I.J. Intelligent Systems and Applications, 2012, 2, 49-57
Published Online March 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2012.02.06

Imperialist Competitive Algorithm with Adaptive Colonies Movement

Helena Bahrami
Dept. of Electronic, Computer and IT,
Qazvin Azad University, Qazvin, Iran,
Email: bahramihelena@yahoo.com

Marjan Abdechiri
Dept. of Electronic, Computer and IT,
Qazvin Azad University, Qazvin, Iran,
Young Researchers Club, Mobarakeh Branch, Islamic Azad University, Isfahan, Iran
Email: Marjan.Abdechiri@qiau.ac.ir

Mohammad Reza Meybodi
Dept. of Computer Engineering and Information Technology,
Amirkabir University of Technology, Tehran, Iran,
Email: mmeybodi@aut.ac.ir

Abstract—**The novel Imperialist Competitive Algorithm (ICA) that was recently introduced has a good performance in some optimization problems. The ICA inspired by socio-political process of imperialistic competition of human being in the real world. In this paper, a new Imperialist Competitive Algorithm with Adaptive Radius of Colonies movement (ICAR) is proposed. In the proposed algorithm, for an effective search, the Absorption Policy changed dynamically to adapt the radius of colonies movement towards imperialist's position. The ICA is easily stuck into a local optimum when solves high-dimensional multi-modal numerical optimization problems. To overcome this shortcoming, we use probabilistic model that utilize the information of colonies positions to balance the exploration and exploitation abilities of the Imperialist Competitive Algorithm. Using this mechanism, ICA exploration capability will enhance. Some famous unconstraint benchmark functions used to test the ICAR performance. Simulation results show this strategy can improve the performance of the ICA algorithm significantly.**

Index Terms—**Imperialist Competitive Algorithm; Absorption Policy; Density Probabilistic Model; Evolutionary Algorithm.**

I. Introduction

Development of appropriate algorithms for solving complex problems such as optimization and search problem has traditionally been applicable one of the most important issues in Computer Science. The global optimization methods have been applications in many fields of science for example business and engineering. The main concern for the optimization techniques is if there would be several local optimums in the system. During the last decade, evolutionary Algorithms (EAs) have been applied to optimization problems in a various areas [1]. In artificial intelligence, an EA is a generic population-based meta-heuristic algorithm which has a subset of evolutionary computation. In fact, a typical EA model using the evolution process of biological populations which can adapt to the changing environment to find the optimum value through the candidate solutions. In other words, EAs are optimization techniques that work on a set of population or individuals by applying stochastic operators to them to find an optimum solution.

Several different EAs have been developed for optimization which among them we can point to the following methods.

The first is Genetic Algorithm (GA) proposed by Holland [2] which is inspired from the biological evolutionary process. Particle Swarm Optimization algorithm proposed by Kennedy and Eberhart [3], in 1995. Simulated Annealing [4] is designed by use of thermodynamic effects and Cultural Evolutionary algorithm (CE), developed by Reynalds and Jin [5], in the early 1990s. The ant colony optimization algorithm (ACO) mimics the behavior of ants foraging for food [6]. Differential evolution (DE) is another optimization algorithm. The DE method is originally due to Storn and Price [7] and works on multidimensional real-valued functions which are not necessarily continuous or differentiable. Harmony search (HS) is a phenomenon-mimicking algorithm inspired by the improvisation process of musicians [8]. Artificial Immune System AIS, simulate biological immune systems [9].

As a newly developed type of meta-heuristic algorithm, the charged system search (CSS) is introduced for design of structural problems [9]. The

Gravitational Search Algorithm (GSA) presented by Rashedi et al. [10, 11] is introduced using physical phenomena.

The Imperialist Competitive Algorithm (ICA) has been proposed by Atashpaz-Gargari and Lucas [12] which has not inspired natural phenomenon, but of course from a socio-human from phenomenon. This algorithm has looked at imperialism process as a stage of human's socio-political evolution. In the ICA algorithm, the colonies move towards the imperialist with a random radius of movement. In [13] CICA algorithm has been proposed that improved performance of ICA algorithm by the chaotic maps are used to adapt the angle of colonies movement towards imperialist's position to enhance the escaping capability from a local optima trap. The ICA algorithm has been used to design an optimal controller which not only decentralizes but also optimally controls an industrial Multi Input Multi Output (MIMO) Evaporator system [14]. ICA algorithm is used for reverse analysis of an artificial neural network in order to characterize the properties of materials from sharp indentation test [15]. The Chaotic Imperialist Competitive Algorithm (CICA) is used for Neural Network Learning [16].

In this paper, we have proposed a new algorithm called Imperialist Competitive Algorithm with Adaptive Radius of Colonies movement (ICAR) that uses the probability density function to adapt the radius of colonies movement towards imperialist's position during iterations dynamically. This mechanism, enhance the global search capability of the algorithm. This idea increases the performance of the ICA algorithm effectively in solving the optimization problems.

We examined the proposed algorithm in several standard benchmark functions that usually are tested in Evolutionary Algorithms. The empirical results obtained from the implementation of the algorithm indicate that the speed of convergence and the quality of solutions are better than the ICA, PSO (Using a Sugeno function as inertia weight decline curve) Differential Evolution (DE) and GA algorithms.

The rest of this paper organized as follows. Section 2, provides a set of benchmark functions. Section 3, presents the ICA algorithm. In section 3, Imperialist Competitive Algorithm with Adaptive Radius of Colonies movement is proposed. Section 4 is devoted to the empirical results of proposed algorithm and their Comparisons with the results obtained by ICA, PSO, DE and GA algorithms. The last section includes conclusions and future works.

II. Related Work

A. Benchmark Functions

Many real-world problems can be formulated as optimization problems most of which can be converted to the following form:

Minimize f(x), x = $[x_1, x_2, \ldots, x_D]$, $Where\ x \in [\,\underline{x}, \overline{x}\,]$ (1)

In the papers, the mentioned algorithms apply some well-known benchmark functions to show their abilities. In order to compare and evaluate different algorithms, various benchmark functions with various properties have been proposed. This section provides a set of benchmark functions for large-scale numerical optimization.

B. Definition

In the following we introduce some well-known benchmark functions and their properties:

1. The Sphere Function

The Sphere function is a highly convex, uni-modal test function and is defined as:

$$f1(x) = \sum_{i=1}^{D} x_i^2 \quad (1)$$

where x is a n-dimensional vector located within the range [-100, 100]. The global minimum is zero.

2. The Rastrigin Function

Rastrigin's multi-modal function [17] is defined as:

$$f2(x) = \sum_{i=1}^{D}(x_i^2 - 10 \times \cos(2\pi x_i) + 10) \quad (2)$$

where x is an n-dimensional vector located within the range [-5.12, 5.12]. The location of local minima is regularly distributed. The global minimum is located at the origin and its function value is zero.

3. The Rosenbrock's Function

Rosenbrock's function is mathematically defined as:

$$f3(x) = \sum_{i=1}^{D-1}(100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (3)$$

where x is an n-dimensional vector located within the range [-30, 30]. The global minimum is zero.

This function exhibits a parabolic-shaped deep valley. To find the valley is trivial, but to achieve convergence to the global minimum is a difficult task. In the optimization literature it is considered a difficult problem due to the nonlinear interaction between variables.

4. The Griewank Function

The mathematical formula that defines Griewank's function is [18]

$$f6(x) = \frac{1}{4000} \times \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (4)$$

where x is an n-dimensional vector located within the range [-600,600]. The location of local minima is regularly distributed. The global minimum is located at the origin and its value is zero.

5. The Michalewicz Function

Michalewicz's function is a parameterized, multi-modal function with many local optima located between plateaus. In our experiments, m is set to 10. The formula defines as:

$$f4(x) = -\sum_{i=1}^{n} \sin(x_i)\,[sin\frac{ix_i^2}{\pi}]^{2m} \quad (5)$$

where x is a n-dimensional vector that is normally located within the range [-π, π].

6. The Ackley's Function

Ackley's function is defined as:

$$f5(x) = -20\,exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_j^2})- \exp\,(\frac{1}{n}\sum_{k=1}^{n}\cos 2\pi x_k) + 20 + \text{e} \quad (6)$$

where x is a n-dimensional vector that is normally located within the range [-32, 32]. Ackley's function is a highly multimodal function with regularly distributed local optima. The global minimum is zero.

7. The Booth Function

The definition of the Booth function is:

$$f7 = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad (7)$$

where *n* is 2 (number of variables). Search domain is $-10 \le x_i \le 10,\ i = 1, 2.$ This function has several local minima. The global minimum is *zero*.

8. The Zakharov Function

The function is

$$z_n(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4 \quad (8)$$

where the search domain is $-5 \le x_i \le 10,\ i = 1, 2, \ldots, n$. The global minimum is *0*.

9. The Trid Function

The Trid function is

$$f(x) = \sum_{i=1}^{n} ix_i^2 \quad (9)$$

where the search domain is $-n^2 \le x_i \le n^2,\ i = 1, 2, \ldots, n.$ There is not local minimum except the global one. The global minimum is *-50* for *n=6 and -200* for *n=10.*

10. Sum Squares Function

The sum squares function is

$$f(x) = \sum_{i=1}^{n} ix_i^2 \quad (10)$$

where the search domain is $-10 \le x_i \le 10,\ i = 1, 2, \ldots, n.$ The global minimum is *0.*

11. The Schwefel Function

Definition:

$$f9 = 418.9829n - \sum_{i=1}^{n}(x_i \sin\sqrt{|x_i|}) \quad (11)$$

Search domain is $-500 \le x_i \le 500,\ i = 1, 2, \ldots, n.$ Number There are several local minima. The global minimum is zero.

12. The Branin Function

The formula of the function is

$$f8 = (x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_i + 10 \quad (12)$$

where n is 2 that is number of variables. Search domain is $-5 \le x_1 \le 10,\ 0 \le x_2 \le 15.$ This function has one global minimum. The global minimum is *0.397887*.

III. Introduction of Imperialist Competitive Algorithm

Imperialist Competitive Algorithm (ICA) is a new evolutionary algorithm in the Evolutionary Computation field based on the human's socio-political evolution. The algorithm starts with an initial random population called countries. Some of the best countries in the population selected to be the imperialists and the rest form the colonies of these imperialists. In an N dimensional optimization problem, a country is a $1 \times N$ array. This array defined as below

$$country = [p_1, p_2, \ldots, p_N] \quad (13)$$

The cost of a country is found by evaluating the cost function *f* at the variables $(p_1, p_2, p_3, \ldots, p_N)$. Then

$$c_i = f(country_i) = f(p_{i1}, p_{i2}, \ldots, p_{iN}) \quad (14)$$

The algorithm starts with initial *N* countries and the N_{imp} best of them (countries with minimum cost) chosen as the imperialists. The remaining countries are colonies that each belong to an empire. The initial colonies belong to imperialists in convenience with their powers. To distribute the colonies among imperialists proportionally, the normalized cost of an imperialist is defined as follow

$$C_n = max_i\, c_i - c_n \quad (15)$$

Where, $cost_n$ is the cost of *n*th imperialist and C_n is its normalized cost. Each imperialist that has more cost value, will have less normalized cost value. Having the normalized cost, the power of each imperialist is calculated as below and based on that the colonies distributed among the imperialist countries.

$$p_n = \left|\frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}\right| \quad (16)$$

On the other hand, the normalized power of an imperialist is assessed by its colonies. Then, the initial number of colonies of an empire will be

$$NC_n = rand\{p_n.(N_{col})\} \quad (17)$$

Where, NC_n is initial number of colonies of nth empire and N_{col} is the number of all colonies.

To distribute the colonies among imperialist, NC_n of the colonies is selected randomly and is assigned to their imperialist. The imperialist countries absorb the colonies towards themselves using the absorption policy. The absorption policy shown in Fig. 1, makes the main core of this algorithm and causes the countries move towards to their minimum optima. The imperialists absorb these colonies towards themselves with respect to their power that described in Eq. (18). The total power of each imperialist is determined by the power of its both parts, the empire power plus percents of its average colonies power.

$$TC_n = cost(imperialist_n) + \xi mean\{cost(colonies\ of\ empire_n\} \quad (18)$$

Where TC_n is the total cost of the nth empire and ξ is a positive number which is considered to be less than one.

$$x \sim U(0, \beta \times d) \quad (19)$$

In the absorption policy, the colony moves towards the imperialist by x unit. The direction of movement is the vector from colony to imperialist, as shown in Fig.1, in this figure, the distance between the imperialist and colony shown by d and x is a random variable with uniform distribution. Where β is greater than 1 and is near to 2. So, a proper choice can be $\beta = 2$. In our implementation γ is $\pi/4$ (Rad) respectively.

$$\theta \sim U(-\gamma, \gamma) \quad (20)$$

In ICA algorithm, to search different points around the imperialist, a random amount of deviation is added to the direction of colony movement towards the imperialist. In Fig. 1, this deflection angle is shown as θ, which is chosen randomly and with an uniform distribution. While moving toward the imperialist countries, a colony may reach to a better position, so the colony position changes according to position of the imperialist.

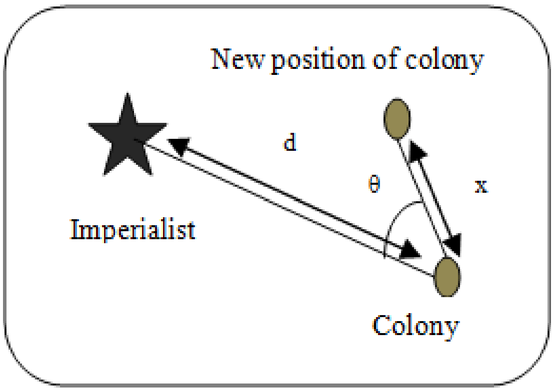


Figure1. Moving colonies toward their imperialist

In this algorithm, the imperialistic competition has an important role. During the imperialistic competition, the weak empires will lose their power and their colonies. To model this competition, firstly we calculate the probability of possessing all the colonies by each empire considering the total cost of empire.

$$NTC_n = max_i\{TC_i\} - TC_n \quad (21)$$

Where, TC_n is the total cost of nth empire and NTC_n is the normalized total cost of nth empire. Having the normalized total cost, the possession probability of each empire is calculated as below

$$p_{p_n} = \left| \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i} \right| \quad (22)$$

after a while all the empires except the most powerful one will collapse and all the colonies will be under the control of this unique empire. The sequence of ICA algorithms shown in Fig.2.

(1) Initialize the empires and their colonies positions randomly.
(2) Colonies move towards the imperialist's position).
(3) Compute the total cost of all empires (Related to the power of both the imperialist and its colonies).
(4) Pick the weakest colony (colonies) from the weakest empire and give it (them) to the empire that has the most likelihood to possess it (Imperialistic competition).
(5) Eliminate the powerless empires.
(6) If there is just one empire, then stop else continue.
(7) Check the termination conditions

Figure2. The sequence of ICA algorithm.

IV. THE PROPOSED ALGORITHM

The ICA algorithm like many Evolutionary Algorithms suffers the lack of ability to global search properly in the problem space. During the search process, the algorithm may trap into local optima and it is possible to get far from the global optima. This causes the premature convergence.

In this paper, a new method suggested that balance the exploration and exploitation abilities of the proposed algorithm, using colonies positions information. In the ICA algorithm absorption policy, mentioned in the previous section, the colonies move towards imperialists with a radius which is a random variable. The colonies movement because of the constant x parameter has a monotonic nature, so the colonies movement could not be adapted with the search process. Therefore, if the algorithm traps in the local optima, it cannot leave it and move towards the global optima. For solving this problem and making balance between the exploration and exploitation we define the x parameter adaptively in the search space.

A. the definition of adaptive movement radius in the absorption policy

As mentioned before in ICA algorithm the colonies move towards the imperialist by a random radius. The x parameter is radius. In this paper, we extract the statistical information about the search space from the current population of solutions to provide an adaptive movement radius. We proposed a probabilistic model, to modify the ICA global search capability. The probabilistic model *P(x)* that we use here is a Gaussian distribution model [19, 20]. The joint probability distribution of each country is given by the product of the marginal probabilities of the countries:

$$p(Country) = \prod_{i=1}^{n} N(Country_i\,; \mu_i, \sigma_i), \quad (23)$$

where

$$N(Country_i; \mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi\sigma_i}} e^{\frac{1}{2}\left(\frac{Country_i-\mu_i}{\sigma_i}\right)^2}, \quad (24)$$

the average, μ, and the standard deviation, σ, for the colony countries of each empire is approximated as below [21,22]:

$$\hat{\mu_i} = \overline{Country_i} = \frac{1}{M}\sum_{t=1}^{M} Country_{t,i} \quad (25)$$

$$\hat{\sigma_i} = \sqrt{\frac{1}{M}\sum_{t=1}^{M}(Country_{t,i} - \overline{Country_i}\,)^2} \quad (26)$$

During iterations, the countries densities compute using the probabilistic model in Eq. (23). If the countries density in the current iteration is more than the previous iteration, then with %15 the previous radius of the movement of the countries towards their empires will be shrunk and with %85 the mentioned radius will be expanded.

$$x_{iter} = 0.15(x_{iter-1} + \alpha) + 0.85(x_{iter-1} - \alpha) \quad (27)$$

x_{iter} , is the current radius of movement. x_{iter-1} , is the previous radius and α is the step size of shrinking and expanding the radius of movement. The value of this step size is varying from 0.0001 to 0.1.

Otherwise, if the countries density in the current iteration is less than the previous iteration, then with %15 the previous radius of the movement of the countries towards their empires will be expanded and with %85 the radius will be shrunk.

$$x_{iter} = 0.85(x_{iter-1} + \alpha) + 0.15(x_{iter-1} - \alpha) \quad (28)$$

If the countries density in the current iteration is more than the previous iteration, it means that may be the countries are converging to an optimum point. So, in Eq. (27), depending on the density of the countries distribution, we set the radius of movement so that each country can escape from the dense area with %85 and with %15 the country will move towards its empire with a shrinking radius. In the cases that the countries converge to local optima, this method will help to escape from falling into the local optima's trap with possibility of %85. In this way, we add explorative search ability to the ICA algorithm.

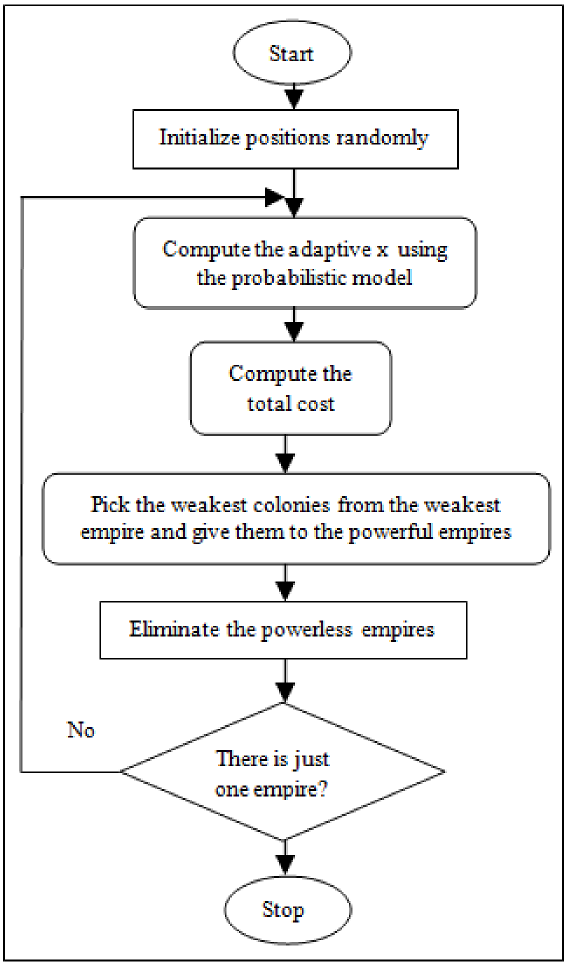


Figure3. The ICAR Algorithm.

In Eq. (28), if the countries density in the current iteration is less than the previous iteration, each country with possibility of %85 will move towards its empire with a shrinking radius and with %15 the country will move towards its empire with an expanding radius. This way, provides a more efficient search in all over the search space of the problem.

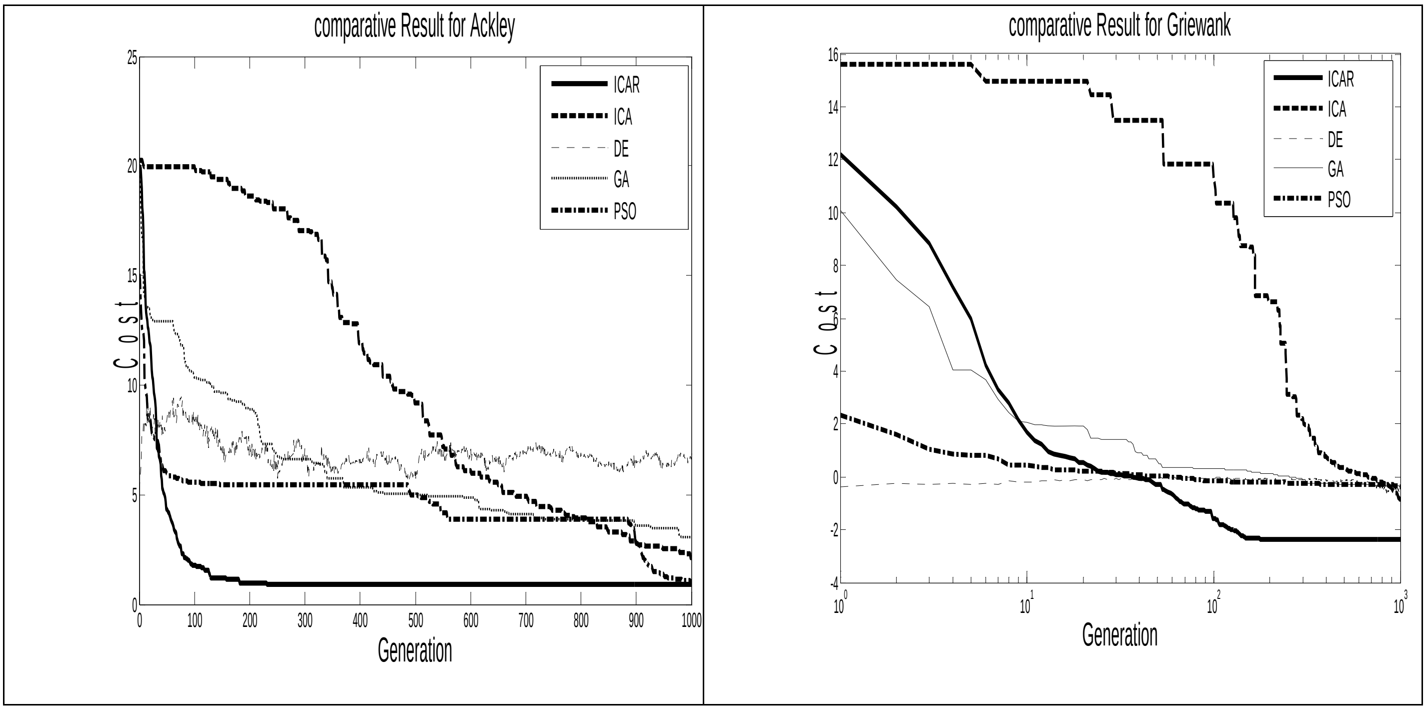
The results show that the quality of solutions and the speed of convergence of imperialist competitive algorithm with adaptive absorption policy are better than to ICA, PSO, DE and GA algorithms. This is observable in analysis and conclusion section. The flowchart of ICAR can be seen in Fig. 3.

V. Analysis and Consideration of Empirical Results

In this paper, the proposed algorithm, that called Imperialist Competitive Algorithm with Adaptive Radius of Colonies movement (ICAR), applied to some well-known benchmark functions in order to verify the ICAR algorithm performance and compared with ICA and PSO, DE and GA algorithms. These benchmarks presented in Table1.

TABLE I. BENCHMARKS FOR SIMULATION.

	Mathematical Representation	Range
Sphere	$f1(x) = \sum_{i=1}^{D} x_i^2$	(-100,100)
Rastrigrin	$f2(x) = \sum_{i=1}^{D}(x_i^2 - 10 \times \cos(2\pi x_i) + 10)$	(-10,10)
Rosenbrock	$f3(x) = \sum_{i=1}^{D-1}(100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	(-100,100)
Griewank	$f4(x) = \frac{1}{4000} \times \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	(-600,600)
Michalewicz	$f5(x) = -\sum_{i=1}^{n} \sin(x_i)\, [sin \frac{ix_i^2}{\pi}]^{2m}$	$(0,\pi)$
Ackley	$f6(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_j^2}) - \exp(\frac{1}{n}\sum_{k=1}^{n} \cos 2\pi x_k) + 20 + e$	(-32,32)
Booth	$f7 = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	(-10,10)
Zakharov	$f8(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$	(-5,10)
Trid	$f9(x) = \sum_{i=1}^{n} ix_i^2$	$(-n^2, n^2)$
Sum Squares	$f10(x) = \sum_{i=1}^{n} ix_i^2$	(-10,10)
Schwefel	$f11(x) = 418.9829n - \sum_{i=1}^{n}(x_i \sin\sqrt{\lvert x_i \rvert})$	(-500,500)
Brain	$f12(x) = (x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_i + 10$	$-5 \le x_1 \le 10,\ 0 \le x_2 \le 15$



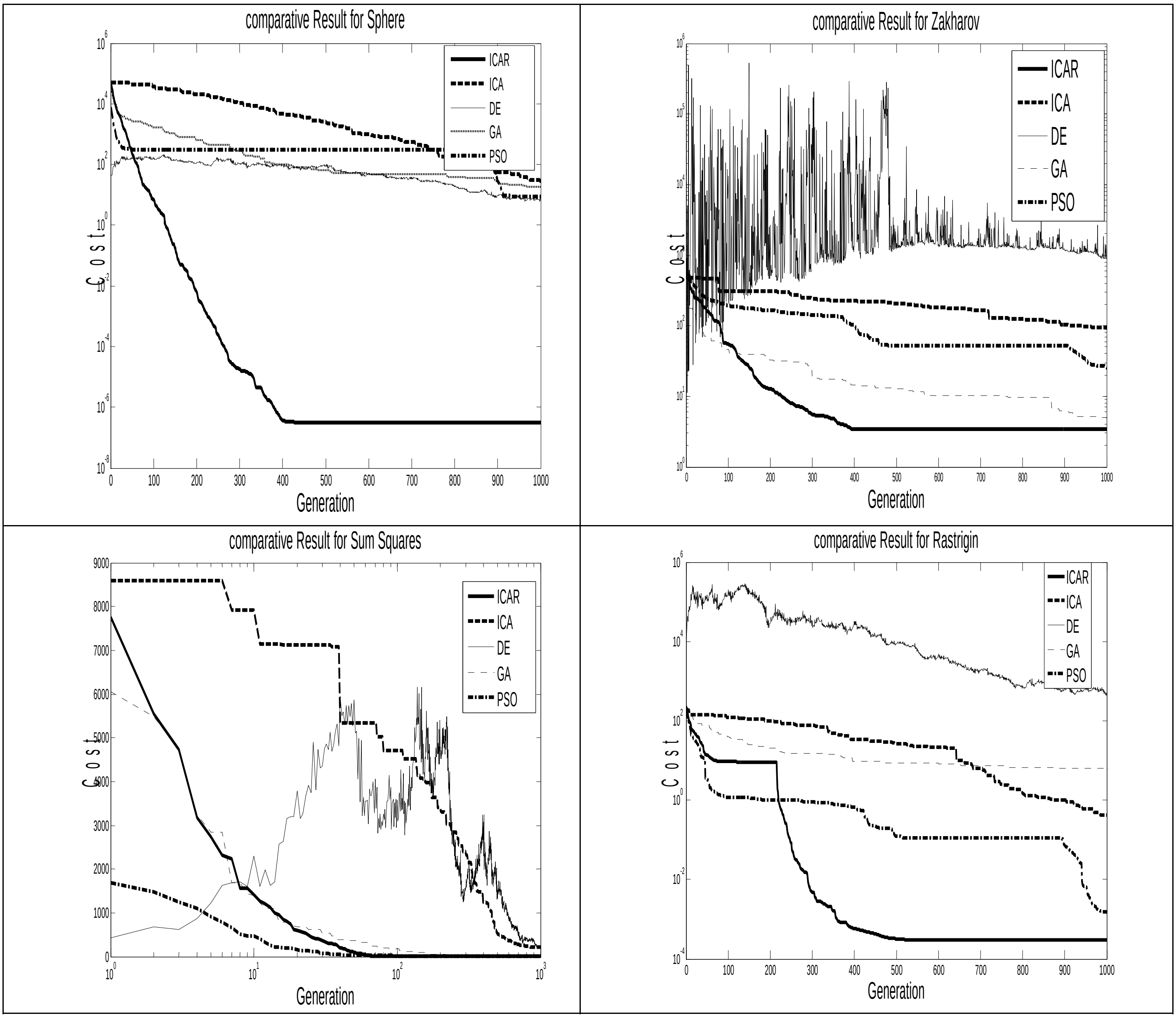


Figure4. The experimental results for Rasrigin, Griewank , Zakharov , Ackley , Sphere and Sum Squares functions

We made simulations for considering the rate of convergence and the quality of the proposed algorithm optima solution, in comparison to ICA, PSO using a Sugeno function as inertia weight, DE and GA algorithms that all the benchmarks tested by 10, 20 and 30 dimensions separately. The average of optimum value for 20 trails obtained. In these experiments, all the simulations are done during 1000 generations for the benchmarks functions which are presented in Table1. In these simulations for ICAR and ICA algorithms, we set the β parameters to 2 and α to 0.001. The number of imperialists and the colonies are set respectively to 8 and 80. In PSO, $c_1 = c_2 = 1.5$ and inertia factor (w) is decreasing linearly from 0.9 to 0.2. In GA, crossover, Gaussian mutation and roulette wheel selection are used. The crossover and mutation probabilities were set to 0.4 and 0.01. The results of these experiments presented in Table 2. Fig.4 shows the performance over time for some of our experiments, they all have average best fitness plotted for each evaluation for the implemented ICA and PSO using a Sugeno function as inertia weight, DE and GA. The results on performance clearly show that the ICAR is a much stronger optimizing method than ICA and PSO, DE and GA on all the test problems except on the Sum Squares function. As it can be seen in the Sum Squares function, at first iterations ICA won the competition but finally ICAR found the best solution.

In Sphere function it is observable that the quality of global optima solution and the convergence velocity towards the optima point has improved in compare with the other four algorithms. In the log plot of the Sphere function, at the first 20 iterations, PSO and DE algorithms have better convergence speed than the ICA and ICAR algorithms but after that iteration the ICAR won the competition.

As we can see in Rasrigin multi-modal function the PSO algorithm has better performance rather than the ICA, DE and GA algorithms. The proposed algorithm has shown a good performance in this function and has been able to escape from the local peaks and reach to global optima.

In Griewank multi-modal function the proposed algorithm shows good performance and can reach to optima solution and in convergence speed rather than the ICA, DE, PSO and GA algorithms.

In Ackley and Zakharov functions, the proposed algorithm has better performance both in optima solution quality and in convergence speed rather than the ICA, PSO, DE and GA algorithms.

TABEL II. The Average optimum value for 20 trails for benchmarks in 10 dimensions

	GA	PSO	DE	ICA	ICAR
F1	0.2499	1.1405×10^{-17}	4.4822×10^{-4}	6.2799×10^{-11}	2.2349×10^{-25}
F2	1.0993×10^{-11}	0	1.0521×10^{-3}	0	0
F3	48.9673	0.0020	11.0433	0.4466	0.0017
F4	-0.0972	-0.0987	-0.0984	-0.0987	-0.0987
F5	-8.7287	-8.6351	-8.5270	-8.6544	-9.1466
F6	0.1016	0.0298	3.2257	1.9386×10^{-7}	8.6153×10^{-14}
F7	3.2875×10^{-10}	0	0.3986	0	0
F8	0.0515	3.7903×10^{-7}	2.0202	2.1969×10^{-8}	3.3235×10^{-9}
F9	-80.5906	-122.0000	-123.5667	-124.5667	-124.6667
F10	8.2867×10^{-4}	8.9635×10^{-16}	0.0347	4.7998×10^{-13}	8.9895×10^{-28}
F11	1.2728×10^{-5}	1.2728	415.0376×10^{-5}	1.2728×10^{-5}	1.2728×10^{-5}
F12	0.3979	0.3979	0.3982	0.3979	0.3979

TABEL III. The Average optimum value for 20 trails for benchmarks in 20 dimensions

	GA	PSO	DE	ICA	ICAR
F1	3.2958	0.0143	0.1785	0.0732	5.2348×10^{-12}
F2	0.6609	5.4021×10^{-4}	1.8432×10^{-3}	8.4028×10^{-4}	1.5952×10^{-12}
F3	1.3182×10^{3}	19.5173	20.2581	1.9191×10^{3}	14.8710
F4	-0.3373	-0.7114	-0.6238	-0.7206	-0.7225
F5	-14.3368	-14.4732	-14.3710	-14.5085	-14.6277
F6	2.1664	0.2105	3.9297	0.1205	0.0138
F7	1.0993×10^{-11}	0	0.0041	0	0
F8	1.0686	0.5184	31.0009	7.4750	0.0067
F9	-198.9844	-461	-574	-558.3333	-579.9995
F10	0.1397	0.0012	8.1281	0.3489	2.3970×10^{-12}
F11	1.2728×10^{-5}	415.0399	1.2728×10^{-5}	1.2728×10^{-5}	1.2728×10^{-5}
F12	0.3979	0.3979	0.4257	0.3979	0.3979

TABEL IV. The Average optimum value for 20 trails for benchmarks in 30 dimensions

	GA	PSO	DE	ICA	ICAR
F1	18.8398	9.0371	6.4654	28.7678	3.1896×10^{-7}
F2	6.3023	0.0015	436.2848	0.4128	2.9894×10^{-4}
F3	9.8512×10^{3}	28.7533	1.3604×10^{3}	1.0179×10^{6}	26.5384
F4	-0.3702	-0.3687	-0.5546	-0.8319	-2.3712
F5	-20.1838	-17.4673	-20.3587	-21.0112	-22.6101
F6	3.0695	1.0289	6.6120	19.9587	0.9313
F7	1.9588×10^{-9}	0	0.0046	0	0
F8	4.8792	25.7779	893.9510	93.3675	3.3848
F9	-262.1999	-1.3100×10^{3}	-1.2053×10^{3}	-1.3260×10^{3}	-1.3442×10^{3}
F10	5.4177	0.0764	198.2871	211.5959	1.4994×10^{-7}
F11	1.2728×10^{-5}	1.2728×10^{-5}	124.0356	1.2728×10^{-5}	1.2728×10^{-5}
F12	0.3979	0.3979	0.4014	0.3979	0.3979

In table 2,3 and 4 the average of optimum value for 20 trails for 10,20 and 30 dimensional which are obtained from proposed algorithm, ICA, DE, PSO and GA are shown. Table 3 and 4 are shown experimental results for 20 and 30 dimensional respectively and the stop condition was 1000 generations. The numerical results show that the proposed algorithm has recovered the global optima solution remarkably.

The numerical results show that the proposed algorithm improves optima solution significantly.

VI. Conclusion

In this paper, a new imperialist algorithm is called Imperialist Competitive Algorithm with Adaptive Radius of Colonies movement (ICAR) is introduced. The proposed algorithm uses the probability density function to adapt the radius of colonies movement towards imperialist's position during iterations dynamically. This mechanism, enhance the global search capability of the algorithm. This idea balances the exploration and exploitation abilities of the proposed algorithm, using colonies positions information. We examined the proposed algorithm in several standard benchmark functions that usually tested in Evolutionary Algorithms. Experimental results show that the proposed algorithm is a promising method with good global convergence performance than the ICA, GA, DE and PSO algorithms. In the future, we will work on the effect of the different probability models on the performance of the proposed algorithm.

REFERENCE

[1]H. Sarimveis and A. Nikolakopoulos, "A Line Up Evolutionary Algorithm for Solving Nonlinear Constrained Optimization Problems," *Computers & Operations Research*, 32(6):pp.1499–1514, 2005.

[2]M. Melanie, "An Introduction to Genetic Algorithms," *Massachusett's MIT Press*, 34(7):pp.1-9, 1999.

[3]J. Kennedy and R.C. Eberhart, "Particle swarm optimization," *in: Proceedings of IEEE International Conference on Neural Networks*, Piscataway: IEEE, pp. 1942–1948, 1995.

[4]L. A. Ingber, "Simulated annealing: practice versus theory," *J. Math. Comput. Modell.,* 18(11):pp. 29–57, 1993.

[5]B. Franklin and M. Bergerman, "Cultural Algorithms: Concepts and Experiments," *In Proceedings of the IEEE Congress on Evolutionary Computation*, 2: pp. 1245–1251, 2000.

[6]M. Dorigo, V. Maniezzo and A. Colorni, "The ant system: optimization by a colony of cooperating agents," *IEEE Transaction System Man Cybern*, B 26(1):pp. 29–41, 1996.

[7]R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, 11(4):pp. 341–359, 1997.

[8]K. Lee and Z. Geem, "A new structural optimization method based on the harmony search algorithm," *Computers and Structures*, 82:781-98, 2004.

[9]F. J. Von Zuben and L. N. De Castro, "Artificial Immune Systems: Part I - Basic Theory and Applications," *School of Computing and Electrical Engineering*, State University of Campinas, Brazil, Technical Report DCA-RT 01/99, 1999.

[10]A. Kaveh, S. Talatahari "A novel heuristic optimization method: Charged system search," Acta Mechanica, doi:10.1007/s00707-009-0270-4.

[11]E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "A Gravitational Search Algorithm," *Information Science, Special Section on High Order Fuzzy Sets*, 179(13): pp. 2232-2248, 2009.

[12]E. Atashpaz-Gargari and C. Lucas, "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition," *IEEE Congress on Evolutionary Computation (CEC 2007)*, pp 4661-4667, 2007.

[13]H. Bahrami, K. Faez, M. Abdechiri, "Imperialist Competitive Algorithm using Chaos Theory for Optimization," *UKSim-AMSS 12th International Conference on Computer Modeling and Simulation,* 2010.

[14]R. Rajabioun, F. Hashemzadeh, E. Atashpaz-Gargari, B. Mesgari and F. Rajaei Salmasi, "Identification of a MIMO Evaporator and Its Decentralized PID Controller Tuning Using Colonial Competitive Algorithm", Accepted to be presented in IFAC World Congress, 2008.

[15]A. Biabangard-Oskouyi, E. Atashpaz-Gargari, N. Soltani and C. Lucas, "Application of Imperialist Competitive Algorithm for materials property characterization from sharp indentation test," *International Journal of Engineering Simulation*, under revision, 1(3):pp. 337-355, 2008.

[16]M. Abdechiri, K. Faez and H. Bahrami, "Neural Network Learning based on Chaotic Imperialist Competitive Algorithm," *The 2nd International Workshop on Intelligent System and Applications (ISA2010)*, 2010.

[17]L. Rastrigin, "External control systems," *In Theoretical Foundations of engineering cybernetics series*. Moscow, Russian, Nauka, 1974.

[18]A. Griewangk "Generalized descent of global optimization," *Optim. Theor. Appl.*, 34: pp. 11–39, 1981.

[19]A. Papoulis, "Probability Random Variables and Stochastic Processes," *McGraw-Hill*, 1965.

[20]R. C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," the International Journal of Robotics Research, 5(4), Winter 1986.

[21]T. K. Paul and H. Iba, "Linear and Combinatorial Optimizations by Estimation of Distribution Algorithms," *9th MPS Symposium on Evolutionary Computation*, IPSJ, Japan, 2002.

[22]Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, "Estimation with Applications to Tracking and Navigation," *John Wiley & Sons*, 2001.

Helena Bahrami was born in Tehran, Iran. She received her B. Sc. degree in Computer Engineering (Software) from Shahrake Ghods Islamic Azad 2006. Now she is student in M. Sc. degrees in Computer Engineering from Qazvin Islamic Azad University.
Her research interests are in Evolutionary Algorithms Image Processing,NeuralNetworks.Email: H_Bahrami@qiau.ac.ir

Marjan Abdechiri was born in Isfahan, Iran. She received her B. Sc. degree in Computer Engineering (Hardware) from Najafabad Islamic Azad 2007. Now she is student in M. Sc. degrees in Computer Engineering from University of Qazvin Islamic Azad University. Her research interests are in Evolutionary Algorithms Image Processing, NeuralNetworks. Emails:Marjan.Abdechiri@gmail.com, Marjan.abdechiri@qiau.ac.ir.

Mohammad Reza Meybodi received the B. Sc. and M. Sc. degrees in economics from Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the M. Sc. and Ph.D. degrees in computer science from Oklahoma University,
USA, in 1980 and 1983, respectively. Currently, he is a Full Professor in Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran. Prior to current position, he worked as an Assistant Professor at Western Michigan University (1983 to 1985), as an Associate Professor at Ohio University, USA (1985 to 1991). His research interests include wireless networks, fault tolerant systems, learning systems, parallel algorithms, soft computing and software development.

