

Improvement of Imperialist Competitive Algorithm based on the Cosine Similarity Criterion of Neighboring Objects

Maryam Houtinezhad and Hamid Reza Ghaffary

Department of Computer Engineering, Ferdows Branch, Islamic Azad University, Ferdows, Iran

Abstract: The goal of optimizing the best acceptable answer is according to the limitations and needs of the problem. For a problem, there are several different answers that are defined to compare them and select an optimal answer; a function is called a target function. The choice of this function depends on the nature of the problem. Sometimes several goals are together optimized; such optimization problems are called multi-objective issues. One way to deal with such problems is to form a new objective function in the form of a linear combination of the main objective functions. In the proposed approach, in order to increase the ability to discover new position in the Imperialist Competitive Algorithm (ICA), its operators are combined with the particle swarm optimization. The colonial competition optimization algorithm has the ability to search global and has a fast convergence rate, and the particle swarm algorithm added to it increases the accuracy of searches. In this approach, the cosine similarity of the neighboring countries is measured by the nearest colonies of an imperialist and closest competitor country. In the proposed method, by balancing the global and local search, a method for improving the performance of the two algorithms is presented. The simulation results of the combined algorithm have been evaluated with some of the benchmark functions. Comparison of the results has been evaluated with respect to metaheuristic algorithms such as Differential Evolution (DE), Ant Lion Optimizer (ALO), ICA, Particle Swarm Optimization (PSO), and Genetic Algorithm (GA).

Keywords: Imperialist competitive algorithm, particle swarm optimization, optimization problem.

Received March 24, 2018; accepted November 17, 2019
<https://doi.org/10.34028/iajit/18/3/1>

1. Introduction

Optimization methods are divided into four categories: Enumerative Method, Calculus-Based Method, Heuristic, and meta-Heuristic. The goal of Heuristic algorithms is to provide a solution within a reasonable time frame that is appropriate to solve the problem; Heuristic algorithms can be combined with optimization algorithms to improve the efficiency of the algorithm. Finding the best answer for a problem is called optimization. The goal of optimization is to find the best answer acceptable considering the limitations of the problem. Each optimization problem has a number of independent variables. The optimization variables are set according to the minimum or maximization of the target function. On the other hand, selecting the appropriate target function is one of the most important steps in optimization. Real world optimization issues have several optimal points. In such cases, if all of these points are close to the solution to the problem, each of these points is considered a local optimal and the largest one is called global optimal. Different algorithms and several methods have been proposed to finding global optimization, including evolutionary methods. Evolutionary algorithms have the ability to adapt to different conditions of optimization problems. The

most widely used algorithms are Genetic Algorithm (GA) [8], Particle Swarm Optimization (PSO) [5, 15], and Ant Colony Optimization (ACO) [6], Artificial Bee Colonies (ABC) [14], Imperialist Competitive Algorithm (ICA) [2]. But these methods have problems such as early convergence or remain in the optimal local area. In this paper, a new approach is proposed based on the combination of two meta-heuristic algorithms.

The particle swarm optimization algorithm is a collective intelligence-based optimization method in which population particles solve the problem by exchanging information and searching the problem space [15]. This algorithm has a high ability to reach the optimal global solution. Another advantage of the particle swarm optimization is the simplicity of implementation and high speed of computing. Despite its high capabilities, this algorithm has limitations and problems such as early convergence and Failure to maintain the diversity of particle. If the value of the best individual and the best overall is equal, the particle velocity reaches zero and the population gets convergence [12, 15, 17, 24]. In this research, to avoid staying in optimal localization, the combination of particle swarm optimization algorithm and imperialist competitive algorithm [2] have been used. The

proposed method improves the performance of the algorithm by balancing global and local search. The core of this algorithm is the policy of assimilation, colonial competition, and revolution. In this algorithm, during the repetitive process, the answers are improved to finally solve the optimization problem.

2. Related Work

Optimization methods and algorithms are categorized into two groups of exact algorithms and approximate algorithms. Deterministic algorithms are capable of finding the optimal solution accurately, but they are not efficient enough for NP-Hard optimization problems and their execution time increases exponentially with the dimensions of the problems. Approximate algorithms are capable of finding good (near optimal) solutions at a short time for NP-Hard optimization problems. Approximate algorithms are divided into three groups of heuristic algorithms and meta-heuristic algorithms and hyper heuristic algorithms. Meta-heuristic Algorithms are one of a variety of approximate optimization algorithms that provide exit strategies from local optimal points. On the other hand, stochastic methods are divided into two groups of single-solution and population-based. In the first technique: optimization starts with a random solution and improves it with repetition. Other algorithms that use this technique are Simulation Annealing (SA) algorithms [16], hill climbing [20], and local search [9, 13]. The simplicity of implementation and a low number of evaluation functions are the benefits of this technique. However, this technique is weak in finding the optimal local point, because only one object in the search space is in progress and so they cannot share information with other objects. In the population-based optimization technique, the optimization process starts with a set of random solutions and improves with repetition. A set of solutions are considered as desirable solutions. The population-based algorithms are divided into three categories: evolution, physics, and particles. This algorithms include PSO, GA, ACO [6, 7], ABC [14], Differential Evolution (DE) [23] and etc., Sharing information in this technique is one of its benefits. All objects are involved in this technique. Unlike the previous technique, this technique has high evaluation functions as well as more complex implementation. However, population-based techniques are more likely to be accepted. Despite all the advantages of the inventive algorithms, it cannot be said that all of them are used to solve all the problems [25]. For this reason, their combination is effective in achieving the desired response. In each of the algorithms, there is a series of factors in the optimization. For example, in PSO, the inertia weight keeps particles moving toward the optimum point. In the genetic algorithm, crossover and mutation operators are effective in reaching the

elite chromosomes and having a better generation. The exploitation and exploration of the two opposite points are considered in the algorithms. Therefore, ensuring balancing in these points ensures the optimal solution.

3. Concepts of Optimization Algorithms

3.1. Basic Concept

In this research, the terms and concepts of optimization algorithms have been used, that are briefly described in Table 1.

Table 1. Basic concepts of optimization algorithms.

Single-objective and multi-objective	A single-objective optimization problem has only one objective function. But in a multi-objective problem, the number of functions that are together optimized is more than one.
Discrete and continuous optimization	A discrete optimization problem is an issue in which problem variables have discrete variations in a particular range. While continuous variables, contain infinite numbers of possible values. If one considers the optimal value of $F(x)$ on a domain of real numbers, then a continuous problem is considered.
Search space	The space in which the answer to the problem is acceptable is the search space. Each point of the search space range represents one of the solutions to the problem.
Population	A population is called a set of possible solutions to solve an issue. The population is subset of the search space.
Target function	The target function is a function that expresses the purpose of the problem. If the objective function is algebraic, static optimization and if it is integral, it will be dynamic optimization.
Decision-making	The target function has one or more independent variables which are called decision-making variables. The process of achieving an optimal response is based on the decision variable.
Local and global Optimal	The optimization problem has several maximum and minimum points. Each of these points is called an optimal local. If within the local optimal set to find a point that is better than the other points, then they are called global optimal.
Convergence	When the majority of existing solutions have the same value, the population converges.

3.2. Optimization Algorithm

In this section, after describing the algorithms used, the method of their combination will be explained

3.2.1. Imperialist Competitive Algorithm

ICA, is based on the social life of humans. This algorithm is suitable for optimization problems since various parameters are considered in this algorithm. ICA such as other evolutionary algorithms begins to search and optimization processes with an initial population. The initial population in this algorithm begins on the basis of objects have been known as the country. The countries are divided into two categories of imperialist and colony countries [2]. The best country in the collection is picked as imperialists. The weak country has been considered as colonies. The value of each country is determined in the N-dimensional space. The value of any country according to all depends feature is equal to:

$$C_i = f(\text{country}_i) = f(p_{i1}, p_{i2}, p_{i3}, \dots, p_{iN_{var}}) \quad (1)$$

Standard imperialist competitive algorithm pseudocode is shown in Algorithm 1.

Algorithm 1: imperialist competitive algorithm pseudo-code [6]

1. Initialization and appraise the empires
2. While it follows the rules Do
 - Move The Weak object Toward Their linked Imperialist
 - 3. If There Is A Object In An Empire Which Has A Lower Cost Than The Imperialist Then
 - Switch The Positions Of That Weak object And The Imperialist
 - End If
 - 4. Compute The Total value Of All Strong Object(as empires)
 - Imperialistic Competition
 - 5. If There Is An Empire With No dependent object Then
 - Terminate This Empire
 - End If
- End While

3.2.1.1. Initial Empires (Empire Creation)

The initial division into colonial and imperial groups is according to Equations (2) and (3).

$$p_n = \left\lfloor \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right\rfloor \quad (2)$$

$$C_n = \max_i c_i - c_n \quad (3)$$

Where C_n is the cost of n^{th} imperialist (C_n is its normalized value).

3.2.1.2. Movement of Each Colony

Imperial competition and the policy of assimilation make the colony move towards the optimal state.

$$x = x + \beta d \cdot \gamma U(-\theta, \theta) \quad (4)$$

That, x is a stochastic variable with uniform distribution, and β is a value that is more than 1 in most of the implementations, d is the spacing between the colony and the imperialist. Assimilation and revolution policy constitute the core of this algorithm. This movement is shown in Figure 1. During this movement, if the colony gets been a higher value than its imperialist does, they will exchange their positions.

After that, the algorithm will continue with the new imperialist. The power of each empire is based on the function of imperial costs and colonies. Competition among empires is the more critical part of ICA (it based on empires power) [2].

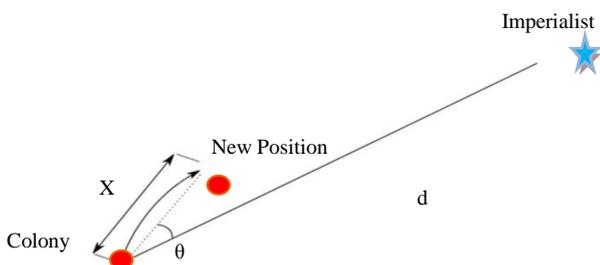


Figure 1. Colony to imperialist movement [2].

3.2.1.3. Total Cost of Empire

The policy of assimilation and imperial competition leads to moving a country towards the optimal point. The performance of an empire based on the power of the imperial country plus a percentage of the total colonies is calculated:

$$TC_n = \text{Cost}(\text{Imperialist}_n) + \varepsilon * \text{mean} \{ \text{Cost}(\text{Colonies of Empire}_n) \} \quad (5)$$

The ε is considered positive and less than 1. The imperialist competition will make strong nations increase their power and weak nations gradually be destroyed. Finally, an empire will remain, and the other countries will be colonized. ICA algorithm will terminate when only one empire remains in the population (this is the optimum point) [1, 2, 3, 10, 26].

3.2.2. Particle Swarm Optimization

PSO is an optimization technique based on the probability. This algorithm can solve complex problems and numerical functions. This algorithm based on group motion like birds. PSO is initialized with a population of particles randomly. By updating generations, searches are made to find optimal solutions. In each generation, the position and velocity of the i^{th} particle are updated with the personal best and global best (as P_{best} and G_{best}) position in the population. Ultimately, the best global position is considered as the ultimate solution. It should be noted that the position and speed are updated using Equations (6).

$$\begin{aligned} v_{id}(t+1) &= v_{id}(t) + c_1 * r_1 * (P_{id} - x_{id}(t)) + c_2 * r_2 * (P_{id} - x_{id}(t)) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \quad (6)$$

That r_1 and r_2 are random numbers between (1 and 0) and c_1 and c_2 are acceleration constant [4, 5, 15, 21].

4. Proposed Algorithm

In the proposed algorithm such as the standard ICA, countries that are of higher value were chosen as imperialists and other weaker countries are considered as colonies. Other countries are assigned as colonies in each imperialist. The strongest imperialist in the collection is chosen as emperor. After the division of countries, particle memory is also created (to update the velocity and position of the colonies and imperialists). In the next step, each imperialist chooses the weak countries randomly as colonies. The colonial movement is based on Equation (7):

$$x = x + \beta d \cdot \gamma U(-\theta, \theta) \quad (7)$$

If $\beta > 1$ colony moves towards the imperialist. d represents interval among the colony and the imperialist, and γ represent the coefficient of assimilation and $\gamma < 1$ its value. θ is the random change in responses and increases the chances of achieving

optimal response. In the next step, with the addition of the PSO operators, the new position of the independent states is obtained. In this way, by changing the particle swarm optimization algorithm, assuming that there are three types of best positions in multi-group space, the speed of convergence and new convergence is controlled. Also, a group with a large number of components is converted into smaller groups with fewer members. Searching is conducted independently as P_g in groups dependent. In this approach, the cosine similarity of the neighboring countries is measured by the nearest colonies of an imperialist and closest competitor country. The cosine similarity value is between -1 and 1 and is based on the following formula:

$$C_{i,x(country)} = \frac{\langle y - x, \delta_i(x) - x \rangle}{\|y - x\|_2 \|\delta_i(x) - x\|_2} \quad (8)$$

Where x is a sample of the country, also, δ the closest colonies in the rival empire (external). y is the colonies are in an empire (internal), And θ is the angle between x , and y . It is clear that the closer the cosine is, it is a sign of the desirability of a country and determines its direction. This process is shown in Figure 2.

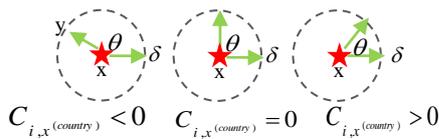


Figure 2. Cosine similarity for near country.

Thus, the search space with higher exploration rate is explored and the possibility to find better solutions is increased. According to [3], in the assimilation phase of the proposed combined algorithm, movement of countries is considered based on following parameters:

- The best personal experience of the related (country) object (P_L)
- The best P_L value in the autonomous object (P_g)
- The best group position of the object (P_m)
- The velocity of the current object (V_i^n)

Countries individually move in search space based on the following equations:

$$\begin{aligned} V_{i+1}^n &= w * V_i^n + c_1 * r_1 * (P_L - obj_i^n) + c_2 * r_2 * (P_g - obj_i^n) * c_3 \\ &* r_3 * (P_m - obj_i^n) \\ obj_{i+1}^n &= obj_i^n + V_i^n \end{aligned} \quad (9)$$

Where w inertia weight is applied to best group position (P_m). c_1, c_2 is a constant acceleration of motion and r_1, r_2 random number is between 0 and 1 and obj_i^n indicate the current status of the independent country. In summary, an empire exists with several colonies. Firstly, the modified assimilation operator is applied to colonies of an empire. Parameters of this phase include updating the velocity and the position of each country

is based on particle swarm optimization algorithm. After adding the modified operator, the process of operators of the imperialist competition algorithm continues. The moves are made in three groups, the movement of each country, the movement of the imperialists and the movement of the empires. For each of them, velocity and position are measured and directed toward the best. Figure 3 shows how the weak country moved towards the imperialist.

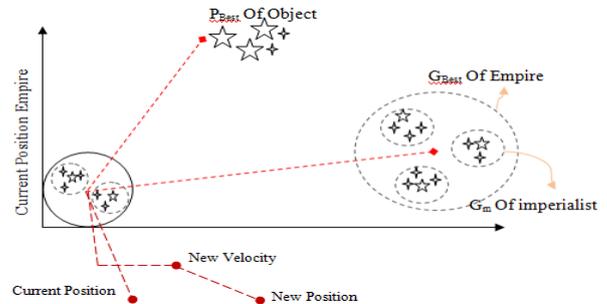


Figure 3. Moving the weak country toward a strong imperialist.

The permanence of the empire depends on the strong and the number of dominated colonies. As a result, during the country's internal revolution, imperialists will be forced to increase their own ability to provide the conditions for the development of their dominated colonies. If a colony is better than it's imperialist, after exchange it will be itself imperialist in the empire. As a result, during the imperialist revolution, the authority of the empires will gradually be the enhancement, and the weak empires will be eliminated. This process until continues only an absolute empire remains in the collection. The flowchart of the combined algorithm is shown in Figure 4.

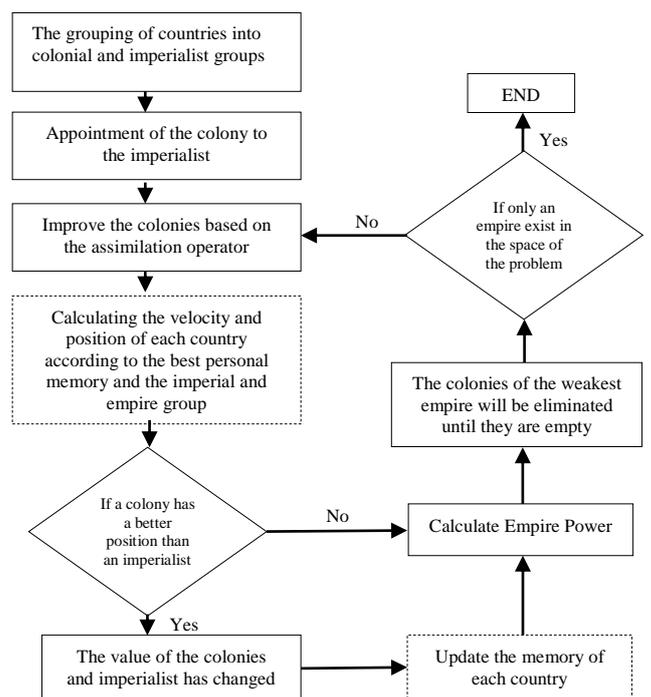


Figure 4. The hybrid method flowchart.

The power of each empire depends on both its constituent parts, the empire as the central core and its colonies. This dependence is calculated by describing the strength of the empire as the total power of the empire, plus a percentage of the average power of its colonies. To calculate the total cost of an empire is used the Equation (10):

$$TC_n = C(imp) + \xi \cdot \frac{\sum_{i=1}^{Ncol_n} C(col_i)}{Ncol_n} \quad (10)$$

Where ξ is a parameter whose value is less than 1. N_{col_n} represents the number of colonies of N empire, C_{imp} denotes imperialist cost, Col_i represents a colony of the empire. As stated above, with the formation of colonies and empires, imperialist competition begins between them; this competition takes place within the revolution phase. Ultimately, if there is no colony left for the empire, there are two conditions for the empire, the removal or transfer of the colony to another empire.

5. Experimental Setup

In this phase, several benchmark functions are applied to the proposed algorithm. The criteria for this phase are standard deviation, average, and medium. In this research, simulation and implementation using MATLAB software have been performed on a system with CPU=CORE i5, 2.50 GHz and RAM=6GB. After applying the proposed algorithm and calculating the parameters they are compared to eight other algorithms for evaluation of the results. Initially, to evaluate the performance of the proposed algorithm, the results are compared with ICA [2], PSO [15], GA+PSO, MVO [19], DE [22], ALO [18]. The proposed hybrid algorithm is based on the combination of operators of imperialist competition algorithms and particle swarm optimization. In all algorithms, answers are named by the type of algorithm. The problem can consider as a chromosome in the genetic algorithm, particle in particle swarm optimization algorithm and country in ICA, However, all refer to an answer. Each of these algorithms has different parameters; therefore, in this section, various tests have been performed to determine the range and values of these settings. Parameters of the proposed algorithm and the same parameters for other algorithms are shown in Table 2. Evaluation is based on benchmark functions. The benchmark functions used in the experiments are shown in Table 3. Benchmark functions are standard measurements in optimization. The benchmark functions are based on mathematical functions that can use as target functions for measuring and testing the efficiency of optimization algorithms. Each function has different properties. For example, the Ackley function is widely used for Multimodal functions. The Sphere function has only one global and local optimal point, while Griewank function includes large number of the local and global minimum optimal points. The

methods for solving the optimization problems depend on the features of the function f (x). In general, there are a large number of benchmark functions for global optimization [11, 27]. In this paper, assessments have made on the number of 17 benchmark functions.

Table 2. Settings parameters.

parameter	Description	value
N_{emp}	The Number Of Empire States	10
Revolution Rate	Revolution Rate (Similar To Mutation)	0.05
$C_{1,2,3}$	Acceleration Factor	0.5
$P_{Revolution}$	The Probability Of Revolution	0.1
ξ	For The Average Cost Of Colonies (Colonies)	0.1
β	To Balance Colonies	2
α	To Determine The Accuracy Of The Selection	1
w	Inertial Weight	1
N_{var}	Number of variables	20
Pop	Population	100
Iteration	Iteration	1000
Range	Range of variables	[-10 10]

The evaluation results are presented in Table 4. These results are achieved in more than 800 simulations of functions on algorithms (taking into account standard deviation, average and average results).

According to Table 4 of the results are obtained in BEALE, the combined algorithm, GA, PSO have the best average value. After that, ICA equal to 3.94E-30 has the lowest and MVO equal to 0.0075 has the worst average value of the minimization. In BOH3, the proposed algorithm, PSO, GA+PSO, ALO have the best value. After that, MVO equal to 0.0070 has the worst value. In COLVILLE, the ALO algorithm equal to 7.38E-08 has the best value. Also, other algorithms have almost the same average value. In GRIEWANK, the proposed algorithm equal to 0.50 has the worst value and the best value for the GA+PSO algorithm equal to 0.0011. In the HART, the ICA algorithm equal to -3.86 has been the best value, and then the combined algorithms and the DE have approximately the same value. In HUMP Benchmark functions, the proposed algorithm, DE, ICA, PSO and GA+PSO have reached the same value. After that, the ALO equal to 2.16E-02 has the best value and the MVO equal to 0.4 is the worst value. In LEVY, the ICA algorithm equal to 2.47 has the worst value. Then the algorithm GA + PSO and hybrid algorithm equal to 4.30E-06 have the best value. In MICH, the DE algorithm equal to 4.00E-01 has the worst value and ICA equal to -1.98 has the best value. After that, PSO, GA + PSO and hybrid algorithm have the best values. In PERM, the ALO algorithm equal to 3.80E-04 is the best value. The worst value is the DE algorithm equal to 1.86E-01 and then the hybrid algorithm equal to 4.25E-02. In POWELL, ALO algorithm equal to 2.81E-04 is the best value. And then, the GA+PSO, PSO, MVO algorithms and the hybrid algorithm have the worst value. In ROSEN, the GA+PSO algorithm and then the combined algorithm equal to 8.49E-19 have the best

value. It also the worst amount is for MVO algorithm equal to 0072. In SCHW, the ICA, PSO, GA+PSO algorithms have close values. It also, the hybrid algorithm and, ultimately, the algorithm DE have the worst values. In SHEKEL, the DE equal to 1.03E+01 algorithm has the worst value. The PSO, ICA algorithms have the same values and then the hybrid algorithm has a better value. In SPHEREF, the ALO algorithm equal to 4.66E-09 and, the ICA algorithm equal to 5.90E-04 have the best value respectively.

Also, the DE algorithm has the worst value. In ZAKH, the best value for the ICA algorithm equal to 1.97E-107 and then the hybrid algorithm equal to 8.21E-68. MVO algorithm equal to 0.010 has the worst value. In SUMSQUARES, ALO equal to 3.97E-01 and then the hybrid equal to 3.64E+02 have the best value and DE algorithm has the worst value. In ACKLEY, MVO and hybrid algorithm have the best value and ICA algorithm equal to 6.52E+03 has the worst value.

Table 3. Benchmark functions.

Functions	Mathematical Representation	Search Domain
Ackley	$f(x) = -\exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 - e$	$-15 \leq x_i \leq 30, i=1,2,\dots,n$
Beale	$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	$-4.5 \leq x_i \leq 4.5, i=1,2$
Bohachevsky3	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$	$-100 \leq x_i \leq 100, i=1,2$
Colville	$f(x) = 100(x_1^2 - x_1)^2 + (x_1 - 1)^2 + (x_3 - x)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$	$-10 \leq x_i \leq 10, i=1,2,\dots,n$
Griewank	$f(x) = 1 + \sum_{i=1}^n (\frac{x_i^2}{4000}) - \prod_{i=1}^n (\cos(\frac{x_i}{\sqrt{i}}))$	$-600 \leq x_i \leq 600, i=1,2$
Hartmann	$f(x) = -\sum_{i=1}^4 \alpha_i \exp(-\sum_{j=1}^3 A_{ij} (x_j - P_{ij})^2)$	$0 \leq x_i \leq 1, i=1,2,3$
Hump	$f(x) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1 x_2 + (-4 + 4x_2^2)4x_2^2$	$-5 \leq x_i \leq 5, i=1,2$
Levy	$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_d + 1)] + (w_d - 1)^2 \dots [1 + \sin^2(2\pi w_d)]$, $w_i = 1 + \frac{x_i - 1}{4}$, for all $i = 1, \dots, 4$	$-10 \leq x_i \leq 10, i=1,2,\dots,n$
Michalewics	$f(x) = -\sum_{i=1}^d \sin(x_i) \sin^{2m}(\frac{ix_i^2}{\pi})$	$0 \leq x_i \leq \pi, i=1,2,\dots,n$
perm	$f(x) = \sum_{i=1}^d (\sum_{j=1}^d (j^i + \beta) (\frac{x_j}{j})^i - 1)^2$	$-n \leq x_i \leq n, i=1,2,\dots,n$
Powell sum	$f(x) = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} + 10x_{4i})^2 \dots + (x_{4i-2} + 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	$-4 \leq x_i \leq 5, i=1,2,\dots,n$
Rosenbrock	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-5 \leq x_i \leq 10, i=1,2,\dots,n$
Schwefel	$f(x) = f(x_1, x_1, \dots, x_1) = 418.9829d - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	$-500 \leq x_i \leq 500, i=1,2,\dots,n$
shekel	$f(x) = \sum_{i=1}^m [c_i + \sum_{j=1}^n (x_j - a_{ji})^2]^{-1}$	$0 \leq x_i \leq 10, i=1,2,3,4$
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$5.12 \leq x_i \leq 5.12, i=1,2,\dots,n$
SumSquares	$f(x) = \sum_{i=1}^n i^2 x_i^2$	$-10 \leq x_i \leq 10, i=1,2,\dots,n$
zakharov	$f(x) = \sum_{i=1}^n x_i^2 + (\frac{1}{2} \sum_{i=1}^n i x_i)^2 + (\frac{1}{2} \sum_{i=1}^n i x_i)^4$	$-5 \leq x_i \leq 5, i=1,2,\dots,n$

6. Conclusions

In this paper, a new method was introduced, which was created by combining two evolutionary algorithms. These two algorithms are particle swarm optimization and imperialist competitive algorithm. One of the main problems with algorithms is to remain at the optimal local point. During the search process, the algorithm may remain into local optimal. The combination two

algorithms PSO and ICA have created a balance between exploration and exploitation processes By measuring the cosine similarity between each country In other words, updating velocity and position of the countries in each movement has been accelerated the operation (so that the countries could achieve better positions). The proposed combined algorithm was tested with several benchmark functions. According to the results, the proposed algorithm is able to perform

well in most benchmark functions. A Comparison of convergence of other optimization algorithm showed that the proposed combined algorithm has found the optimal solution with higher performance. The combined algorithm can be used to solve other optimization problems in engineering science in future works; in addition, other evolutionary algorithms can

be considered in the combination process. Also As future research, the similarity distance of neighbouring countries in the algorithm can be considered using the method introduced in [28] based on probability distribution measurements.

Table 4. Performance evaluation benchmark function.

		Proposed	DE	ICA	PSO	GA+PSO	MVO	ALO
BEALE	Std	0	6.8E-10	5.31E-30	2.78E-21	0	0.00348	1.22E-08
	Avg	0	1.11E-09	3.94E-30	1.53E-21	0	0.00758343	2.16E-08
	Med	0	1.53E-09	3.94E-31	2.37E-22	0	0.008833	3.04E-08
BOH3	Std	0	1.229E-11	8.33E-17	0	0	0.002260845	0
	Avg	0	1.24E-11	4.16E-17	0	0	0.00704199	0
	Med	0	6.50E-12	0.00E+00	0	0	0.007417236	0
COLVILLE	Std	0.0484371	0.0433757	0.978963	0.018195	2.44E-09	0.001649599	8.769E-08
	Avg	0.688363	0.768396	1.474139	0.674183	0.666667	0.00984594	7.38E-08
	Med	-1	-1	-1	-1	-1	0.005765	2.90E-08
GRIEWANK	Std	1.34102370	0.040841	0.0138117	0.004825	0.000761	0.16250782	0.07120662
	Avg	0.501171	0.492092	0.13514692	0.017955	0.001199	0.215823	0.211108
	Med	0.00140762	0.495748	0.138379	0.017964	0.001049	0.334502259	0.26076880
HART3	Std	0	0	0	0	0	0.001932047	3.03113E-09
	Avg	3.86278215	3.862782	-3.86278	-3.86278	-3.86278	0.007234	5.12E-09
	Med	3.86278214	3.862782	-3.862782	-3.86278	-3.86278	0.007747507	4.68E-09
HUMP	Std	0.00E+00	4.33E-16	9.93E-17	0	1.22E-16	0.054901029	0.02928645
	Avg	4.65E-08	4.65E-08	4.65E-08	4.65E-08	4.65E-08	0.049782	2.16E-02
	Med	4.65E-08	4.65E-08	4.65E-08	4.65E-08	4.65E-08	0.024213877	4.34E-04
LEVY	Std	4.68E-06	1.01E-05	3.8813767	0.692537	2.09E-09	0.021697201	0.00012419
	Avg	4.30E-06	3.08E-05	2.474617	3.64E-01	1.85E-09	0.021945	1.15E-04
	Med	2.04E-06	2.40E-05	0.4546673	4.67E-05	1.12E-09	0.024213877	5.38E-05
MICH	Std	8.74E-01	-2.00E+00	2.48E-16	0.098902	0.091222	0.001644	1.24E-09
	Avg	-1.56E+00	4.00E-01	-1.98795	-1.9E+00	-1.90E+00	0.005589	3.78E-09
	Med	-1.97E+00	2.00E+00	-1.987951413	-1.9E+00	-1.97E+00	0.00952	3.65E-09
PERM	Std	3.66E-02	0.154318	0.153729	0.002596	0.01578	0.021617	0.000348
	Avg	4.25E-02	1.86E-01	1.38E-01	5.02E-03	2.07E-02	0.037032	3.80E-04
	Med	2.18E-02	1.62E-01	1.14E-01	5.48E-03	2.39E-02	0.026459	5.44E-04
POWELL	Std	1.07E-01	103.3572	1.960053	0.00153	7.5E-05	0.010592	0.000563
	Avg	1.56E-01	3.23E+02	1.949295	5.51E-03	3.03E-04	0.028909	2.81E-04
	Med	1.35E-01	3.15E+02	0.795681534	5.37E-03	3.22E-04	0.027597	5.07E-05
ROSEN	Std	1.17E-18	0.001415	0.003573	1.94E-13	0	0.004502	1.79E-10
	Avg	8.49E-19	1.07E-03	1.72E-03	1.02E-13	0.00E+00	0.007277	3.85E-09
	Med	3.48E-21	5.78E-04	1.02E-04	2.08E-14	0.00E+00	0.004971	3.77E-09
SCHW	Std	1.96E+01	14822.16	172.9397	7.511201	0.569324	0.158747	1.39E-09
	Avg	-6.32E+00	1.03E+01	-9.26E+00	-9.4E+00	-9.45E+00	0.040321	4.36E-03
	Med	4.11E+02	-1.07E+04	4.10E+02	4.52E+02	4.48E+02	0.009038	5.21E-09
SHEKEL	Std	1.96E+01	14822.16	172.9397	7.511201	0.569324	0.158747	1.39E-09
	Avg	-6.32E+00	1.03E+01	-9.26E+00	-9.4E+00	-9.45E+00	0.040321	4.36E-03
	Med	-1.05E+01	1.03E+01	-1.05E+01	-1.0E+01	-1.05E+01	0.04358	1.42E-03
SPHERE	Std	1.31E-02	1.581217	0.00034781	0.430643	0.0153778	0.008525	2.14E-09
	Avg	7.05E-02	3.75E+01	5.90E-04	1.11E+00	3.22E-02	0.008449	4.66E-09
	Med	6.65E-02	36.77758	4.98E-04	1.01E+00	3.27E-02	0.005785	7.84E-05
ZAKH	Std	1.84E-67	8.09E-13	4.4E-107	3.14E-23	8.29E-47	0.004801572	2.34E-09
	Avg	8.21E-68	6.14E-13	1.97E-107	3.58E-23	6.91E-47	0.010174	6.29E-09
	Med	1.97E-83	4.02E-13	8.86E-110	2.35E-23	2.36E-47	0.0084	7.17E-09
SUMSQUARES	Std	6.86E+01	59705.85	5315.622273	542.8685	32.269098	0.019181	0.823122
	Avg	3.64E+02	1.02E+06	2.59E+04	2.68E+03	6.68E+01	0.04393	3.97E-01
	Med	3.73E+02	1.02E+06	2.55E+04	2.61E+03	8.00E+01	0.043885	1.15E-04
ACKLEY	Std	7.26E-03	0.149979	14576.13309	0.402004	0.5013241	0.006667	1.026621
	Avg	8.87E-02	9.12E+00	6.52E+03	1.76E+00	6.19E-01	0.039902	1.54E+00
	Med	8.54E-02	9.08E+00	5.64E+00	1.99E+00	6.33E-01	0.037805	1.65E+00

References

[1] Abdeyazdan M., "Data Clustering Based on Hybrid Kharmonic Means and Modifier Imperialist Competitive Algorithm," *The Journal of Supercomputing*, vol. 68, no. 2, pp. 574-598, 2014.

[2] Atashpaz-Gargari E. and Lucas C., "Imperialist Competitive Algorithm: An Algorithm for

Optimization Inspired by Imperialistic Competition," in *Proceedings of IEEE Congress on Evolutionary Computation*, Singapore, pp. 4661-4667, 2007.

[3] Ban M. and Jalal S., "A Novel Hybrid Imperialist Competitive Algorithm for Global Optimization," *Australian Journal of Basic and Applied Sciences*, vol. 7, pp 330-341, 2013.

[4] Bansal J., Singh P., Saraswat M., Verma A.,

- Jadon S., and Abraham A., "Inertia Weight Strategies in Particle Swarm Optimization," in *Proceedings of 3rd World Congress on Nature and Biologically Inspired Computing*, Salamanca, pp. 633-640, 2011.
- [5] Branke J., *Evolutionary Optimization in Dynamic Environment*, Kluwer Academic Publishers, 2001.
- [6] Dorigo M. and Stützle T., *Handbook of Metaheuristics*, Springer Link, 2003.
- [7] Dorigo M., Birattari M., and Stutzle T., "Ant Colony Optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, 2006.
- [8] Goldberg D., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Reading, 1989.
- [9] Hoos H. and Stützle T., *Stochastic Local Search: Foundations and Applications*, Elsevier, 2004.
- [10] Idoumghar L., Chérina N., Siarry P., Roche R., and Miraouic A., "Hybrid ICA-PSO Algorithm for Continuous Optimization," *Applied Mathematics, and Computation*, vol. 129, no. 24, pp. 11149-11170, 2013.
- [11] Jamil M. and Yang X., "A Literature Survey of Benchmark Functions for Global Optimization Problems," *Journal of Mathematical Modeling and Numerical Optimization*, vol. 4, no. 2, pp. 150-194, 2013.
- [12] Jin Y. and Branke J., "Evolutionary Optimization in Uncertain Environments A Survey," *IEEE Transactions Evolutionary Computation*, vol. 9, no. 3, pp. 303-317, 2005.
- [13] Johnson D., and Papadimitriou C., and Yannakakis M., "How Easy is Local Search?," *Journal of Computer and System Sciences*, vol. 37, no. 79-100, 1988.
- [14] Karaboga D. and Basturk B., "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm," *Journal of Global Optimization*, vol. 39, pp. 459-471, 2007.
- [15] Kennedy J. and Eberhart R., "Particle Swarm Optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, 1995.
- [16] Kirkpatrick S., "Optimization by Simulated Annealing: Quantitative Studies," *Journal of Statistical Physics*, vol. 34, pp. 975-986, 1984.
- [17] Li C. and Yang S., "A Clustering Particle Swarm Optimizer for Dynamic Optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, Trondheim, pp. 439-446, 2009.
- [18] Mirjalili S., "The Ant Lion Optimizer," *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015.
- [19] Mirjalili S., Mirjalili S., and Hatamlou A., "Multi-Verse Optimizer: A Nature-Inspired Algorithm for Global Optimization," *Neural Computing and Applications*, vol. 27, pp. 495-513, 2016.
- [20] Mitchell M., Holland J., and Forrest S., "When Will A Genetic Algorithm Outperform Hill Climbing?" *Advances in Neural Information Processing Systems*, pp. 51-58, 1993.
- [21] Pan X., Xue L., Lu Y., and Sun N., "Hybrid Particle Swarm Optimization with Simulated Annealing, Multimedia Tools Application," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 29921-29936, 2019.
- [22] Rocca P., Oliveri G., and Massa A., "Differential Evolution as Applied to Electromagnetic," *IEEE Antennas and Propagation Magazine*, vol. 53, no. 1, pp. 38-49, 2011.
- [23] Storn R. and Price K., "Differential Evolution-A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," *Journal Global Optim*, vol. 11, pp. 341-59, 1997.
- [24] Van-den-Bergh F. and Engelbrecht A., "A New Locally Convergent Particle Swarm Optimizer," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Yasmine Hammamet, 2002.
- [25] Wolpert D. and Macready W., "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, 1997.
- [26] Yin Z. and Gao Q., "A Novel Imperialist Competitive Algorithm for Scheme Configuration Rules Mining of Product Service System," *Arabian Journal for Science and Engineering*, vol. 45, pp. 3157-3169, 2020.
- [27] Zhang Y., Hu X., and Wu C., "Improved Imperialist Competitive Algorithms for Rebalancing Multi-Objective Two-Sided Assembly Lines with Space and Resource Constraints," *International Journal of Production*, vol. 58, no. 12, pp. 3589-3617, 2019.
- [28] Zheng L., Zou J., Liu B., Hu Y., and Deng Y., "A Novel Evidence Distance in Power Set Space," *The International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 8-15, 2020.



Maryam Houtinezhad completed her B.Sc. degree in computer engineering at Azad Islamic University, her M.Sc. degree at the University of Science and Research, and her Ph.D. at the Islamic Azad University, Ferdows Branch. Her interest research areas are image processing, pattern recognition and the application of optimization and web mining algorithms.



Hamid Reza Ghaffary completed his B.Sc. degree in computer science at Sharif University of Technology and his M.Sc. degree at the University of South Tehran and his Ph.D. at Ferdowsi University. He is currently a faculty member and assistant professor at the Faculty of Computer Engineering, Ferdows Azad University. His interest research areas are machine learning, pattern recognition and image processing.