

# An Improved Imperialist Competitive Algorithm Based on a New Assimilation Strategy

Seyed Mojtaba Saif<sup>1</sup>

Received (2015-08-05)

Accepted (2016-03-16)

---

**Abstract** — Meta-heuristic algorithms inspired by the natural processes are part of the optimization algorithms that they have been considered in recent years, such as genetic algorithm, particle swarm optimization, ant colony optimization, firefly algorithm. Recently, a new kind of evolutionary algorithm has been proposed that it is inspired by the human evolution process. This new algorithm has been called Imperialist Competitive Algorithm (ICA). The ICA is a population-based algorithm where the population is represented by countries that are classified as colonies or imperialists. This paper is going to present a modified ICA with high accuracy, referred to here as ICA2. The ICA2 is tested with six well-known benchmark functions. Results show high accuracy and avoidance of local optimum traps to reach the minimum global optimal.

Two main operators of the ICA are Assimilation and Revolution. Assimilation policy is the more important than Revolution. This research focuses on an assimilation policy in the ICA to propose a meta-heuristic optimization algorithm for optimizing function with high accuracy and avoiding to trap in local optima rather than using original ICA.

**Index Terms** — evolutionary algorithm, optimization algorithm, imperialist competitive algorithm, assimilation policy.

---

1- Department of Computer Engineering, Islamic Azad university, Safashahr Branch, Safashahr, IRAN (MojtabaSaif@Gmail.com, Mojtaba.Saif@IAUSafashahr.ir)

## I. INTRODUCTION

FIND the optimal solution is a major challenge in many scientific problems. The heuristic algorithms have widely been used to reach global optimum of different problems. Genetic Algorithm (GA) [2], Particle Swarm Optimization (PSO) [3], Ant Colony Optimization (ACO) [4], Differential Evolution (DE) [5], Gravitation Algorithm (GA) [6] and Firefly Algorithm [7] are some familiar meta-heuristic studies. Recently, inspired by a socio-politically motivated, a meta-heuristic algorithm called Imperialist competitive algorithm (ICA) is proposed by Atashpaz and Lucas [8]. The ICA is a multi-agent algorithm with each agent being a country [9]. In ICA, the countries divided into two groups based on their power, colonies and imperialists [8]. But these algorithms have defects to deal with the local optimum trap and the accuracy rate [1]. The local optimum trap is an important shortcoming in optimization. A local optimum is a solution that is optimal within a neighboring set of solutions. This is in contrast to a global optimum, which is the optimal solution among all possible solutions that may occur under different situations [10]. ICA is much more successful than other optimization methods to eradicate these problems [8, 9, and 10]. It has been used to solve different kinds of problems, such as PID controller design [11], initial and boundary-value problems [12], a new hybrid data-clustering algorithm [13, 14], economic problems [15, 16, 17] and many engineering problems [1, 9, 18, 19, 20].

The motivation of this study is to present a meta-heuristic algorithm for optimizing with high accuracy and avoiding to trap in local optima, based on the ICA, because of, ICA has successful application in various fields and good

performance in comparison to other optimization methods [11]. Various attempts have been made to improve the ICA, include modifications in ICA [10,24,25,26,27] and a hybrid with local search or other optimization techniques [28,29,30,31,32,33].

This study focuses on the assimilation policy because the movement of colonies toward their imperialist has the biggest influence of them all [24]. ICA2 with a new adaptive assimilation strategy avoids from trap of local optimum.

Organization of the paper is in order. Section 2 provides a precise description of the basic ICA and reviews the developments on the ICA. Section 3 presents the proposed algorithm. Experimental setup and results are discussed in Section 4. Section 5 includes summary and conclusion.

## II. IMPERIALIST COMPETITIVE ALGORITHM

ICA simulates the social political process of imperialism and imperialistic competition. This algorithm contains a population of agents or countries. The Steps of algorithm are described as below [8].

### 1. Initializing

ICA starts with an initial population of randomly generated solutions  $\{p_1, p_2, \dots, p_N\}$ , where each solution  $p_i$  is called country and is a  $1 \times n$  array of variables to be optimized, and  $N$  denotes the number of countries. ICA computes the cost of each  $p_i$  as  $C_i$ . Then sort ascending the initial population based on cost values. The most powerful countries are selected as imperialists (equation 1), and the others are divided among imperialists, based on the power of each imperialist (equation 2). That is,

$$POW_i = \left( \max_{1 \leq j \leq N_{imp}} \{C_j\} \right) - C_i \quad (1)$$

$$NC_i = \text{round} \left( \left| \frac{POW_i}{\sum_{j=1}^{N_{imp}} POW_j} \right| \times [N - N_{imp}] \right) \quad (2)$$

Where  $C_i$  is the cost of  $i$ th imperialist and  $POW_i$  is the power of  $i$ th imperialist. Also,  $NC_i$  denotes the number of colonies assigned to empire  $i$ ,

where  $1 \leq i \leq N_{imp}$  and  $N_{imp}$  is a constant value.

### 2. Moving

Two main operators of this algorithm are Assimilation and Revolution. In this step, the algorithm assimilates a solution with low-quality or a country with low power to a country with high power or the algorithm assimilates colonies toward relevant imperialists. Each country is assumed to be a point in the search space. Assimilation within an empire is achieved by moving all colonies toward their imperialist by  $\delta$  units (equation 3). The distance of such an assimilation move is a random variable with uniform distribution [8]:

$$\delta \sim U(0, \beta \times d) \quad (3)$$

Here, the  $d$  is the Euclidean distance between imperialist and colony, and  $\beta$  is a parameter of ICA called assimilation coefficient. The value of  $\beta$  is in the half-closed interval  $(1, 2]$ , and 2 is recommended. In addition to assimilation, the revolution was used to change the position of the colonies. Revolution is sudden changes in the position of some of the countries. According to this strategy, a random amount of deviation is in the direction of movement. The direction of the assimilation move is the direction from imperialist to colony plus a random variable  $\theta \sim U(-\gamma, \gamma)$ , where  $\gamma$  is a parameter of ICA, called revolution coefficient. The suggested value for  $\gamma$  is  $\pi/4$  [22].

After moving a colony to a new position, its cost is recomputed. If the cost of the colony is smaller than the cost of the imperialist, exchange colonies and their imperialist. A colony may reach a position with lower cost than that of the imperialist while moving toward the imperialist.

### 3. Imperialistic competition

Next, the cost (denoted by  $Q_i$ ) and the power (denoted by  $E_i$ ) of each empire are calculated. It is influenced by the power of imperialist country (denoted by  $c_i$ ) and the colonies of an empire (See equations 4 and 5) [8].

$$Q_i = c_i + \xi \text{ mean } \{ \text{cost}(\text{colonies of empire } i) \} \quad (4)$$

$$E_i = \left( \max_{1 \leq j \leq N_{imp}} \{Q_j\} \right) - Q_i \quad (5)$$

Where,  $\xi$  is a parameter whose value is in the open interval  $(0, 1)$ . Suggest value for  $\xi$  is 0.1.

Competition among all empires is achieved by taking the weakest colony away from the weakest empire and giving it to a chosen empire, where the probability of empire being chosen is

$$\left| \frac{E_i}{\max_{1 \leq j \leq N_{imp}} E_j} \right| \quad [8].$$

When an empire loses all its colonies, it is assumed to be collapsed and will be eliminated. At the end, the colonies will be under the control of the most powerful empire.

The steps of implementing ICA are shown in fig1 [8].

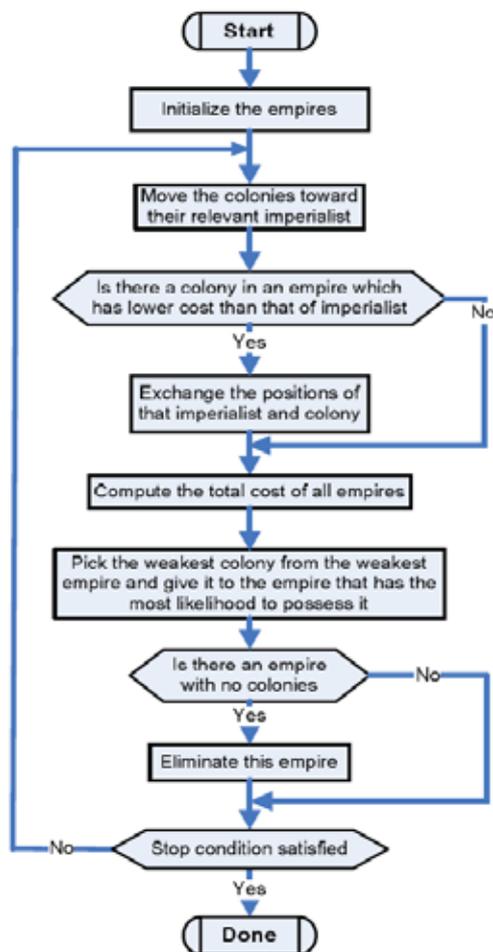


Figure 1 : Imperialist Competitive Algorithm [8]

### III. REVIEWS THE DEVELOPMENTS ON THE ALGORITHMIC RESEARCH WITH ICAS

In ICA, two operators are called assimilation and revolution, and one strategy called imperialistic competition are the main building blocks [23], which are responsible to lead the search for a better solution. The movement of

colonies toward their imperialist has the biggest influence of them all, since it occurs for all the colonies during all the iterations, whereas the revolution happens with a low probability and the imperialistic competition affects only one colony per iteration [24]. For this reason, researchers have focused on the improving the assimilation policy for improve ICA. There are several studies for improve ICA by modify assimilation [10, 24, 25, 26, 27]. Also, researchers have presented several hybrid approaches consisting ICA and other algorithms for improve the performance of ICA [28, 29, 30, 31].

Esmailzadeh [10] proposes a new assimilation strategy based on this fact, which the imperialists also need to model, and they move toward top imperialist state. In the modified ICA by Esmailzadeh, which referred to here as ESICA, the imperialists assimilate toward the best imperialist, and this movement affects all colony movements.

An improvement in the ICA by implementing an attraction and repulsion concept during the search for better solutions, the AR-ICA approach is proposed in [24]. A concept of attraction and repulsion is introduced to overcome the premature convergence problem. This problem is caused when the algorithm gets trapped in a local optimum. In this method the algorithm switches between the attraction and repulsion phase according to a threshold value for the population diversity. The diversity guided can be implemented in the ICA by varying the assimilation coefficient according to the distance between the colony and its imperialist [24]. If the movement equation is reformed by

$$Pos_{i+1} = Pos_i + A \cdot \delta \quad (7)$$

$$A = \gamma \cdot d \quad (8)$$

Where A determines the boundaries of the region, the next colony position can take place  $Pos_i$  is the vector of the colonies' position on the  $i$ th iteration,  $\delta$  is a random number normally distributed between [0, 1], and  $d$  is vector containing the variables distance between the colony and its imperialist. Then the diversity guided adapted for the ICA is given by the equation 10

$$\gamma = \begin{cases} 2 & d > d_{div} \\ \gamma_{div} & d \leq d_{div} \end{cases} \quad (10)$$

Here,  $d_{div}$  is the distance threshold value, and  $\gamma_{div}$  is the number greater than 2.

The normal distributed assimilation is another improvement in the ICA for to control convergence rate of the ICA, and to overcome being trapped in a local optima problem [25]. According to this strategy, the movement equation is

$$Pos_{i+1} = Pos_{imp,i} + \delta \quad (11)$$

Where  $Pos_{imp,i}$  is the vector of the imperialist's position on the  $i$ th iteration,  $\delta$  denotes the amount of movement chosen from a normal distribution between  $[0,d]$ . To divert the search to less explored regions, colonies are allowed to move in their self-neighbors instead of the imperialist's neighbor with a small probability [25]. It means that the position of a colony is updated as  $Pos_{i+1} = Pos_{imp,i} + \delta$ .

In addition, the normal distributed assimilation, adaptive assimilation is presented in [25]. In this strategy, an adaptive controller based on the progress history was used in the assimilation function. The movement vector should be increased if the success rate is high, and it should be decreased if the success rate is low. The success rate means that the proportion of colonies reached a better position [25]. The ICA with adaptive assimilation called here as *Adapt\_ICA*.

The chaotic sequences can be used in an assimilation function instead of random sequences [26].

The Different types of the crossover operations of genetic algorithm are used as an assimilation policy for the improvement in the ICA [27, 31].

#### IV. THE PROPOSED APPROACH

As mentioned in the previous section, there are three processes in the ICA, and the most important of them is the movement of colonies toward their imperialist. The ICA performance can be improved significantly by modification in the assimilation process.

ES\_ICA [10] proposes an assimilation strategy based on two assimilation types. The empire movements are toward the best imperialist, and the colonies movements toward their imperialist are assimilation types in the ES\_ICA. A pseudo-code for the ES\_ICA algorithm is shown in algorithm 1.

ICA2 can avoid the local optimum trap by

manipulations diversity. If the colony is near its imperialist (the diversity is low), the colony is repelled by making a better exploration of the search space; if the colony is far from, its imperialist (the diversity is high), the colonies are attracted to each other. Repellency and attraction colonies can be implemented by varying the assimilation coefficient. If the assimilation coefficient increases, a possibility for the colony to move away from the imperialist is created. By decreasing the assimilation coefficient, the chance the colony diverges from the imperialist will also decrease.

The probability of finding the optimal solution is higher in being close to the empire's positions. The empires are local optimal solutions. When the colonies are closer to its empire, the search must be done carefully. Moreover, the search should avoid the local optimum trap by preventing premature convergence through increased diversity [25].

ICA2 defines a new assimilation policy based on ES\_ICA assimilation strategy and AR\_ICA assimilation strategy. In the ICA2, the moving step has been modified and improved by the empire's movements toward the best imperialist and the colonies' movements toward their imperialist. ICA2 will control colonies diversity adaptively (See algorithm 2). In the ICA2, the following equations represent the new position of a colony:

$$X_{new1} = X_{old} + \beta \times d1 \quad (12)$$

$$X_{new2} = X_{new1} + \beta \times d2 \quad (13)$$

Where  $d1$  is the vector with the distances between the colony and the imperialist and  $d2$  is the distance vector between the colony and the best imperialist.  $X_{old}$  is old position of colony and  $X_{new2}$  is the colony final position. The  $\beta$  is a parameter for adjusting the momentum and modifying the area that colonies randomly search around the imperialist [8]. see algorithm 2 for more details:

**Algorithm 1** Modify Imperialist Competitive Algorithm [10]

```

1 The empire allocated
2 While you do not reach to a given number of repetitions in the problem do
3   For all imperialists except best imperialist do
4     If imperialists gain a better position than before, then
5       Imperialists assimilate to the best imperialist with a moderate tune;
6     end
7   end
8   For all empires do
9     For all colonies belong to empire do
10      Assimilate colonies toward relevant imperialist with a predefined degree;
11      Assimilate colonies toward top imperialist with a predefined degree;
12      Revolve colonies;
13    end
14    Update colonies;
15    For all colonies belong to empire do
16      If a colony gains a better position than its imperialist then
17        Exchange the imperialist and the best colony;
18    end
19  end
20 end
21 Imperialistic competition;
22 Display results;
23 end

```

In algorithm 2, The  $\gamma$  is a constant value and The  $\beta$  is equal 2 when the distance between the colonies to its empire is more than threshold, and if the distance between the colonies to its empire is less than threshold, to increase the diversity and prevent premature convergence, at 20 percent of the colonies the  $\beta$  value is increasing. The area that colonies randomly search around its imperialist expands if the  $\beta$  increases.

The ICA2 is presented if the algorithm 2 inserts at line 11 of the algorithm 1.

**Algorithm 2** Assimilate of colonies

```

11.1 if (the distance between the colonies to its empire is
more than threshold)
11.2    $\beta=2$ ;
11.3 else
11.4   Generate a pseudorandom scalar drawn from the
standard uniform distribution on the open interval (0, 1),
and assign it to P.
11.5   if ( $P>0.9$ )
11.6      $\beta=2*\gamma$ ;
11.7   else if ( $P>0.8$ )
11.8      $\beta=\gamma$ ;
11.9   else
11.10     $\beta=2$ ;
11.11   end
11.12 end
11.13 end

```

**V. EXPERIMENTS AND RESULTS**

Performance of ICA2 has been compared with the following ICA-variant on a set of 6 numerical benchmarks: the original ICA [8], the modified ICA by Esmailzadeh (ES\_ICA) [10], the AR-

ICA [24], and the ICA with adaptive assimilation (Adapt\_ICA) [25].

The parametric settings of the contestant algorithms are summarized in Table 1.

Table 1: The parameters considered for ICAs

Parameter	Value	Description
MaxIt	90,000	Maximum number of iterations
nEmp	4	Number of empire/imperialists
beta	2	Colony assimilation coefficient
pRevolution	0.1	Revolution probability
Zeta	0.1	Colonies mean cost-coefficient
Threshold	0.8	The distance threshold value is used in the AR-ICA and ICA2
$\gamma$	3	A const value is used in the AR-ICA and ICA2
Beta1	0.5	Empire assimilation coefficient is used in the ES_ICA

The proposed ICA has been tested on 6 benchmark functions [34, 35] which were also used for validating the well-known ICA variants such as AR\_ICA. All the benchmark functions (f1-f6) used in this study are minimization problems. f1– f4 are continuous unimodal functions, and f5-f6 are multimodal functions. The list of benchmark functions are summarized in the Table 2 that the global optimal, the global minimum value (fmin), and the search range are shown in two, three, and four columns, respectively, according to the definitions in [35].

All result tables report the average best cost of the objective function over 10 independent runs for each algorithm. The termination condition for the each run of an algorithm was to reach 90,000 iterations. The number of dimensions was 30 for all benchmark functions. The Wilcoxon rank sum test was used for comparisons of the

results obtained with the ICA2 algorithm and how they differed from the final results the other competitors in a statistically significant way. It is a nonparametric statistical test for two populations when samples are independent [35, 36]. The significance level was 5%.

Table 3 shows the average value of the results obtained with the performing algorithm achieved from 10 independent runs for each algorithm and p-values computed for all the pairwise comparisons concerning ICA2. To make the comparison fair enough, all the ICA-variants were made to run with the parameter summarized in Table 1. In Table 3, the P-values obtained from the Wilcoxon test indicate that the difference of the mean values obtained with ICA2 and other algorithms are statistically significant in most experiments. Convergence characteristics of ICA2 along with other peer ICA-variants are shown in Fig 2.

As evident from the Table 2, ICA2 version has outperformed most of the contender ICA algorithms on all of the functions. ES\_ICA appears to be the best in simple unimodal functions (f1, f2) and the multimodal function f5 where AR\_ICA offers exceptionally good results for functions f3, f4, and f6. But ICA2 has performed almost equally well in all of the functions. ICA2 has achieved second rank for all function. Other ICAs are good only for some functions.

Functions f1 to f4 are continuous unimodal where the ICA2 won two ranks among all ICAs. The f3 is the Rosenbrock functions which is unimodal for  $D = 2$  and 3, but with an increase in complexity of the problem, a problem may become multimodal in high dimensions [37].

The results evidently support that ICA2 can show significantly improved performances in multimodal functions f5-f6. High-quality mean solutions were obtained by ICA2 in the Schewefels function (f5). ICA2 offered superior performance among all ICAs except AR\_ICA for the Griewank function (f6).

In addition to the ICAs discussed earlier, ICA2 has also been compared with four other methods of optimization algorithms having varied origins. These algorithms include Fast Evolutionary Programming (FEP) with Cauchy mutation [34], Particle Swarm Optimizer with Difference Means based on Perturbation (DMP-PSO) [35], adaptive Differential Evolution (JADE) with optional external archive [37], and Improved Artificial

Bee Colony algorithm for global optimization [38]. ICA2 manages to yield very competitive results against some of these algorithms on most of the test cases reported. Table 4 lists the results. The results of JADE, DMP-POS, FEP, and IABC are all quoted directly from [34, 35, 36, 37, and 38] directly.

## VI. CONCLUSION

This paper proposes a new meta-heuristic optimization algorithm that is based on ICA. In this study, the goal is to present a meta-heuristic optimization algorithm with high accuracy and to avoid trapping in local optimal with modification of the assimilation policy of the ICA. The proposed assimilation provides the necessary trade-off between exploitation and exploration to detect the minimum global optima of unimodal and multimodal functional landscapes with considerable accuracy. The performance of the proposed approach is compared to various ICAs and other well-known meta-heuristics using benchmark functions. Experiments are conducted to confirm the ability of the proposed algorithm.

## ACKNOWLEDGMENT

This work was supported by the Safashahr Branch of Islamic Azad University Research Adjutancy.

**Table 2 Six benchmark functions used to test ICA2**

Type	Function	Search range	Global Opt.	f <sub>min</sub>	Name
Unimodal	$f_1 = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	$[0]^D$	0	Sphere
Unimodal	$f_2 = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	$[-10, 10]^D$	$[0]^D$	0	SchwefelsP2.22
Unimodal	$f_3 = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^D$	$[1]^D$	0	Rosenbrock
Unimodal	$f_4 = \sum_{i=1}^D i \cdot x_i^4 + rand[0,1)$	$[-1.28, 1.28]^D$	$[0]^D$	0	Noise
Multimodal	$f_5 = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^D$	$[0]^D$	0	Ackley
Multimodal	$f_6 = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	$[0]^D$	0	Griewank

**Table 3 Mean and standard deviation values obtained by ICA2 compared to those obtained with the other ICAs**

Function		ICA	AR_ICA	ES_ICA	Adapt ICA	ICA2
F1	Average	1.5795e-58	2.1196e-41	<b>5.3132e-76</b>	3.9496e-38	<b>2.3263e-67</b>
	STD	4.6334e-58	4.8206e-41	1.4366e-75	8.4141e-38	5.7736e-67
	P_Value	0.0211	1.7861e-04	1.8267e-04	1.4851e-04	NA
F2	Average	3.9026e-30	2.8439e-22	<b>5.059e-38</b>	1.0229e-20	<b>6.3006e-34</b>
	STD	4.1875e-30	2.1601e-22	6.0034e-38	1.6164e-20	8.572e-34
	P_Value	1.8267e-04	1.8267e-04	1.8267e-04	1.7861e-04	
F3	Average	4.6704e-4	<b>3.121e-13</b>	4.2494e-4	6.1095	<b>8.4423e-13</b>
	STD	1.4749e-3	6.6024e-13	0.00134	5.2693	4.3095e-13
	P_Value	1.8267e-04	0.0046	1.8267e-04	1.7962e-04	
F4	Average	2.5762e-4	<b>5.9491e-05</b>	4.2201e-4	3.3886e-4	3.144e-4
	STD	7.4297e-05	1.9688e-05	1.2868e-4	1.5099e-4	1.4002e-4
	P_Value	0.5205	1.6876e-04	0.0883	0.7333	
F5	Average	8.7041e-15	1.1546e-14	<b>7.9936e-15</b>	2.2116e-13	<b>7.9936e-15</b>
	STD	2.2469e-15	3.7449e-15	0	1.6476e-13	0
	P_Value	0.3681	0.0137	NaN	6.3403e-05	
F6	Average	0.012801	<b>7.396e-4</b>	0.010334	0.026991	<b>7.1445e-3</b>
	STD	0.011876	2.3388e-3	0.014458	0.02109	8.6478e-3
	P_Value	0.2908	0.0916	0.8735	0.0106	

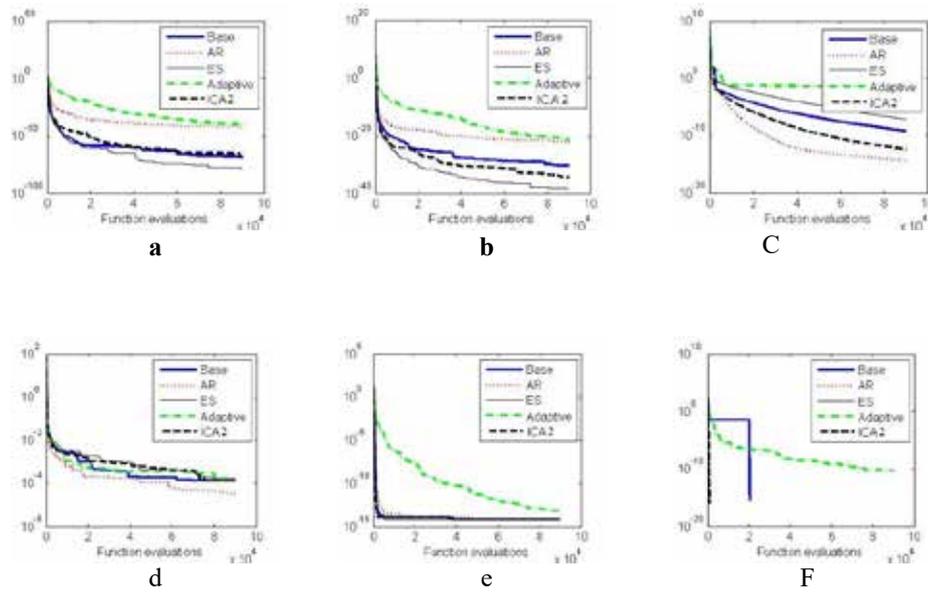


Fig 2: Sample convergence characteristics of the compared ICA-variant on six test function (a) f1 (b) f2 (c) f3 (d) f4 (e) f5 (f) f6

Table 4: Comparison of results obtained with ICA2 and five other evolutionary computation algorithms

Function	JADE (Zhang and Sanderson 2009) mean (STD)	DMP-PSO (Kundu et al. 2014) mean (STD)	FEP (Yao et al. 1999) mean (STD)	IABC (Gao and Liu 2011) mean (STD)	ICA2 mean (STD)
F1	1.3E-54 (9.2E-54)	2.78E-55 (5.64E-53)	5.7E04 (1.3E-04)	6.75e-57 (1.72e-56)	<b>2.3263e-67</b> (5.7736e-67)
F2	3.9E-22 (2.7E-21)	8.96E-25 (3.30E-25)	8.1E-03 (7.7E-04)	6.47e-03 (9.25e-03)	<b>6.3006e-34</b> (8.572e-34)
F3	3.2E-01 (1.1E00)	2.12E-06 (9.32E-06)	5.06E00 (5.87E00)	2.83e+02 (8.94e+01)	<b>8.4423e-13</b> (4.3095e-13)
F4	6.8E-04 (2.5E-04)	1.41E-04 (2.46E-04)	7.6E-03 (2.6E-03)	9.53e-03 (2.45e-03)	<b>3.144e-4</b> (1.4002e-4)
F5	<b>4.4E-15</b> (0)	1.5E-14 (7.38E-15)	1.8E-02 (2.1E-03)	3.87e-14 (8.52e-15)	7.9936e-15 (0)
F6	2.0E-04 (1.4E-03)	<b>0</b> (0)	1.6E-02 (2.2E-02)	<b>0</b> (0)	7.1445e-3 (8.6478e-3)

## REFERENCES

- [1] TALATAHARI, S., et al. Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation*, 2012, 17(3): 1312-1319. doi:10.1016/j.cnsns.2011.08.021
- [2] Holland, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press. (1975).
- [3] Kennedy, J., & Eberhart, R. Particle swarm optimization. In *Proceedings of IEEE international conference on neural Networks*, 1995, 4(2):1942-1948.
- [4] Dorigo, M., & Gambardella, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation*, IEEE Transactions on, 1997, 1(1): 53-66.
- [5] Storn, R., & Price, K. *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces*. Berkeley: ICSI, 1995.
- [6] Webster, B., & Bernhard, P. J. A local search optimization algorithm based on natural principles of gravitation, 2003. <http://hdl.handle.net/11141/117>
- [7] Yang, X. S. *Firefly algorithms for multimodal optimization*. In *Stochastic algorithms: foundations and applications*, Springer Berlin Heidelberg, 2009, (pp. 169-178). doi : 10.1007/978-3-642-04944-6\_14
- [8] Atashpaz-Gargari, E., & Lucas, C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *Evolutionary Computation*, 2007. CEC 2007. IEEE Congress on, (2007, September), (pp. 4661-4667). IEEE. doi : 10.1109/CEC.2007.4425083
- [9] Kaveh, A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Springer International Pu. (2014),(page 349-355, ch 11) .doi: 10.1007/978-3-319-05549-7
- [10] Esmailzadeh, M. (2013). A modified colonial competitive algorithm for optimizing convex functions. *International Journal of Intelligent Computing and Cybernetics*, 2013, 6(4):370-385] doi: 10.1108/IJICC-02-2013-0006
- [11] Atashpaz-Gargari, E., Hashemzadeh, F., & Lucas, C. Designing MIMO PID controller using colonial competitive algorithm: applied to distillation column process. In *Evolutionary Computation*, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on (2008, June),(pp. 1929-1934). IEEE. doi: 10.1109/CEC.2008.4631052
- [12] Nemati, K., Shamsuddin, S. M., & Darus, M. An optimization technique based on imperialist competition algorithm to measurement of error for solving initial and boundary value problems. *Measurement*, (2014),48: 96-108. doi: 10.1016/j.measurement.2013.10.043
- [13] Zarandi, M. H., Zarinbal, M., Ghanbari, N., & Turksen, I. B. A new fuzzy functions model tuned by hybridizing imperialist competitive algorithm and simulated annealing. Application: Stock price prediction. *Information Sciences*, 2013,222, 213-228. Doi: 10.1016/j.ins.2012.08.002
- [14] Niknam, T., Taherian Fard, E., Pourjafarian, N., & Rousta, A. An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering. *Engineering Applications of Artificial Intelligence*, 2011,24(2): 306-317. doi: 10.1016/j.engappai.2010.10.001
- [15] Kayvanfar, V., & Zandieh, M. The economic lot scheduling problem with deteriorating items and shortage: an imperialist competitive algorithm. *The International Journal of Advanced Manufacturing Technology*, 2012, 62(5-8):759-773. doi: 10.1007/s00170-011-3820-6
- [16] Karimi, N., Zandieh, M., & Najafi, A. A. Group scheduling in flexible flow shops: a hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism. *International Journal of Production Research*, 2011, 49(16): 4965-4977. doi : 10.1080/00207543.2010.481644
- [17] Mohammadi-ivatloo, B., Rabiee, A., Soroudi, A., & Ehsan, M. Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch. *Energy*, 2012, 44(1): 228-240. doi: 10.1016/j.energy.2012.06.034
- [18] Ahmadi, M. A., Ebadi, M., Shokrollahi, A., & Majidi, S. M. J. Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir. *Applied Soft Computing*, 2013,13(2): 1085-1098. doi: 10.1016/j.asoc.2012.10.009
- [19] Meysam Mousavi, S., Tavakkoli-Moghaddam, R., Vahdani, B., Hashemi, H., & Sanjari, M. J. A new support vector model-based imperialist competitive algorithm for time estimation in new product development projects. *Robotics and Computer-Integrated Manufacturing*, 2013, 29(1): 157-168. doi: 10.1016/j.rcim.2012.04.006
- [20] Coelho, L. D. S., Afonso, L. D., & Alotto, P. A modified imperialist competitive algorithm for optimization in electromagnetics. *Magnetics*, IEEE Transactions on, 2012,48(2): 579-582. doi: 10.1109/TMAG.2011.2172400
- [21] Lucas, C., Nasiri-Gheidari, Z., & Tootoonchian, F. Application of an imperialist competitive algorithm to the design of a linear induction motor. *Energy Conversion and Management*, 2010,51(7): 1407-1411. doi: 10.1016/j.enconman.2010.01.014
- [22] Liu, J. Y. C., Su, C., & Chiu, C. T. On the Convergence of Imperialist Competitive Algorithm. In *Modelling Symposium (AMS)*, 2013 7th Asia (pp. 16-20). IEEE. doi: 10.1109/AMS.2013.9
- [23] Xing, B., & Gao, W. J. Imperialist Competitive Algorithm. In *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, Springer International Publishing,2014, 203-209. doi: 10.1007/978-3-319-03404-1\_15
- [24] Afonso, L. D., Mariani, V. C., & Dos Santos Coelho, L. Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization. *Expert Systems with Applications*, 2013, 40(9):3794-3802. doi: 10.1016/j.eswa.2012.12.093
- [25] Naimi Sadigh, A., Mozafari, M., & Karimi, B.

- Manufacturer–retailer supply chain coordination: A bi-level programming approach. *Advances in Engineering Software*, 2012,45(1):144-152. doi: 10.1016/j.advengsoft.2011.09.008
- [26] Duan, H., Xu, C., Liu, S., & Shao, S. Template matching using chaotic imperialist competitive algorithm. *Pattern recognition letters*, 2010, 31(13):1868-1875. doi: 10.1016/j.patrec.2009.12.005
- [27] Behnamian, J., & Zandieh, M. A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. *Expert Systems with Applications*, 2011, 38(12):14490-14498. doi: 10.1016/j.eswa.2011.04.241
- [28] Ramezani, F., Lotfi, S., & Soltani-Sarvestani, M. A. A hybrid evolutionary imperialist competitive algorithm (HEICA). In *Intelligent Information and Database Systems Springer Berlin Heidelberg*, 2012 ,(pp. 359-368). doi: 10.1007/978-3-642-28487-8\_37
- [29] Lepagnot, J., Idoumghar, L., & Fodorean, D. Hybrid Imperialist Competitive Algorithm with Simplex Approach: Application to Electric Motor Design. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*,(2013, October), (pp. 2454-2459). IEEE. doi: 10.1109/SMC.2013.419
- [30] Goldansaz, S. M., Jolai, F., & Zahedi Anaraki, A. H. A hybrid imperialist competitive algorithm for minimizing makespan in a multi-processor open shop. *Applied Mathematical Modelling*, 2013, 37(23): 9603-9616. doi: 10.1016/j.apm.2013.05.002
- [31] Jula, A., Othman, Z., & Sundararajan, E. A hybrid imperialist competitive-gravitational attraction search algorithm to optimize cloud service composition. In *Memetic Computing (MC), 2013 IEEE Workshop on* (2013, April) (pp. 37-43). IEEE. doi : 10.1109/MC.2013.6608205
- [32] Xu, J., He, H., & Man, H. DCPE co-training for classification. *Neurocomputing*, 2012, 86:75-85.
- [33] Xu, J., & Man, H. Dictionary learning based on laplacian score in sparse coding. In *Machine Learning and Data Mining in Pattern Recognition, Springer Berlin Heidelberg*, 2011 ,(pp. 253-264).
- [34] Yao, X., Liu, Y., & Lin, G. Evolutionary programming made faster. *Evolutionary Computation, IEEE Transactions on*, 1999, 3(2):82-102.
- [35] Kundu, R., Das, S., Mukherjee, R., & Debchoudhury, S. An improved particle swarm optimizer with difference mean based perturbation. *Neurocomputing*, 2014,129, 315-333. doi : 10.1016/j.neucom.2013.09.026
- [36] Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics bulletin*, 1945,1:80-83.
- [37] Zhang, J., & Sanderson, A. C. JADE: adaptive differential evolution with optional external archive. *Evolutionary Computation, IEEE Transactions on*, 2009, 13(5): 945-958. doi: 10.1109/TEVC.2009.2014613
- [38] Gao, W., & Liu, S. Improved artificial bee colony algorithm for global optimization. *Information Processing Letters*, 2011, 111(17), 871-882. doi: 10.1016/j.ipl.2011.06.002
- [39] Kaveh, A., & Talatahari, S. Optimum design of skeletal structures using imperialist competitive algorithm. *Computers & structures*, 2010,88(21):1220-1229.