Applied Mathematics and Computation 211 (2009) 75–84

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A new solution algorithm for improving performance of ant colony optimization

Ozgur Baskan *, Soner Haldenbilen, Huseyin Ceylan, Halim Ceylan

Department of Civil Engineering, Engineering Faculty, Pamukkale University, Denizli 20017, Turkey

ARTICLE INFO

Keywords:
Ant colony optimization
Reduced search space
Function minimization
Meta-heuristics

ABSTRACT

This study proposes an improved solution algorithm using ant colony optimization (ACO) for finding global optimum for any given test functions. The procedure of the ACO algorithms simulates the decision-making processes of ant colonies as they forage for food and is similar to other artificial intelligent techniques such as Tabu search, Simulated Annealing and Genetic Algorithms. ACO algorithms can be used as a tool for optimizing continuous and discrete mathematical functions. The proposed algorithm is based on each ant searches only around the best solution of the previous iteration with β. The proposed algorithm is called as ACORSES, an abbreviation of *ACO R*educed *SE*arch *S*pace. β is proposed for improving ACO's solution performance to reach global optimum fairly quickly. The ACORSES is tested on fourteen mathematical test functions taken from literature and encouraging results were obtained. The performance of ACORSES is compared with other optimization methods. The results showed that the ACORSES performs better than other optimization algorithms, available in literature in terms of minimum values of objective functions and number of iterations.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Optimization techniques can be classified into two categories: Deterministic and stochastic. Deterministic methods for optimization require the existence of derivatives and continuity. If $f(x)$ is continuously differentiable, the optimum value of given function can be found as a point where $\frac{\partial f(x)}{\partial x} = 0$. However, when the function is not differentiable, it needs stochastic methods. They require random search techniques such as ant colony optimization (ACO) [1], genetic algorithm, etc. The main reasons of the popularity of heuristic methods are their efficiency in solving complicated (discrete) problems and their potentiality of finding global optimum. Many heuristic methods such as modified ant colony optimization (MACO) [2], adaptive random search technique (ARSET) [3], heuristic random optimization (HRO) [4], genetic algorithm by random search technique (IGARSET) [5], dynamic random search technique (DRASET) [6], and successive zooming genetic algorithm (SZGA) [7], have been developed to find global optimum.

Although ACO algorithm is capable of finding global or near global optimum of continuous functions, it may be further improved to locate better global optima for any given test functions. Thus, this study proposes ACO Reduced Search Space (ACORSES) algorithm to improve ACO's performance for finding global optimum of discrete and continuous functions. It differs from other approaches in that its feasible search space (FSS) is reduced with best solution obtained so far using the previous information at the each iteration. At the core of ACORSES, ants search randomly the solution within the FSS to reach global optimum by jumping on each direction. At the end of the each iteration, only one ant is near to global optimum. After

* Corresponding author.
E-mail address: obaskan@pau.edu.tr (O. Baskan).

0096-3003/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2009.01.025

the first iteration, when global optimum is searched around the best solution of the previous iteration using reduced search space, the ACORSES will reach to the global optimum quickly without being trapped in bad local optimum.

This paper is organized as follows. A short brief on ACO is given in Section 2, and definition of the ACORSES algorithm is provided in Section 3. Performance of the ACORSES was experimented on the test functions in Section 4. Last section is about the conclusions.

2. Ant colony optimization (ACO)

ACO has recently been developed, as a population based meta-heuristic [8,19,20], that has been successfully applied to several NP-hard combinatorial optimization problems, such as vehicle routing [9,10], quadratic assignment [11,12], dynamic continuous [13], network design [14,15], vehicle scheduling [16] and travelling salesman problems [17]. Several algorithms are also used based on ACO to find global optimum of functions [1,2].

The ACO is the one of the most recent techniques for approximate optimization methods. The main idea is that it is indirect local communication among the individuals of a population of artificial ants. The core of ant's behavior is the communication between the ants by means of chemical *pheromone* trails, which enables them to find shortest paths between their nest and food sources [21]. The role of pheromone is to guide the other ants towards the target points. This behavior of real ant colonies is exploited.

The general ACO algorithm is illustrated in Fig. 1. The first step consists mainly on the initialization of the pheromone trail. At beginning, each ant builds a complete solution to the problem according to a probabilistic state transition rules. They depend mainly on the state of the pheromone.

Once all ants generate a solution, then global pheromone updating rule is applied in two phases; an evaporation phase, where a fraction of the pheromone evaporates, and a reinforcement phase, where each ant deposits an amount of pheromone which is proportional to the fitness. This process is repeated until stopping criteria is met.

3. Problem formulation

The ACORSES is consisted of three main phases; initialization, pheromone update and solution phase. All of these phases build a complete search to the global optimum as can be seen in Fig. 2. At the beginning of the first iteration, all ants search randomly to the best solution of a given problem within the FSS, and old ant colony is created at initialization phase. After that, quantity of pheromone is updated. In the solution phase, new ant colony is created based on the best solution from the old ant colony using Eqs. (1) and (2). Then, the best solutions of two colonies are compared. At the end of the first iteration, FSS is reduced by $\boldsymbol{\beta}$ and best solution obtained from the previous iteration is kept. $\boldsymbol{\beta}$ guides the bounds of search space during the ACORSES application, where $\boldsymbol{\beta}$ is a vector, $\boldsymbol{\beta}_j = (j = 1,2,\ldots,n)$, and n is the number of variables. The range of the $\boldsymbol{\beta}$ may be chosen between minimum and maximum bounds of any given problem. Optimum solution is then searched in the reduced search space during the algorithm progress. The ACORSES reaches to the global optimum as ants find their routes in the limited space.

Let number of m ants being associated with m random initial vectors (x^k, $k = 1,2,3,\ldots,m$). The solution vector of the each ant is updated using following expression:

$$x_t^{k(new)} = x_t^{k(old)} \pm \alpha \quad (t = 1,2,\ldots,I), \tag{1}$$

where $x_t^{k(new)}$ is the solution vector of the kth ant at iteration $t, x_t^{k(old)}$ is the solution obtained from the previous step at iteration t, and α is a vector generated randomly to determine the length of jump. α controls the global optimum search direction not being trapped at bad local optimum. Ant vector $x_t^{k(new)}$ obtained at tth iteration in (1) is determined using the value of same ant obtained from previous step. Furthermore, in expression (1), (+) sign is used when point x_t^k is on the left of the best solution on the x coordinate axis. $(-)$ sign is used when point x_t^k is on the right of the best solution on the same axis. The direction of search is defined by expression (2).

$$\bar{x}_t^{best} = x_t^{best} + (x_t^{best} * 0.01) \tag{2}$$

If $f(\bar{x}_t^{best}) \leqslant f(x_t^{best})$, (+) sign is used in (1). Otherwise, $(-)$ sign is used. (±) sign defines the search direction to reach to the global optimum. α value is used to define the length of jump, and it will be gradually decreased in order not to pass over global

Step 1:	Initialize Pheromone trail
Step 2:	Iteration Repeat for each ant Solution construction using pheromone trail Update the pheromone trail Until stopping criteria

Fig. 1. A generic ant algorithm.

Initialization

FOR i=1 TO I (I=iteration number)
IF i=1 THEN generate m random ants within FSS
ELSE reduce FSS with range $[x_{t-1}^{best} + \beta; x_{t-1}^{best} - \beta]$
END IF
FOR i=1 TO m
Determine $f(x_t^{best})$ ***(old ant colony)***
Save x_t^{best}
END

Pheromone update

Pheromone evaporation using (3)
Update pheromone trail using (4)

Solution phase

Determine search direction using (2)
Generate the values of α vector
FOR i=1 TO m
Determine the values of new colony using (1) ***(new ant colony)***
Determine new $f(x_t^{best})$
Save x_t^{best}
END
IF $f(x_t^{best})^{new} \leq f(x_t^{best})^{old}$ THEN $x^{global\,\min} = (x_t^{best})^{new}$
ELSE $x^{global\,\min} = (x_t^{best})^{old}$
END IF
$\alpha_t = \alpha_{t-1} * 0.99$
$\beta_t = \beta_{t-1} * 0.99$
END

Fig. 2. Steps of ACORSES.

optimum, as shown in Fig. 2. At the end of the each iteration, a new ant colony (see Fig. 2, second loop) is developed which is different from all literature, as the number of colony size that is generated at the beginning of the each iteration.

Quantity of pheromone (τ_t) is reduced to simulate the evaporation process of real ant colonies using (3) in the pheromone update phase. After reducing of the number of pheromone, it is updated using (4). Quantity of pheromone only intensifies around the best objective function value. This process is repeated until the given number of iteration, I, is completed

$$\tau_t = 0.1 * \tau_{t-1}, \tag{3}$$

$$\tau_t = \tau_{t-1} + 0.01 * f(x_{t-1}^{best}), \tag{4}$$

where initial pheromone intensity is set to 100.

As can be seen in Fig. 2, pheromone update phase is located after the initialization phase, means that quantity of pheromone intensifies at the each iteration within the reduced search space. Thus, global optimum is searched within the reduced search space using best values obtained from new ant colony in the previous iteration. Main advantageous of the ACORSES is that FSS is reduced with $\boldsymbol{\beta}$ and it uses the information taken from previous iteration.

Consider a problem of five ants represents the formulation of the problem. For example as shown in Fig. 3, five ants being associated five random initial vectors. At the beginning of the first epoch (Fig. 3a), old ant colony is randomly created within the feasible search space for any given problem. After pheromone update phase, new ant colony is created at the last phase of the first epoch according to old ant colony using Eqs. (1) and (2). After that, the best values of the two colonies are compared ($E_1 \leftrightarrow E'_1, E_2 \leftrightarrow E'_2, \ldots, E_I \leftrightarrow E'_I$) where I is the iteration number, $\boldsymbol{i} = 1,2,\ldots,I$. According to the best value obtained so far by comparing the old and new colonies and $\boldsymbol{\beta}$, the FSS is reduced at the beginning of the second epoch and once again old ant colony is created, as can be seen in Fig. 3b. The new ant colony is created at the last phase of the second epoch according to randomly generated α value using Eq. (1). Any of the newly created solution vectors may be outside the reduced search space that is created at the beginning of the second epoch. Therefore, created new ant colony prevents being trapped in bad local optimum.

The ACORSES algorithm that achieves the global or near global optimum in many test problems without being trapped in bad local optimum because it is used concurrently to reduce search space and the orientation of all ants to the global

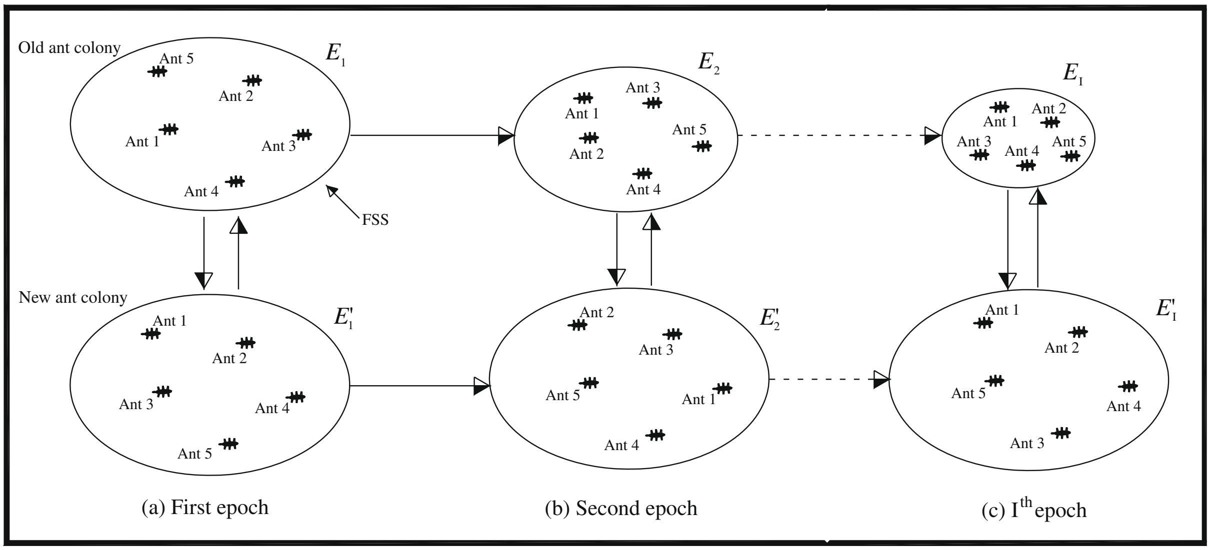

Fig. 3. Main idea of the ACORSES.

optimum using Eqs. (1) and (2). When the results are compared with the test problems, the ACORSES obtains better functional values then the techniques proposed in literature such as ECTS, PSACO, MACO, etc.

4. Application of the ACORSES

The ACORSES is compared with available methods in literature and they are given in Table 1. Fourteen test problems are chosen to test the performance and effectiveness of the ACORSES and they are given in Tables 2a and 2b.

First nine problems are two-dimensional and the rest of the test problems are three and four-dimensional. Their graphs are given in Appendix apart from more than two-dimensional functions. The two-dimensional test problems are taken from

Table 1
The compared algorithms with ACORSES.

Method	Reference	Test problems
ACO	Ant colony optimization [1]	[1]-[2]-[3]-[7]-[9]
MACO	Modified ant colony optimization [2]	All two-dimensional problems except 5
ARSET	Adaptive random search technique [3]	[1]-[2]-[3]-[9]
DRASET	Dynamic random search technique [6]	[8]
SZGA	Successive zooming genetic algorithm [7]	[4]
IGARSET	Improving GAs by random search technique [5]	All two-dimensional problems except 5
ACO	Ant colony optimization [23]	[12]
PSACO	Particle swarm and ant colony algorithm[22]	[12]-[13]-[14]
ECTS	Enhanced continuous Tabu Search [24]	[10]-[11]
ACORSES	Ant colony optimization by reduced search space (this study)	All problems

Table 2a
Two-dimensional test problems.

Pr. no.	Function	Theoretical optimal function value	Graph no.
1	$f(x,y) = (100^*(x-y^2)^2) + (1-x)^2$	$x = 1, y = 1, f(x,y) = 0$	Fig. 4
2	$f(x,y) = \frac{x}{1+\lvert y\rvert}$	$x = -10, y = 0, f(x,y) = -10$ within the range $[-10, 10]$	Fig. 5
3	$f(x) = \left[x * \sin\left(\frac{1}{x}\right)\right]^4 + \left[x * \cos\left(\frac{1}{x}\right)\right]^4$	$x = 0, f(x) = 0$	Fig. 6
4	$f(x,y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7$	$x = 0, y = 0, f(x,y) = 0$	Fig. 7
5	$f(x) = \sum_{i=1}^{n} -x_i * \sin\sqrt{\lvert x_i\rvert}$	$x_i = 420.9687, f(x) = -n^* 418.9829$ within the range $[-500\,500]$	Fig. 8
6	$f(x,y) = x^2 + y^2 - \cos(18x) - \cos(18y)$	$x = 0, y = 0, f(x,y) = -2$	Fig. 9
7	$f(x,y) = \frac{(x-3)^8}{1+(x-3)^8} + \frac{(y-3)^4}{1+(y-3)^4}$	$x = 3, y = 3, f(x,y) = 0$	Fig. 10
8	$f(x,y) = \exp\left(\frac{1}{2}(x^2+y^2-25)^2\right) + \sin^4(4x-3y) + \frac{1}{2}(2x+y-10)^2$	$x = 3, y = 4, f(x,y) = 1$	Fig. 11
9	$f(x) = \left\{\begin{matrix} x^2 & \text{if } x \leqslant 1, \\ (x-3)^2 - 3 & \text{if } x > 1 \end{matrix}\right\}$	$x = 3, f(x) = -3$	Fig. 12

Table 2b
Test problems.[a]

Pr. no.	Function	Theoretical optimal function value and search domain
10	$f(x) = x_1^2 + x_2^2 + x_3^2$	$x = (0.0, 0.0, \ldots, 0.0)$; $f(x) = 0.0$ $-5.12 \leqslant x_i \leqslant 5.12$, $i = 1, 2, \ldots, n$
11	$f(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^4$	$x = (0, \ldots, 0)$; $f(x) = 0$ $-5 \leqslant x_i \leqslant 10$, $i = 1, 2, \ldots, n$
12	$f(x)^{b} = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{n} \alpha_{ij}(x_j - p_{ij})^2\right]$	$x = (0.11, 0.555, 0.855)$; $f(x) = -3.86278$ $0 \leqslant x_j \leqslant 1$, $j = 1, 2, \ldots, n$
13	$f(x) = \sum_{i=1}^{n} x_i^2/4000 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x = (0, \ldots, 0)$; $f(x) = 0$ $-300 \leqslant x_i \leqslant 600$, $i = 1, 2, \ldots, n$
14	$f(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$x = (1, \ldots, 1)$; $f(x) = 0$ $-5 \leqslant x_i \leqslant 10$, $i = 1, 2, \ldots, n$

[a] Test problems are three-dimensional except problems 13 and 14 that they are four-dimensional.
[b] α_{ij} and p_{ij} values can be obtained from [22].

[2] apart from test problem 5 that is taken from [18]. The rest of the test functions are taken from [22]. The starting values of the ants of the ACORSES algorithm for each problem were selected randomly. The results are given in Table 3a and 3b. Obtained best function values, epoch numbers (EN), solution times and the results of compared algorithms are also given in Tables. Based on findings, the ACORSES can find the global optimum with less iteration number on many test problems when it is compared with the other methods except IGARSET that it is better than the ACORSES in many test problems in terms of iteration number. While the colony size is given same for all test problems, initial value of α is randomly generated, as can be seen in Tables 3a and 3b.

Table 3a
The results of two-dimensional functions.

Pr. no.	The parameters of ACORSES	Method	Best function value	EN	Best solution time (s)
1	$m = 20$, $\boldsymbol{\beta} = [2,2]$, $\alpha = 1/\text{random}(10)$	ACO	0	5000	NA[a]
		MACO	0	3600	0.0620
		ARSET	4.02E−16	50000	NA
		IGARSET	0	2174	0.0568
		ACORSES	0	3402	0.0590
2	$m = 20$, $\boldsymbol{\beta} = [5,5]$, $\alpha = 1/\text{random}(10)$	ACO	−10	5000	NA
		MACO	−10	3750	0.0440
		ARSET	−10	30000	NA
		IGARSET	−10	1205	0.1043
		ACORSES	−10	2167	0.0620
3	$m = 20$, $\boldsymbol{\beta} = [2,2]$, $\alpha = 1/\text{random}(10)$	ACO	1.40E−45	5000	NA
		MACO	5.60E−45	5000	0.032
		ARSET	2.21E−43	50000	NA
		IGARSET	1.01E−74	1789	0.0669
		ACORSES	4.58E−83	4101	0.0550
4	$m = 20$, $\boldsymbol{\beta} = [4,4]$, $\alpha = 1/\text{random}(10)$	MACO	0	3750	0.0080
		SZGA	2.98E−8	4000	NA
		IGARSET	0	1004	0.0485
		ACORSES	0	1832	0.0520
5	$m = 20$, $\boldsymbol{\beta} = [250,250]$, $\alpha = 1/\text{random}(10)$, $n = 2$	ACORSES	−837.9658	1176	0.0690
6	$m = 20$, $\boldsymbol{\beta} = [2,2]$, $\alpha = 1/\text{random}(10)$	MACO	−2	235	0.0120
		IGARSET	−2	2400	0.0614
		ACORSES	−2	1610	0.0445
7	$m = 20$, $\boldsymbol{\beta} = [3,3]$, $\alpha = 1/\text{random}(10)$	ACO	0	5000	NA
		MACO	0	3750	0.0180
		IGARSET	2.08E−27	1821	0.0666
		ACORSES	4.06E−52	3624	0.0830
8	$m = 20$, $\boldsymbol{\beta} = [5,5]$, $\alpha = 1/\text{random}(10)$	MACO	1	36000	0.0210
		DRASET	1	29663	14.468
		IGARSET	1	1849	0.0537
		ACORSES	1	1576	0.0630
9	$m = 20$, $\boldsymbol{\beta} = [3,3]$, $\alpha = 1/\text{random}(10)$	ACO	−3	500	NA
		MACO	−3	500	0.0090
		ARSET	−3	1000	NA
		IGARSET	−3	465	0.0420
		ACORSES	−3	500	0.0480

[a] NA: not available.

Table 3b
The results of three and four-dimensional functions.

Pr. no.	The parameters of ACORSES	Method	Best function value	EN	Best solution time (s)
10	$m = 20$, $\beta = [10, 10, 10]$, $\alpha = 1/\text{random}(10)$	ECTS	3E−08[b]	338[b]	NA[g]
		ACORSES	0	4850	0.0780
11	$m = 20$, $\beta = [15, 15, 15]$, $\alpha = 1/\text{random}(10)$	ECTS	4E−6[c]	2254[c]	NA
		ACORSES	0	5530	0.0620
12	$m = 20$, $\beta = [1, 1, 1]$, $\alpha = 1/\text{random}(10)$	PSACO	−3.8627	592[a]	NA
		ACO	−3.86	528[a]	0.7400[f]
		ACORSES	−3.8626	428	0.0560
13	$m = 20$, $\beta = [800, 800, 800, 800]$, $\alpha = 1/\text{random}(10)$	PSACO	6.23E−22[d]	1081[d]	NA
		ACORSES	3.49E−35	4062	0.0520
14	$m = 20$, $\beta = [15,15,15,15]$, $\alpha = 1/\text{random}(10)$	PSACO	1.85e−04[e]	517[e]	NA
		ACORSES	1.78e−04	948	0.0635

[a] This values are given as the average number of function evaluations in [22,23].
[b] This values are given as average error of best objective function values and the average number of function evaluations, respectively, in [24].
[c] This values are given as average error of best objective function values and the average number of function evaluations, respectively in [24] for 5 variables.
[d] This values are given as average error of best objective function values and the average number of function evaluations, respectively, in [22] for 8 variables.
[e] This values are given as average error of best objective function values and the average number of function evaluations, respectively, in [22] for 5 variables.
[f] The standard unit of time was used in [23].
[g] NA: not available.

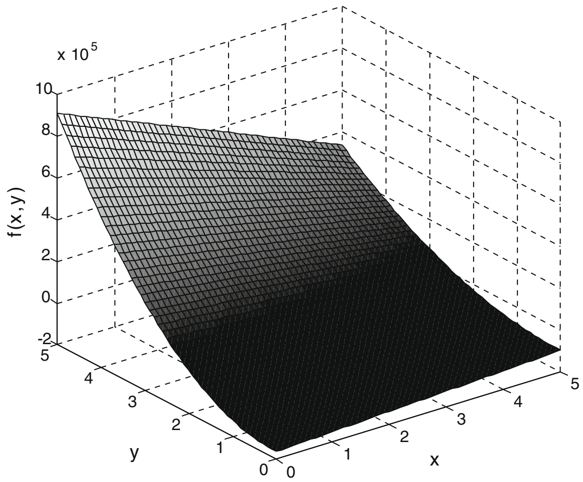

Fig. 4. The graph of test problem 1.

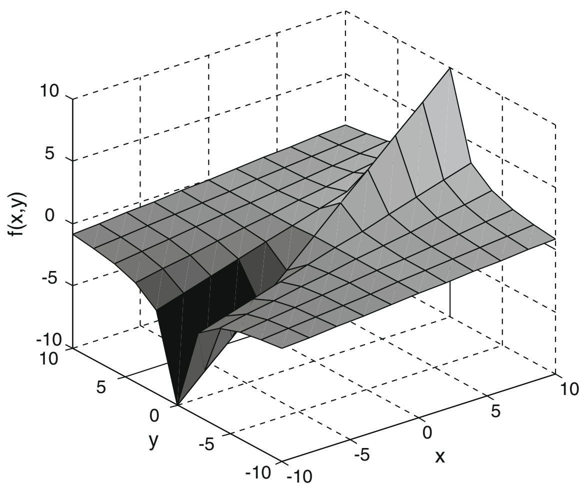

Fig. 5. The graph of test problem 2.

The ACORSES is reached considerably successful results for finding the optimum values of given functions as can be seen in Tables 3a and 3b.

IGARSET gives better results than the ACORSES except for test problems 3, 6 and 8. In those problems, the ACORSES is better than IGARSET. However, the performance of the ACORSES is better than the others (except IGARSET) for all problems apart from test problems 6 and 7. In test problems 6 and 7, MACO gives better result than the ACORSES. The test problems from 10 to 14 are more than two-dimensional. Nonetheless, the ACORSES reached the exact solutions for given test functions in reasonable number of evaluation and solution time when it is compared with other algorithms. The ACORSES is better

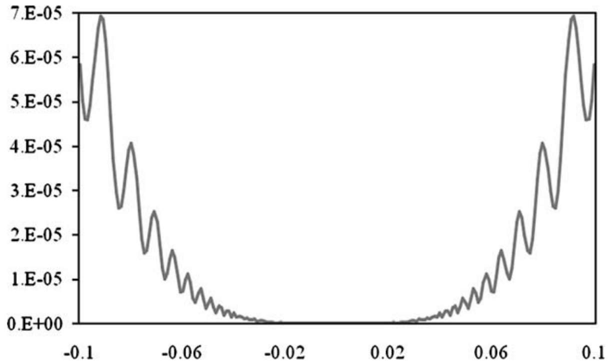

Fig. 6. The graph of test problem 3.

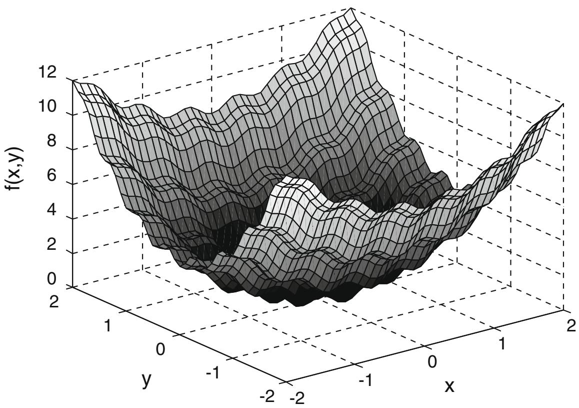

Fig. 7. The graph of test problem 4.

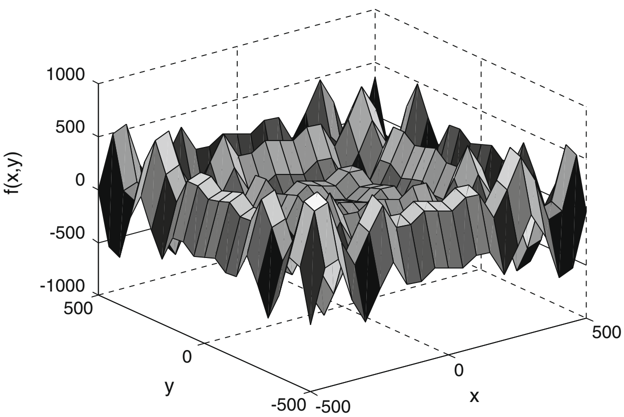

Fig. 8. The graph of test problem 5.

than ACO and PSACO for test problem 12 in terms of number of evaluation. It is also better than PSACO for problem 13 when the best objective function value is compared. In test problems 10 and 11, the ACORSES reached the exact solution in a reasonable solution time. Thus, the ACORSES has ability to reach the exact solution without being trapped in bad local optimum for given test functions. The investigation of the effects of colony size for finding optimal values of functions is the beyond the scope of this study. Therefore, it is not analyzed in this study. Solution times of test problems for compared algorithms are also given in Tables 3a and 3b as last column. According to solution times, the ACORSES algorithm is capable of finding the global optimum in reasonable solution times for given both test functions.

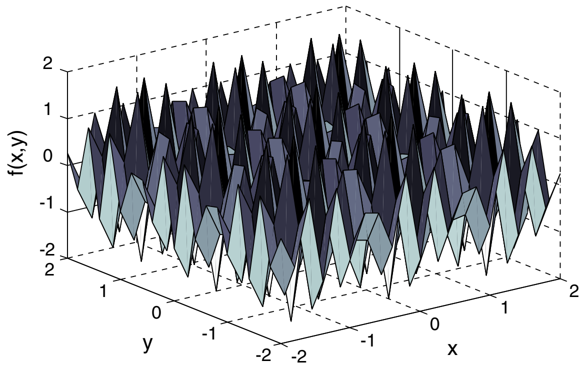

Fig. 9. The graph of test problem 6.

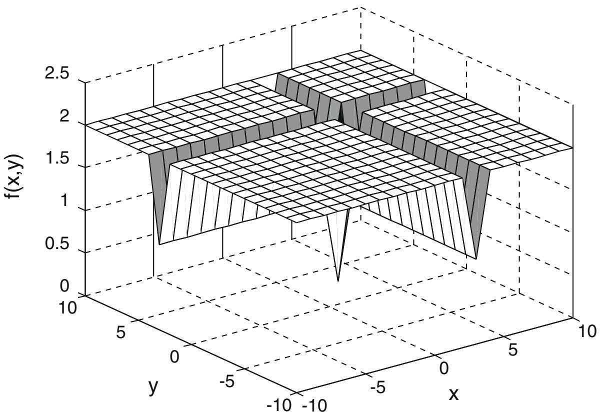

Fig. 10. The graph of test problem 7.

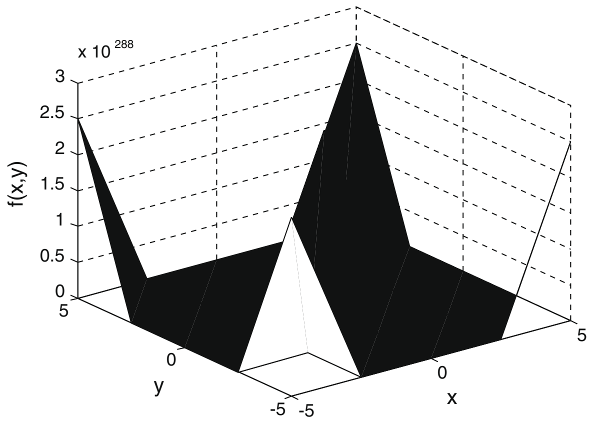

Fig. 11. The graph of test problem 8.

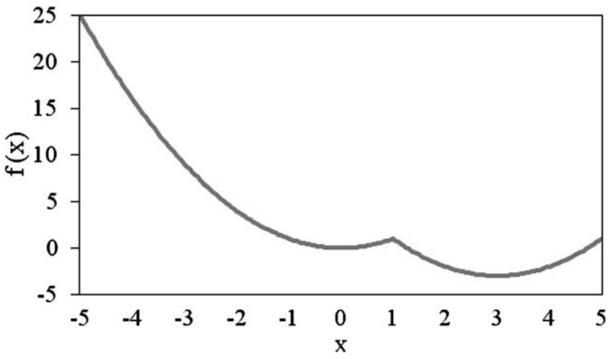

Fig. 12. The graph of test problem 9.

5. Conclusions

A heuristic ACORSES algorithm for finding global optimum has been proposed. It is performed by means of reduced search space with $\boldsymbol{\beta}$ and using the information provided by previous solution. Moreover, ACORSES differs in terms of the generated new colony. Means that at the end of the each iteration, a new colony is developed as a number of colony size, that is generated at the beginning of each iteration and the length of jump is applied to the same ant from the previous step at the iteration t to generate a new colony instead of applying the best ant, which is obtained from the previous iteration.

The ACORSES is tested on 14 different test problems which are two and more dimensional and reached considerably successful results. The ACORSES is compared with different algorithms available in literature such as ACO, MACO, ARSET, DRASET, SZGA, IGARSET, PSACO and ECTS. It is concluded that reduced search space with $\boldsymbol{\beta}$ and generated new colony during the steps of the ACORSES may help to find global optimum without being trapped in bad local optimum in selected test problems within a reasonable solution time.

Acknowledgements

The authors would like to thank the anonymous referees for their constructive and useful comments during the development stage of this paper. Scientific Research Foundation of the Pamukkale University with the Project No. 2007-FBE-002 is also acknowledged.

Appendix. The graphs of two-dimensional test problems are given in Figs. 4–12.

References

[1] M.D. Toksarı, Ant colony optimization for finding the global minimum, Applied Mathematics and Computation 176 (2006) 308–316.
[2] M.D. Toksarı, A heuristic approach to find the global optimum of function, Journal of Computational and Applied Mathematics 209 (2) (2007) 160–166.
[3] C. Hamzacebi, F. Kutay, A heuristic approach for finding the global minimum: adaptive random search technique, Applied Mathematics and Computation 173 (2) (2006) 1323–1333.
[4] J. Li, R.R. Rhinehart, Heuristic random optimization, Computers and Chemical Engineering 22 (3) (1998) 427–444.
[5] C. Hamzacebi, Improving genetic algorithms' performance by local search for continuous function optimization, Applied Mathematics and Computation 196 (2008) 309–317.
[6] C. Hamzacebi, Continuous functions minimization by dynamic random search technique, Applied Mathematical Modelling 31 (2007) 2189–2198.
[7] Y.D. Kwon, S.B. Kwon, J. Kim, Convergence enhanced genetic algorithm with successive zooming method for solving continuous optimization problems, Computers and structures 81 (2003) 1715–1725.
[8] M. Dorigo, G. Di Caro, Ant colony optimisation: a new meta-heuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation, vol. 2, 1999, pp. 1470–1477.
[9] A.V. Donati, R. Montemanni, N. Casagrande, A.E. Rizzoli, L.M. Gambardella, Time dependent vehicle routing problem with a multi ant colony system, European Journal of Operational Research 185 (3) (2008) 1174–1191.
[10] J.E. Bell, P.R. McMullen, Ant colony optimization techniques for the vehicle routing problem, Advanced Engineering Informatics 18 (2004) 41–48.
[11] E.G. Talbi, O. Roux, C. Fonlupt, D. Robillard, Parallel ant colonies for the quadratic assignment problem, Future Generation Computer Systems 17 (2001) 441–449.
[12] N.C. Demirel, M.D. Toksari, Optimization of the quadratic assignment problem using an ant colony algorithm, Applied Mathematics and Computation 183 (2006) 427–435.
[13] J. Dreo, P. Siarry, An ant colony algorithm aimed at dynamic continuous optimization, Applied Mathematics and Computation 181 (2006) 457–467.
[14] S.N. Kuan, H.L. Ong, K.M. Ng, Solving the feeder bus network design problem by genetic algorithms ad ant colony optimization, Advances in Engineering Software 37 (2006) 351–359.
[15] H. Poorzahedy, F. Abulghasemi, Application of Ant system to network design problem, Transportation 32 (2005) 251–273.
[16] H. Wang, J. Shen, Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints, Applied Mathematics and Computation 190 (2007) 1237–1249.
[17] C.B. Cheng, C.P. Mao, A modified ant colony system for solving the travelling salesman problem with time windows, Mathematical and Computer Modelling 46 (2007) 1225–1235.

[18] H.P. Schwefel, Numerical Optimization of Computer Models, Wiley & Sons, Chichester, 1981.
[19] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, Massachusetts Institute of Technology, Cambridge, 2004.
[20] M. Dorigo, Optimization, learning and natural algorithms, Ph.D. Thesis, Politecnico di Milano, Italy, 1992.
[21] J.L. Denebourg, J.M. Pasteels, J.C. Verhaeghe, Probabilistic behaviour in ants: a strategy of errors?, Journal of Theoretical Biology 10 (1983) 259–271
[22] P.S. Shelokar, P. Siarry, V.K. Jayaraman, B.D. Kulkarni, Particle swarm and ant colony algorithms hybridized for improved continuous optimization, Applied Mathematics and Computation 188 (2007) 129–142.
[23] M.D. Toksari, Minimizing the multimodal functions with ant colony optimization approach, Expert Systems with Applications 36 (2009) 6030–6035.
[24] R. Chelouah, P. Siarry, Tabu Search applied to global optimization, European Journal of Operational Research 123 (2000) 256–270.