^2}{\sum_{it=1}^{m} \frac{1}{f_i - f_{min}}}} \qquad (3)$$

Where m is the number of ants, x_i is solution created by the ith ant, f_i is the fitness value of the ith solution, x_{min} is the iteration best solution of the colony and f_{min} is the fitness value of the best solution. This approach means that the center of region discovered during the subsequent iteration is the last best point and the narrowness of its width is dependent on the aggregation of the other competitors around the best one. During 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^L, x_i^U), the height of distribution function increase with respect to the previous iteration and 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 that described previously (positive and negative update).

The algorithm stops if difference between two last best solutions is lower than some predefined tolerance specified by the user, or the maximum number of iterations defined by the user is reached.

Modification of the PDF characteristics (Eq.3) which can be regarded as pheromone update rule leads to a reduction in the standard deviation of the PDF for the next iteration if the iteration best solution gets better from each iteration to the next iteration. In situations where the iteration best solution of the iteration is worst than that of previous iteration, the standard deviation could increase giving more chance to other regions of the search space to be explored. This is a very useful characteristics of the method but could lead to a oscillatory behavior of the method. An elitist strategy is proposed and used in this work in which the best solution of each iteration is directly copied to the next generation. This strategy will guarantee that the best solution of the iteration is always improved with respect to the previous iterations and therefore guarantees the convergence of the method. The proposed mechanism was also found to improve the performance of the algorithms which will be verified later.

Function Optimization

CACO was first used for the optimization of some multi-variable continuous functions. A set of eight benchmark test functions, also used in [26] was employed to evaluate the efficiency of the proposed CACO algorithm. The test functions, the number of variables, the admissible range of the bounds, the known optimum value (global minimum of function) are summarized in Table 1. The results presented here are based on 15 independent runs of the ACO algorithm on each of the test functions. The number of iterations required to find the solution was recorded. The maximum iteration numbers (function evaluations) were fixed to 5000 while the numbers of ants of $m = 10, 25, 50, 100, 200$ were tested. The mean, minimum, and maximum values of the objective function with different number of ants obtained are listed in Table 2. The results of fitness values of benchmark functions indicate that CACO is successful to achieve

Table 1 Optimization Test Functions

Name	Function	Variable Number	Interval	Global Minimum
Sphere	$F_1(x) = \sum_{i=1}^{30} x_i^2$	30	$[-100,100]^n$	0
Schwefel	$F_2(x) = -\sum_{i=1}^{30} (x_i \sin(\sqrt{|x_i|})) + 12569.5$	30	$[-500,500]^n$	0
Rosenbrock	$F_3(x) = \sum_{i=1}^{29} (100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1))^2$	30	$[-30,30]^n$	0
Rastrigin	$F_4(x) = \sum_{i=1}^{30} (x_i^2 - 10\cos(2\pi x_i) + 10)^2$	30	$[-5.12,5.12]^n$	0
Ackley	$F_5(x) = -20 \times \exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32,32]^n$	0
Griewank	$F_6(x) = \frac{1}{4000}\sum_{i=1}^{30} (x_i - 100)^2 - \prod_{i=1}^{30}\cos(\frac{x_i - 100}{\sqrt{i}}) + 1$	30	$[-600,600]^n$	0
Schaffer	$F_7(x) = 0.5 - \frac{\left(\sin\sqrt{x_1^2 + x_2^2}\right)^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	$[-100,100]^2$	0
Freudenstein-Roth	$F_8(x) = \left(-13 + x_1 + ((5 - x_2)x_2 - 2)x_2\right)^2 + \left(-29 + x_1 + ((x_2 + 1)x_2 - 14)x_2\right)^2$	2	$[-100,100]^2$	0

Table 2 Mean, Maximum, and Minimum Test Function Values with different number of ants

Test Function		Number of Ants				
		10	25	50	100	200
F_1	Mean	2.59×10^{-37}	2.63×10^{-126}	3.07×10^{-145}	4.39×10^{-139}	1.49×10^{-139}
	Min	5.81×10^{-43}	2.57×10^{-129}	1.68×10^{-151}	1.53×10^{-147}	5.60×10^{-143}
	Max	2.32×10^{-29}	8.41×10^{-118}	4.78×10^{-143}	2.9×10^{-131}	2.71×10^{-129}
F_2	Mean	4204.78	2997.69	1901.35	2000.34	2154.30
	Min	3954.35	2254.83	997.53	1236.21	1001.35
	Max	5115.87	6478.32	2626.35	2541.23	2800.23
F_3	Mean	53.65	32.32	28.14	25.12	26.29
	Min	1.40	6.28×10^{-2}	4.25×10^{-3}	5.01×10^{-2}	7.35×10^{-3}
	Max	497.02	203.50	93.48	115.23	127.23
F_4	Mean	2.57×10^{-24}	4.27×10^{-30}	4.82×10^{-33}	1.57×10^{-35}	0.00
	Min	1.09×10^{-27}	1.80×10^{-32}	0.00	0.00	0.00
	Max	5.42×10^{-21}	1.79×10^{-27}	5.09×10^{-29}	1.25×10^{-30}	0.00
F_5	Mean	5.46×10^{-10}	1.35×10^{-14}	7.25×10^{-14}	2.78×10^{-14}	6.47×10^{-14}
	Min	4.32×10^{-11}	2.36×10^{-15}	9.89×10^{-14}	9.35×10^{-14}	3.25×10^{-15}
	Max	8.40×10^{-9}	9.78×10^{-13}	1.49×10^{-12}	1.38×10^{-13}	9.78×10^{-13}
F_6	Mean	3.96×10^{-2}	9.85×10^{-3}	3.15×10^{-3}	2.18×10^{-3}	5.47×10^{-3}
	Min	1.99×10^{-8}	2.13×10^{-12}	9.86×10^{-11}	0.00	0.00
	Max	0.10	0.11	1.23×10^{-2}	2.70×10^{-2}	4.51×10^{-2}
F_7	Mean	5.63×10^{-2}	1.85×10^{-3}	2.45×10^{-3}	2.45×10^{-3}	2.45×10^{-3}
	Min	2.45×10^{-3}	2.45×10^{-3}	2.45×10^{-3}	2.45×10^{-3}	2.45×10^{-3}
	Max	0.49	2.14×10^{-2}	2.45×10^{-3}	2.45×10^{-3}	2.45×10^{-3}
F_8	Mean	-	-	-	-	-
	Min	0.00	0.00	0.00	0.00	0.00
	Max	48.98	48.98	48.98	48.98	48.98

International Journal of Civil Engineerng. Vol. 4 , No. 4, December 2006

279

Table 3 Comparison between proposed Continuous Ant Algorithm and other Methods

Function	Mean Best Fitness Values				
	Proposed CACO	PSOPC	GSPSO	LSPSO	CPSO
F_1	$2.63x10^{-126}$	$9.5x10^{-29}$	$1.9x10^{-12}$	$3.0x10^{-3}$	$2.3x10^{-15}$
F_2	2997.69	301.73	1800.68	1640.85	2126.03
F_3	32.32	32.44	52.83	347.92	39.7
F_4	$4.27x10^{-30}$	2.91	21.56	59.07	43.76
F_5	$1.35x10^{-14}$	$2.3x10^{-14}$	$9.00x10^{-8}$	1.41	$1.6x10^{-8}$
F_6	$9.85x10^{-3}$	$3.2x10^{-3}$	$1.4x10^{-2}$	$9.6x10^{-3}$	$1.9x10^{-2}$
F_7	$1.85x10^{-3}$	-	-	-	-
F_8	$0\ or\ 48.98$	-	-	-	-

the optimum solution. Table 3 compares the performance of the proposed CACO algorithm with four other popular meta-heuristic algorithms for given benchmark test functions. The results clearly show that proposed CACO algorithm is a more successful optimizer than other methods on all the test problems used here. However, more detailed analysis on a wider sample of test functions will have to be performed in order to add statistical significance to this claim. Considering the other continuous optimization methods, performance of ACO is similar. Again, larger sets of benchmarks and unified evaluation methods would have to be employed to indicate particular advantages or disadvantages of CACO.

Optimal Reservoir Operation

In reservoir operation problems, to achieve the best possible performance of the system, decisions need to be made on releases and storages over a period of time considering the variations in inflows and demands. In the past, various researchers applied different methods to solve reservoir operation problems. Oliveria and Loucks [27], Marino and Loaiciaga [28], Becker and Yeh [29], solved the optimal operation of multi-

reservoir system with dynamic programming method. East and hall [30], Fahmy et al. [31], Oliveira and Loucks [32], Wardlaw and Sharif [33], used genetic algorithms (GAs) to optimally solve the problem of reservoir operation problems. Peng and Buras [34], Cai et al. [35] used NLP in the optimization of multi-reservoir systems. Jalali [15] has used Ant System (AS) and Ant Colony System (ACS), two basic variants of the Ant Colony Optimization Algorithms (ACOAs), to solve a variety of reservoir operation problems. Meraji et al. [36] used Particle Swarm Optimization algorithm for the simple operation of a reservoir and compared its performance to NLP method.

The fitness function is a measure of the goodness of the generated solutions according to the defined objective function. For this study, Total Squared Deviation of the releases from the required demands is considered as the fitness function defined as:

$$TSD = \sum_{t=1}^{T} \left[R(t) - D(t) \right]^2 \qquad (4)$$

Where $R(t)$ is the release at the end of period t, $D(t)$ is the demand of period t and T is the number of periods for which the operation is to be designed. To determine $R(t)$, the

Table 4 Result of CACO application for Dez Reservoir operation

Dez Reservoir		Number of Ants				
		10	25	50	100	150
TSD	Mean	3.5217	0.9645	0.8211	0.7743	0.7593
	Min	2.9842	0.8712	0.7647	0.7538	0.7418
	Max	5.4328	1.3271	1.0405	0.7953	0.7764

continuity equation along with the following constraints, may be employed as:

$$S(t + 1) = S(t) + R(t) + I(t)$$
$$S_{min} \leq S(t) \leq S_{max} \quad (5)$$
$$R_{min} \leq R(t) \leq R_{max}$$

Where $S(t)$ is the initial storage volume at period t, $S(t+1)$ is the initial storage volumes at period $t+1$ or final storage volume at period , $I(t)$ is the inflow to the reservoir during period t, S_{min} and S_{max} are minimum storage and maximum storage allowed, respectively ,and R_{min} and R_{max} are minimum release and maximum release allowed, respectively.

To illustrate the performance of the model, the Dez reservoir in southern Iran, with an average annual demand of *5900* ^{MCM}, effective storage volume of *2510* ^{MCM}, minimum storage volume of *830* ^{MCM}, and maximum storage volume of *3340* ^{MCM} minimum release volume of *0* ^{MCM}, and maximum release volume of *1000* ^{MCM} is selected. For illustration purposes, a period of 60 months with an average annual inflow of *5303* ^{MCM} is employed. To limit the range of values of the fitness function, a normalized form of equation 4 has been used as:

$$TSD = \sum_{t=1}^{TNP} \left[\frac{R(t) - D(t)}{D_{max}} \right]^2 \quad (6)$$

Where D_{max} is maximum monthly demand. The proposed CACO model was tested for

the Dez reservoir with *S(1) = 1430* ^{MCM} over 20 runs. Results of the model are presented in Table 4 and convergence behavior of CACO is shown in Figure 1 for a typical run using 150 ants. The proposed CACO was able to find the solution of 0.7418 within 2000 iteration with 150 ants. This solution can be compared to the solutions of 0.796 and 0.949 using Non Linear Programming (Lingo 8) and Discrete Ant Colony Optimization (DACO) algorithm, respectively [15]. A note should be made regarding the solution of 0.949 obtained by ACOA reported in [15]. This solution was obtained with a discrete ACO algorithm using 18 points to discretise the allowable range of decision variables. A comparison between demand and release obtained with proposed CACO are also presented in Figure 2.

Conclusions

A new approach based on Continuous Ant Colony Optimization (CACO) algorithm is proposed in this paper. Proposed CACO algorithm uses a new objective-function based heuristic pheromone assignment approach for pheromone update to filtrate solution candidates. Global optimal solutions can be reached more rapidly by self-adjusting the path searching behavior of the ants according to objective values. An elitist strategy is also used in CACO algorithm to improve its performance. The behavior of normal and elitist CACO algorithm is
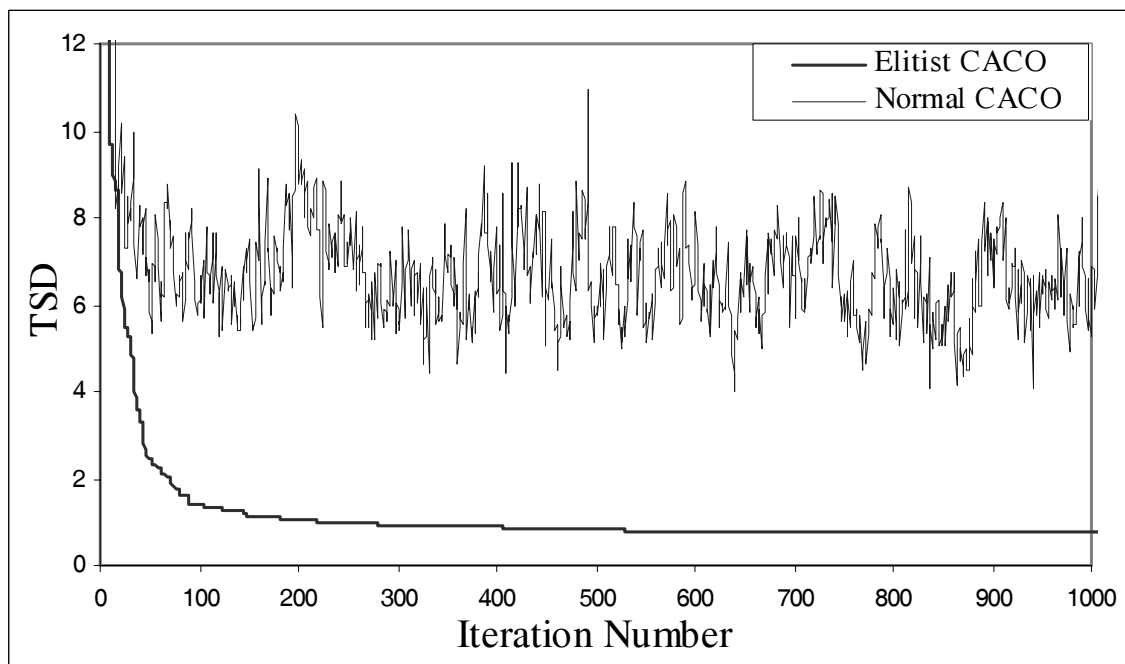
Fig.1 Convergence of Normalized Squared Deviation Function (*TSD*) using 150 Ants.
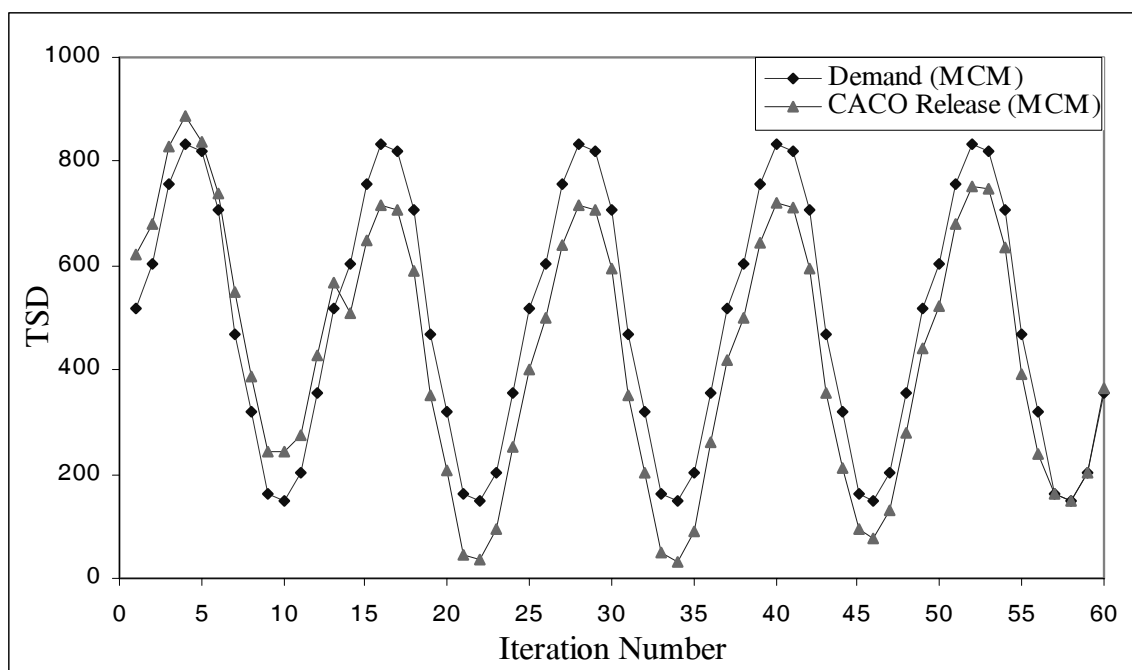

Fig.2 Comparison between demand and releases obtained with CACO algorithm.

investigated and significant role of elitist strategy is exhibited. The performance of proposed algorithm is compared with other meta-heuristic algorithms in solving benchmark test functions. The results indicated the efficiency and reliability of CACO algorithm in solving continuous test functions. Proposed CACO was also applied to find the optimal operation policy of Dez reservoir, in southern Iran. Results of CACO were also compared with the solutions obtained by Non Linear Programming (NLP) and Discrete Ant Colony Optimization (DACO) models. It is observed that the results obtained from CACO algorithm were superior to those obtained from NLP and DACO models. It can be concluded that CACO appears to be a very useful technique for solving global optimization problems in civil engineering, especially in water reservoir engineering.

References

[1] C. Adami, " Introduction to Artificial Life", Published by TELOS, The Electronic Library of Science, Santa Clara, California, 1998, Pages 1-15.

[2] Marco Dorigo, Ottimizzazione, Apprendimento Auto-matico, ed Algoritmi Basati su Metafora Naturale. PhDThesis, Politecnico di Milano, 1992.

[3] M. Dorigo, and G. Di Caro, " The Ant Colony Optimization meta-heuristic" In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw-Hill, New York, NY, USA, 1999.

[4] Krzysztof Socha, " ACO for Continuous and Mixed-Variable Optimization ", IRIDIA, Universite Libre de Bruxelles, CP 194/6, Av.

Franklin D. Roosevelt 50, 1050 Brussels, Belgium

[5] R. Chelouah, and P. Siarry, " Enhanced Continuous Tabu Search" , In S. Voss, S. Martello, I. H. Osman, and C. Roucairol, editors, Meta-Heuristics Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publishers, Boston, MA, USA, 1999, chapter 4, pages 49-61.

[6] R. K. Kincaid, S. Grith, R. Sykes, and J. Sobieszczanski-Sobieski, " A Bell-Curve Genetic Algorithm for Mixed Continuous and Discrete Optimization Problems" , In Proceedings of 43rd AIAA Structures, Structural Dynamics, and Materials Confer-ence. AIAA, Denver, CO, USA, 2002.

[7] M. Mathur, S. B. Karale, S. Priye, V. K. Jyaraman, and B. D. Kulkarni, " Ant Colony Approach to Continuous Function Optimization" , Ind. Eng. Chem. Res., 39, 2000, pages 3814-3822.

[8] P. Siarry, G. Berthiau, F. Durbin, and J. Haussy, " Enhanced Simulated Annealing for Globally Minimizing Functions of Many Continuous Variables", ACM Transactions on Mathematical Software, 1997, 23(2):209-228.

[9] P. A. N. Bosman and D. Thierens, "Continuous Iterated Density Estimation Evolutionary Algorithms within the IDEA Framework" , In M. Pelikan, H. MÄuhlenbein, and A. O. Rodriguez, editors, Proceedings of OBUPM Workshop at GECCO-2000, Morgan-Kaufmann Publishers, San Francisco, CA, USA, 2000,

pages 197-200.

[10] B. Bullnheimer, R. F. Hartl, and C. Strauss, " Applying the Ant System to the Vehicle Routing Problem, Meta-heuristic: Advances and Trends in Local Search Paradigms for Optimization Voss, S., et al(Eds), Kluwer, 1999.

[11] D. Costa, and A. Hertz, " Ants Can Colour Graphs" , Journal of the Operational Research Society, 1997, 48: pages 295-305.

[12] P. Forsyth, and A. Wren, " An Ant System for Bus Driver Scheduling", Proc. Of the 7th Int. Workshop on Computer-Aided Scheuling of Public Transport,1997.

[13] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant Algorithm for Discrete Optimization" , Artificial Life 5 (3), 1999, pages 137-172.

[14] M. Dorigo, E. Bonabeau, and G. Theraulaz, " Ant Algorithm and Stigmergy" , Future Generation Computer Systems 16, 2000, pages 851-871.

[15] Mohammad Reza Jalali, Optimum Design and Operation of Hydrosystems by Ant Colony Optimization Algorithms; A New Meta-Heuristic Approach, PhD Thesis, Iran University of Science and Technology (IUST), Collage of Civil Engineering, 2005.

[16] H. R. Maier, A. R. Simpson, A. C. Zecchin, W. K. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan, "Ant Colony Optimization for Design of Water Distribution Systems" J. Water Resour.

Plng. and Mgmt., Volume 129, Issue 3, May-June 2003, pages 200-209.

[17] G. Bilchev, and I. C. Parmee. " The Ant Colony Metaphor for Searching Continuous Design Spaces" In T. C. Fogarty, editor, Proceedings of the AISB Workshop on Evolutionary Computation, volume 993 of LNCS, Springer-Verlag, Berlin, Germany, 1995, pages 25-39.

[18] M. Wodrich and G. Bilchev, "Cooperative distributed search: the ant's way. Control & Cybernetics" , 1997, (3): pages 413-446.

[19] J. Dreo and P. Siarry, " A New Ant Colony Algorithm Using the Heterarchical Concept Aimed at Optimization of Multiminima Continuous Functions" , In M. Dorigo, G. D. Caro, and M. Sampels, editors, Proceedings of the Third International Workshop on Ant Algorithms (ANTS'2002), volume 2463 of LNCS, Springer-Verlag, Berlin, Germany, 2002, pages 216-221.

[20] J. Dréo, Modélisation de la mobilisation chez les fourmis, Mémoire de DEA, Université Paris7 & Université Libre de Bruxelles, 2001.

[21] C. Ling, S. Jie, O. Ling, and C. Hongjian, " A Method for Solving Optimization Problems in Continuous Space Using Ant Colony Algorithm" , Lecture Notes in Computer Science 2463, pages 288-289, 2002.

[22] L. Y. Jun, W. T. Jun, " An Adaptive Ant Colony System Algorithm for Continuous-space Optimization Problems, Journal of Zhejiang

284

University Science 4 (1), pages 40-46, 2003.

[23] N. Monmarche, G. Venturini, and M. Slimane, " On how Pachycondyla apicalis ants suggest a new search algorithm", Future Generation Computer Systems, 2000, 16:937-946.

[24] J. Dreo, and P. Siarry " Continuous Interacting Ant Colony Algorithm Based on Dense Heterarchy", Future Generation Computer Systems, to appear.

[25] B. Yuan and M. Gallagher. Playing in Continuous Spaces: Some Analysis and Extension of Population-Based Incremental Learning. In Sarker, R. et al., editor, Proceedings of Congress of Evolutionary Computation (CEC), pages 443-450, 2003.

[26] S. He, Q. H. Wu, J. Y. Wen, J. R. Saunders, and R. C. Paton, " A Particle Swarm Optimizer with Passive Congregation" , Biosystems 78 , Available online at w w w . s c i e n c e d i r e c t . c o m , pages 135-147, 2004.

[27] Oliveira R.,Loucks D., Operation rules for multi reservoir systems, water Resource .Res., 1997, 33(4),839-852 .

[28] Marino M.A., Loaiciga H.A., Dynamic model for multi reservoir operation, water Resource Res., 1985, 21(5), 619-630.

[29] Becker L., Yeh W., Optimization of real-time operation of a multiple reservoir system, water Resource .Res, 1974, 10(6), 1107-1112.

[30] East V.,Hall M.J. , water resource system optimization using genetic algorithms , hydro informatics'94 , Pro. , 1st Int. Conf. on Hydro informatics, Balkerma, Rotterdam, The Netherlands, 1994, 225-231.

[31] Fahmy H.S., King J.P.,Wentzle M.W., seton J.A., Economic optimization of river management using genetic algorithms , Int. summer Meeting ,AM. Soc. Agric. engrs., paper no.943034, St. Joseph, Mich. 1994.

[32] Oliveira R.,Loucks D., Operation rules for multi reservoir systems, water Resource .Res., 1997, 33(4),839-852 .

[33] Wardlaw R., Sharif M., Evaluation of genetic algorithms For Optimal Reservoir system operation, Journal of water resources planning and management, January 1999, 25-33.

[34] Peng C.H, Buras N., Dynamic operation of a surface water resources systems, Water Resour. Res., 2000, 36(9),2701-9.

[35] Cai X., Mckinney DC. and Lasdon LS., Piece-by-piece approach to solving large nonlinear water resources management models, ASCE J Water Resour Plann Mgmt 2002, 127(6),363-368.

[36] Meraji. S. Hamed, Afshar. M.H and Afshar. A, Reservoir Operation by Particle Swarm Optimization Algorithm , 7th International conference of civil engineering (Icce7 th), Tehran, Iran, 2006.