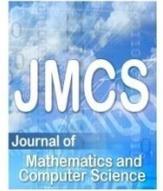




Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: [www.tjmcs.com](http://www.tjmcs.com)



## A Hybrid Model of Particle Swarm Optimization and Continuous Ant Colony Optimization for Multimodal Functions Optimization

Moein Fazeli Hassan Abadi<sup>1,\*</sup>, Hassan Rezaei<sup>1,+</sup>

<sup>1</sup> Department of Computer Science, Faculty of Mathematics, University of Sistan and Baluchestan, Iran.

\* [moein\\_fazeli@pgs.usb.ac.ir](mailto:moein_fazeli@pgs.usb.ac.ir)

+ [hrezaei@cs.usb.ac.ir](mailto:hrezaei@cs.usb.ac.ir)

### Article history:

Received November 2014

Accepted February 2015

Available online February 2015

### Abstract

In many real-world optimization problems, multimodal function optimization is considered, in fact, we are faced with a multimodal optimization problem. Particle swarm optimization (PSO) and continuous ant colony optimization (ACOR) are two population-based optimization techniques that work based on probability laws. The main problem of PSO and ACOR algorithms is premature convergence and falling into local optima. One way to solve the problems is to use combinational methods. This paper presents a combinational method including PSO and ACOR in order to improve the search process. The proposed algorithm tries to solve the problem. Standard benchmark functions are used in order to evaluate the proposed algorithm, proposed method was compared with ACOR, PSO and sequential approach with the enlarged pheromone-particle table of the composition of PSO and ACOR Introduced in [8]. Results show that the proposed method is superior.

**Keywords:** Multimodal function optimization, Continuous Ant Colony Optimization, Particle Swarm Optimization

## 1. Introduction

In multimodal optimization, user has obtained more knowledge about optimal solutions in search space and it helps to use other solutions when the current solution is not possible due to some issues (such as some physical constraints). Obtaining the best possible outcome of a question is called optimization according to governing conditions. In optimization issues of real world, sometimes only one optimal solution is not sufficient. When there are multiple optimal solutions for a problem, demand for various solutions will be more sensitive. The issues that have more optimal points, if all these points contribute to problem solution, we will face with a multifaceted problem and each of these points is called a local optimum and the greatest of points (in maximization issues) is called global

optimum and local optimum may be as good as global optimum in better decision making. Innovative solutions are good ways for obtaining optimal solutions, but do not guarantee optimal solutions. Nowadays, embracing innovative ways is drastically increased with larger and complex issues [1].

PSO algorithm is a heuristic optimization technique that works on the basis of population. The main idea of this algorithm was first proposed in 1995 by Kennedy and Eberhart [2] that was inspired from bunch of fish and birds' behaviors to find food. A group of particles search for food in a random space and there is only one food source and none of the particles are aware of food place and they only know the distance to the food. One of the best strategies is to move towards the particle which is closer to food source. This strategy is the main idea behind the PSO algorithm.

ACOR algorithm that is a generalization of the discrete ant colony (ACO) was proposed for solving optimization problems with continuous values for the first time in 2008 by Soosha and Dorigo [3], good solutions are kept in this solution archives. Each solution has a value of objective the function. New solutions are produced for the next generation using a normal distribution that is obtained on the basis of objective function values from any solution of pheromone table (Solution archive).

Combining algorithms can improve the original algorithm and obtain a solution with superior quality. In evolutionary calculations, combination is essential in order to improve an optimization algorithm performance. Generally, an algorithm performance is lower than combinational algorithm [4,5]. In this paper, a combinational strategy of combining two PSO and ACOR algorithms is proposed that solves greatly problems of ACOR and PSO algorithms.

Classification of remaining paper is as below: firstly, section 2 introduces basic concepts, including particle swarm optimization and continuous ant colony optimization algorithms. Section 3 deals with the concept of multimodal optimization. In Section 4, the proposed method of this paper is described. In Section 5, time and computational complexity of proposed method is discussed. In Section 6, the proposed algorithm has been tested on several benchmark functions. Finally, the conclusion is presented in Section 8.

## 2. Preliminaries

### 2.1. Particle Swarm Optimization

It was developed by Kennedy and Eberhart, in 1995 and has been successfully employed in many scientific and applied areas. This is a population-based algorithm in which anyone is considered as a particle and any population consists of a number of these particles. In PSO, problem-solving space is regarded as a searching environment and anyplace is a problem-dependent solution. In this population, particles tries to find the best situation (best solution) in the searching place (solution space). All particles move according to their speed. The movement of particles in any iteration is calculated by the following formula:

$$v_i^d(t) = w \times v_i^d(t) + c_1 \times \text{rand1} \times (\text{pbest}_i^d(t) - x_i^d(t-1)) + c_2 \times \text{rand2} \times (\text{gbest}^d(t) - x_i^d(t-1)) \quad (1)$$

$$x_i^d(t) = x_i^d(t-1) + v_i^d(t) \quad (2)$$

In relations (1) and (2),  $x_i^d$  is the current position of  $d^{\text{th}}$  dimension of  $i^{\text{th}}$  particle and  $v_i^d$  is the current speed of this dimension of this particle, and  $\text{pbest}_i^d$  is the previous optimal position of the  $d^{\text{th}}$  dimension encountered by  $i^{\text{th}}$  particle,  $\text{gbest}^d$  is the best position of  $d^{\text{th}}$  dimension which has been ever found by the population,  $w$  is the weight of inertia that gives a proportion of the previous speed,  $c_1$  and

$c_2$  are the acceleration coefficients and define the best effect of the position of each particle and the best overall position,  $rand_1$  and  $rand_2$  are two random numbers between 0 and 1. Fig. 1 presents the procedure of particle swarm optimization [6, 7].

```

Initialize a population of particles with random positions and velocities in
the search space.
While (termination conditions are not met)
{
  For each particle i do
  {
    For each dimension of particle I do
    {
      Update the velocity of dth dimension of particle i according to equation (1).
      Update the position of dth dimension of particle i according to equation (2).
      Update pbestd (t) and gbestd (t) if necessary.
    }
    Map the position of particle i in the solution space and evaluate its fitness
    value according to the fitness function.
  }
}

```

Fig. 1. Procedure of particle swarm optimization

## 2.2. Continuous Ant Colony Optimization

Ant colony algorithm is applied mostly for permutation and discrete problems. How we can use Ant Colony Algorithm for continuous domains. Different approaches were proposed for continuous ant colony algorithm before 2008 but were not really accepted. In 2008, Dorigo and Sotoca raised a very interesting idea for the continuous ant colony algorithm that is a generalization of the discrete ant colony algorithm. The number of selections is countable in discrete spaces, but it is not so in continuous space, the number of selections is infinite, so we must consider the points that are close to better solutions in continuous space. In fact, the continuous space and a set of characterized points of search space are emphasized in continuous ant colony, a distribution of pheromone is defined and pheromone distribution can be explained as probability distribution, in this case, the normal distribution is used. It is the basis for normal distribution and its centers are points that we have tried before and are confident of their relative well-being and we use solution archive (pheromone table) in order to maintain these points. Figure 2 shows the structure of solution archive, in this figure,  $S_1^i$  represents the  $i^{\text{th}}$  variable of  $1^{\text{th}}$  solution in solution archive [3].

In ACOR, we assign a probability to each solution based on fitness, then, we obtain a distribution for each decision making variable in solution archives and sample from distribution and create a new solution. Pheromones of this method are probabilities that we assign to each solution and continuity of problem is where we don't consider problem itself, but we focus all around it [3].

$s_1$	$s_1^1$	$s_1^2$	...	$s_1^i$	...	$s_1^n$	$f(s_1)$
$s_2$	$s_2^1$	$s_2^2$	...	$s_2^i$	...	$s_2^n$	$f(s_2)$
	•	•	•	•	•	•	•
	•	•	•	•	•	•	•
	•	•	•	•	•	•	•
$s_i$	$s_i^1$	$s_i^2$	...	$s_i^i$	...	$s_i^n$	$f(s_i)$
	•	•	•	•	•	•	•
	•	•	•	•	•	•	•
	•	•	•	•	•	•	•
$s_k$	$s_k^1$	$s_k^2$	...	$s_k^i$	...	$s_k^n$	$f(s_k)$

Fig. 2. The structure of the solution archive

The algorithm is detailed as follows [8]:

- 1) The target function  $f(s_i)$  is calculated for any solution  $s_i$  in the pheromone table. Available solutions in the table are arranged based on the target function. Thus, for a minimization problem, we have:

$$f(s_1) \leq f(s_2) \leq \dots \leq f(s_i) \leq \dots \leq f(s_k)$$

- 2) Weight ( $w$ ) is calculated for  $i^{\text{th}}$  solution in the pheromone table as below:

$$w_i = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(i-1)^2}{2q^2k^2}} \tag{3}$$

In the above equation  $q$  shows the learning rate, which is a value between 0 and 1.

- 3) As to  $w_i$  for each available solution in the pheromone table, Roulette Wheel probability  $p_i$  is estimated as bellow:

$$p_i = \frac{w_i}{\sum_{j=1}^k w_j} \tag{4}$$

- 4) Stage 4 is iterated  $M$  times to produce  $M$  new ants ( $M \leq K$ ): using normal distribution  $(\mu_i^d, \sigma_i^d)$ , a new ant is achieved for each variable.  $\mu_i^d$  is a value selected from  $d^{\text{th}}$  variable and from  $i^{\text{th}}$  solution in the pheromone table with the probability  $p_i$ .  $\sigma_i^d$  is defined as follows:

$$\sigma_i^d = \tau \sum_{j=1}^k \frac{|x_j^d - x_i^d|}{k-1} \tag{5}$$

In the above equation,  $x_i^d$  is the  $d^{\text{th}}$  variable from the  $i^{\text{th}}$  solution.  $K$  is the size of the pheromone table and  $\tau$  shows the rate of evaporation which is between 0 and 1.

- 5)  $M$  new ants are evaluated and less qualified solutions are replaced in the pheromone table by superior solutions and by  $M$  new ants.

### 3. Multimodal optimization

A function that has more than one optimum is called multimodal function. However, a multimodal function may have one or more than one global optimum. When we plan to optimize a multimodal function, different problems will be arisen. The objective of optimization algorithms is to find a global optimum that are usually applied to functions with one global optimum and some local optimums. Hence, the main problem is to avoid algorithm falling into the trap of local optimums. On the other hand, when the function has several optimums with the same values, the aim of an optimization algorithm may be to find all optimums [9].

Many optimization techniques are designed to deal with the multiple functions of the first type. They usually assume that there is only one solution in search space and use their efforts to isolate it from other false solutions.

The situation is different when dealing with multimodal global optimum functions. In this case, standard techniques usually prefer a solution or may become confused by multiple solutions and fail to converge to one of optimums. Hence, using combinational standard algorithms is essential that can act effectively on multimodal functions.

### 4. Proposed method

In the proposed method, the ACOR and PSO share the same set of solutions, named the “pheromone-particle” table, abbreviated as "PHERO\_PAR". This method consists of two phases, which two phases are executed in parallel. The proposed method is a combination of sequential and parallel Strategies of combining PSO and ACOR introduced in [8]. Figure 2 shows the main concept of the proposed method, and its detailed algorithm is as follows:

#### Algorithm:

- step 1. Create and initialize a pheromone-particle table, “PHERO-PAR”, with K rows and D dimensions using the uniform distribution.
- step 2. Phase 1:
  - 2.1. Calculate fitness for each particle in PHERO-PAR using test function, update pbest for each particle, and update gbest with the best fitness value of all the particles.
  - 2.2. Generate new positions for all particles in PHERO-PAR by performing PSO algorithm and update pbest for each particle, and update gbest.
  - 2.3. Generate new ants, Ant, with M rows by performing ACOR algorithm based on the updated PHERO-PAR performed in the previous step.
  - 2.4. Pheromone particles Table updated by PSO combines with M new ant, and select the best of them to shape “PHERO-PAR1” with K solution.
- step 3. Phase 2:
  - 3.1. Calculate fitness for each particle in PHERO-PAR using test function, update pbest for each particle, and update gbest with the best fitness value of all the particles.

- 3.2. Generate new ants, Ant, with M rows by performing ACOR algorithm based on the updated PHERO-PAR performed in the previous step.
  - 3.3. Generate new positions for all particles in PHERO-PAR by performing PSO algorithm and update pbest for each particle, and update gbest.
  - 3.4. Phomone particles Table updated by PSO combines with M new ant, and select the best of them to shape “PHERO-PAR2” with K solution.
  - 3.5. PHERO-PAR1 combines with PHERO-PAR2, and select the best of them to shape new “PHERO-PAR” with K solution, and update pbest for each particle, and update gbest.
- step 4. If the stop condition is satisfied, the algorithm terminates and is returned gbest, otherwise, go to the second step of the algorithm.

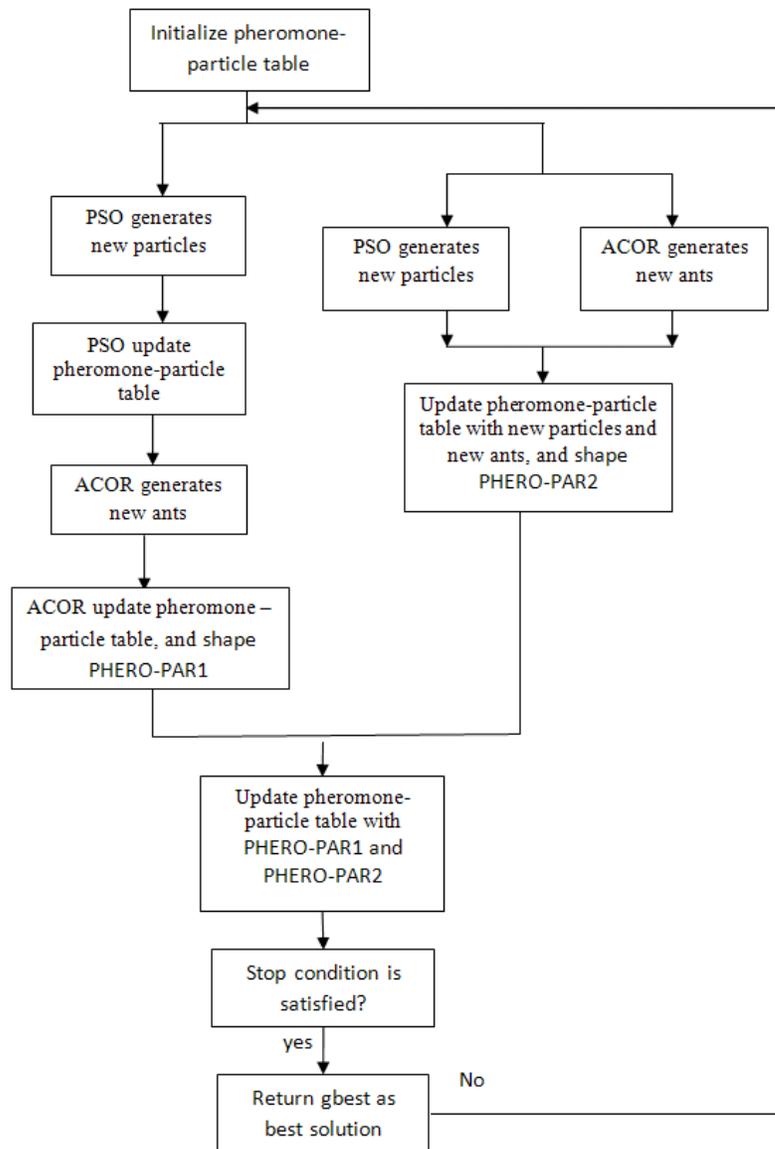


Fig. 3. main concept of the proposed method

### 5. Computational complexity analysis

The time complexity of the Proposed method are proportional to the number of iterations, the number of particles, and the number of ants, and can be computed according to their main steps, as follows:

$$Time\ complexity = N_{iteration \times ns} \times (N_{NewAnts} \times T_{Ant} + N_{NewParticle} \times T_{Particle} + T_{Update}) \tag{6}$$

Where:

$N_{NewAnts}$ : Number of iterations

$N_{NewParticle}$ : Number of new ants

$N_{NewParticle}$ : Number of new particles

$T_{Ant}$ : cpu time for generating a new ant

$T_{Particle}$ : cpu time for generating a new particle

$T_{Update}$ : cpu time for relevant processes including updating the pheromone table, updating the position of a new particle, and other necessary processing times

### 6. Experiments

In this section, proposed algorithm is assessed with ACOR, PSO and sequential approach with the enlarged pheromone-particle table [8] algorithms by 10 test functions; the test functions are introduced in Table 1. In this table, “Range” presents function domain and “Dim” specifies Function dimension. The proposed implementation was executed on the Matlab9 platform, with an Intel Core 2 Duo CPU running at 3.0GHz and 4GB RAM; Parameters needed to be set for running algorithms are given in Table 2. Selecting parameters has been experimentally. Below, the results of running algorithms are given. To summarize the sequence with enlarged pheromone-particle table model was introduced as hybrid I and proposed model as hybrid II.

Table 1. Test functions adopted for our experiments.

Test function	Number Of Global Minimum	Dim	Range
$f_1(x) = \begin{cases} \frac{160}{10}x & \text{for } 0 \leq x < 10 \\ \frac{160}{5}(15-x) & \text{for } 10 \leq x < 15 \\ \frac{200}{5}(x-15) & \text{for } 15 \leq x \leq 20 \end{cases}$	2	1	$0 \leq x < 20$
$f_2(x) = \begin{cases} \frac{160}{10}x & \text{for } 0 \leq x < 15 \\ \frac{160}{5}(x-15) & \text{for } 15 \leq x < 20 \\ \frac{200}{5}(x-20) & \text{for } 20 \leq x \leq 25 \end{cases}$	3	1	$0 \leq x \leq 25$

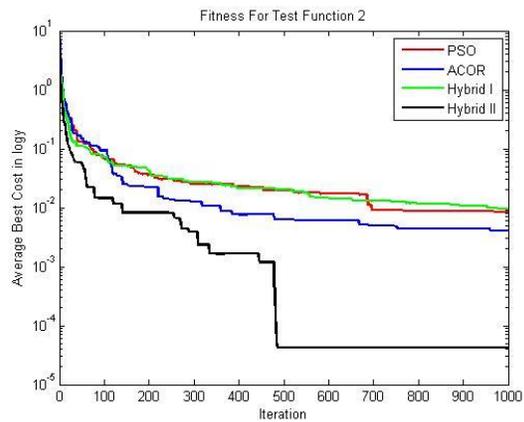
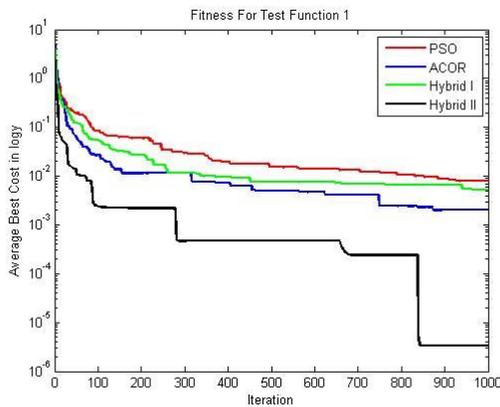
$f_3(x) = \begin{cases} 80(2.5-x) & \text{for } 0 \leq x < 2.5 \\ 64(x-2.5) & \text{for } 2.5 \leq x < 5 \\ 64(7.5-x) & \text{for } 5 \leq x \leq 7.5 \\ 28(x-7.5) & \text{for } 7.5 \leq x \leq 12.5 \\ 28(17.5-x) & \text{for } 12.5 \leq x \leq 17.5 \\ 32(x-17.5) & \text{for } 17.5 \leq x \leq 22.5 \\ 32(27.5-x) & \text{for } 22.5 \leq x \leq 27.5 \\ 80(x-30) & \text{for } 27.5 \leq x \leq 30 \end{cases}$	7	1	$0 \leq x < 30$
$f_3(x) = \sin^6(5\pi x)$	5	1	$-10 \leq x < 1$
$f_5(x) = \begin{cases} 15(15-x) & \text{for } 0 \leq x < 15 \\ \left(\frac{15}{4}\right)(x+10) & \text{for } 15 \leq x < 20 \\ 15(x-20) & \text{for } 20 \leq x < 30 \end{cases}$	2	1	$0 \leq x \leq 30$
$f_6(x) = \sin^6\left(5\pi\left(x^{\frac{3}{4}} - 0.05\right)\right)$	5	1	$0 \leq x < 1$
$f_7(x) = \exp\left[-2\log(2) \cdot \left(\frac{x-0.08}{0.854}\right)^2\right] \cdot \sin^6\left(5\pi\left(x^{\frac{3}{4}} - 0.05\right)\right)$	1	1	$0 \leq x < 1$
$f_8(x) = \tan^{10}(12\pi x)$	5	1	$-10 \leq x < 10$
$f_9(x_1, x_2) = -\cos(x_1 \times \cos(x_2)) \times \exp(-(x_1 - \pi)^2 + (x_2 - \pi)^2)$	1	2	$-100 < x_j < 100, j=1,2$
$f_{10}(x_1, x_2) = x_1^2 + x_2^2$	1	2	$-5.12 < x_j < 5.12, j=1,2$

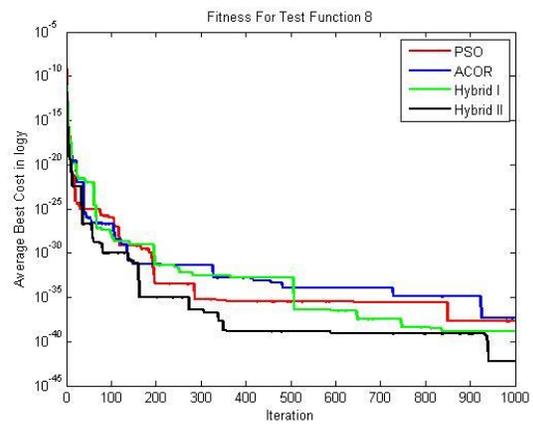
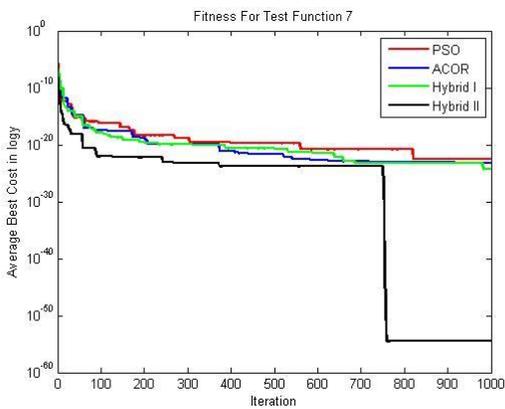
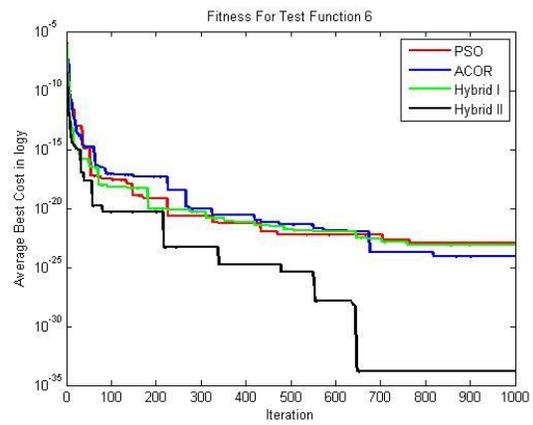
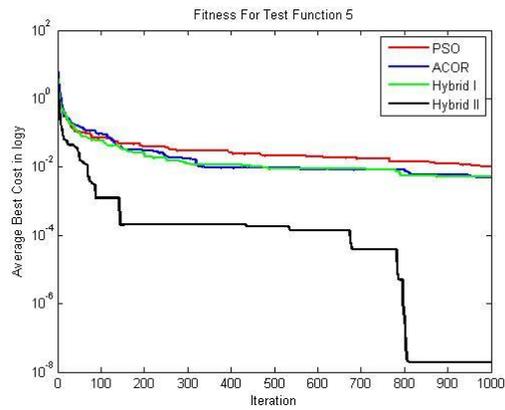
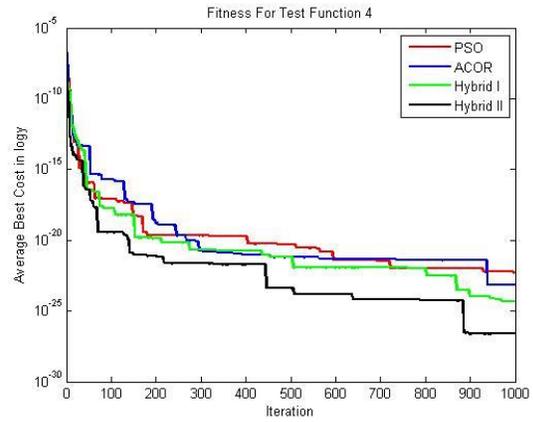
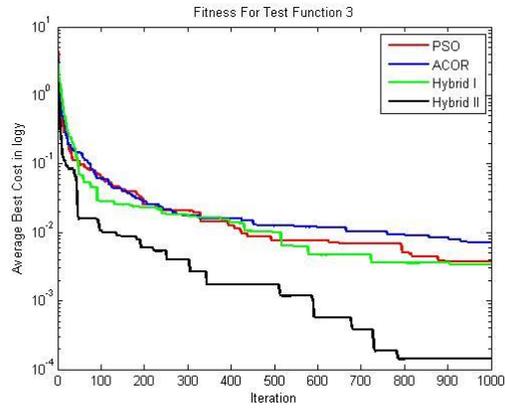
Table 2. System parameter setting

Parameter	ACOR	PSO	Hybrid Models
Learning rate C1		2	2
Learning rate C2		2	2
Learning rate q	0.9		0.9
Evaporation rate	1		1
Number of particles		20	
Size of Pheromone-particle table			10
Number of ants	30		20
Number of iterations	1000	1000	1000

Table 3. Average results of 20 different runs for PSO, ACOR and proposed methods algorithms on benchmark functions are presented in Table 1

Test Function	PSO		ACOR		Hybrid I		Hybrid II	
	Average Best Cost	Average Cpu Time (s)	Average Best Cost	Average Cpu Time (s)	Average Best Cost	Average Cpu Time (s)	Average Best Cost	Average Cpu Time (s)
F <sub>1</sub>	0.38774	0.6136	0.31388	0.6645	0.26528	1.0822	<b>0.050961</b>	<b>1.5672</b>
F <sub>2</sub>	0.34911	0.6580	0.37914	0.7141	0.24491	1.1154	<b>0.11437</b>	<b>1.5848</b>
F <sub>3</sub>	0.26575	0.6113	0.3443	0.6585	0.24099	1.0842	<b>0.095611</b>	<b>1.5890</b>
F <sub>4</sub>	8.3681e-014	0.6346	2.0103e-013	0.6733	2.3859e-013	1.1230	<b>1.0395e-014</b>	<b>1.6617</b>
F <sub>5</sub>	0.31397	0.5937	0.30177	0.6588	0.29456	1.0679	<b>0.060689</b>	<b>1.4670</b>
F <sub>6</sub>	1.2581e-013	0.7884	3.2759e-013	0.7427	2.463e-015	1.6943	<b>1.4233e-015</b>	<b>2.05162</b>
F <sub>7</sub>	4.7057e-013	0.8282	1.3821e-012	0.7574	3.0966e-014	1.3979	<b>3.1345e-017</b>	<b>2.1943</b>
F <sub>8</sub>	8.2144e-025	0.6401	2.5625e-020	0.6807	1.1819e-020	1.1346	<b>3.4216e-023</b>	<b>1.7215</b>
F <sub>9</sub>	-0.99689	0.584	-0.9978	0.64	-0.9999	1.0507	<b>-1</b>	<b>1.519</b>
F <sub>10</sub>	0.1241	0.54	0.054	0.642	0.0087	1.1245	<b>0.000084</b>	<b>1.4511</b>





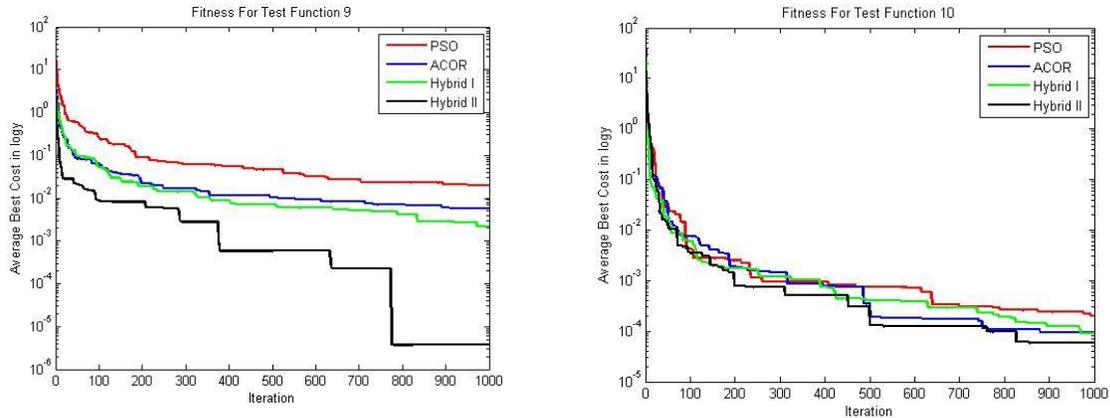


Fig. 4. Graph changes mean the best cost of PSO, ACOR and Hybrid algorithms in 20 different runs on standard benchmark functions at each iteration (X-axis represents the iteration counter, and the y-axis shows the average value of the best cost of algorithm in 20 repeated).

In above implementations, the proposed method has found all global optimum points related to each test function in all test functions in 20 different performances. In other words, the proposed method has had 100 percent success in finding the optimal global points of test functions in Table 1, but PSO and ACOR methods have 100% success rate only for test functions of  $F_1$ ,  $F_2$ ,  $F_5$  and  $F_9$  and Hybrid I have 100% success rate only for test functions of  $F_1$ ,  $F_2$ ,  $F_5$ ,  $F_9$  and  $F_{10}$ . One of the main reasons for the superiority of proposed method compared to other methods is to maintain population diversity while generating new solutions and the new population diversity prevents premature convergence and entrapment in local optima. In this method, superior solutions produced by methods of PSO and ACOR are maintained in each iteration of solution archiving.

## 7. Conclusion

This paper presents a combinational method of two particle swarm optimization and continuous ant colony algorithms for multimodal optimization. The proposed method was compared with PSO, ACOR and sequence with enlarged pheromone-particle table methods and results of this comparison are presented in Section 7. As results of implementation show, using proposed combinational method for multimodal optimization has improved the accuracy of optimal answers compared to PSO, ACOR and sequence with enlarged pheromone-particle table methods.

## References

- [1] H. Nezam Abadi Poor, M. Rostami, "generalization of GCBPSO algorithm", 12<sup>th</sup> Conference on software engineering, Iran. (2008) 29-35.
- [2] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", IEEE. (1995) 1942-1948.
- [3] K. Socha, M. Dorigo, "Ant colony optimization for continuous domains", European Journal of Operational Research. 185(3) (2008) 1155-1173.
- [4] C. Blum, "Ant colony optimization: Introduction and recent trends", Physics of Life Reviews. 2(4) (2005) 353-373.
- [5] T.W. Liao, "Two hybrid differential evolution algorithms for engineering design optimization", Applied Soft Computing. 10(4) (2010) 1188-1199.
- [6] S. Kamali, "Portfolio Optimization using Particle Swarm Optimization and Genetic Algorithm", Journal of mathematics and computer science 10 (2014), 85-90.

- [7] R. Shakerian, S. H. Kamali, M. Hedayati, M. Alipour, “*Comparative Study of Ant Colony Optimization and Particle Swarm Optimization for Grid Scheduling*”, the Journal of Mathematics and Computer Science 2(3) (2011) 469-474.
- [8] CL. Huang, WC. Huang, HY. Chang, YC. Yeh, CY. Tsai, “*Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering*”, Applied Soft Computing. 13(9) (2013)3864-3872.
- [9] Y. Tung Kao, E. Zahara, “*A hybrid genetic algorithm and particle swarm optimization for multimodal functions*”, applied Soft Computing. 8 (2008) 849–857.