

Article

Interaction Enhanced Imperialist Competitive Algorithms

Jun-Lin Lin *, Yu-Hsiang Tsai, Chun-Ying Yu and Meng-Shiou Li

Department of Information Management, Yuan Ze University, 135 Yuan-Tung Road, Chungli, Taoyuan 32003, Taiwan; E-Mails: michael90219@gmail.com (Y.-H.T.); jun2093@gmail.com (C.-Y.Y.); peregrine.plus@gmail.com (M.-S.L.)

* Author to whom correspondence should be addressed; E-Mail: jun@saturn.yzu.edu.tw; Tel.: +886-3-4638800 (ext. 2611); Fax: +886-3-4352077.

Received: 11 July 2012; in revised form: 14 September 2012 / Accepted: 20 September 2012 / Published: 15 October 2012

Abstract: Imperialist Competitive Algorithm (ICA) is a new population-based evolutionary algorithm. It divides its population of solutions into several sub-populations, and then searches for the optimal solution through two operations: assimilation and competition. The assimilation operation moves each non-best solution (called *colony*) in a sub-population toward the best solution (called *imperialist*) in the same sub-population. The competition operation removes a colony from the weakest sub-population and adds it to another sub-population. Previous work on ICA focuses mostly on improving the assimilation operation or replacing the assimilation operation with more powerful meta-heuristics, but none focuses on the improvement of the competition operation. Since the competition operation simply moves a colony (*i.e.*, an inferior solution) from one sub-population to another sub-population, it incurs weak interaction among these sub-populations. This work proposes Interaction Enhanced ICA that strengthens the interaction among the imperialists of all sub-populations. The performance of Interaction Enhanced ICA is validated on a set of benchmark functions for global optimization. The results indicate that the performance of Interaction Enhanced ICA is superior to that of ICA and its existing variants.

Keywords: Imperialist Competition Algorithm; Island Model Genetic Algorithm; optimization

1. Introduction

Optimization is a rapidly developing field due to its practical application to many real life problems. In this paper, we study a special class of optimization problem in the form of:

$$\min f(\mathbf{x}), \text{ s.t. } \mathbf{L} \leq \mathbf{x} \leq \mathbf{U} \quad (1)$$

where $f(\mathbf{x})$ is the objective function to be minimized, $\mathbf{x} = (x_1, x_2, \dots, x_n) \in R^n$ is the variable vector, and $\mathbf{L} = (l_1, l_2, \dots, l_n)$ and $\mathbf{U} = (u_1, u_2, \dots, u_n)$ are the lower bound and upper bound of $\mathbf{x}$, respectively. That is, $l_i \leq x_i \leq u_i$ for $i = 1$ to n .

When the objective function $f(\mathbf{x})$ is nonlinear and has many variables (*i.e.*, n is large), problem (1) becomes hard to solve directly by conventional mathematical methods, such as gradient descent. To resolve this difficulty, evolutionary algorithms based on meta-heuristics have been applied to solve problem (1). Some notable meta-heuristics include Genetic Algorithm [1,2], Particle Swarm Optimization [3], Electromagnetism-like Algorithm [4], Artificial Immune System [5], Artificial Bee Colony Algorithm [6,7], Ant Colony Optimization [8] and Differential Evolution Algorithm [9]. Imperialist Competitive Algorithm (ICA) is a new evolutionary algorithm based on the meta-heuristic of human's socio-political evolution. It was proposed by Atashpaz-Gargari and Lucas in 2007 [10], and since its inception, it has been applied to solve many optimization problems in scheduling [11–15], classification [16,17] and machinery design [18–25].

ICA is a population-based method, whose population contains a set of solutions, called *countries*. ICA divides its population of countries into several sub-populations, called *empires*. Within each empire, ICA moves all non-best countries (called *colonies*) in an empire toward the best country (called *imperialist*) in the same empire through an operation called *assimilation*, which can be regarded as a primitive form of Particle Swarm Optimization [26]. The feature that distinguishes ICA from other evolutionary algorithms is that ICA allows all empires to interact via competition with each other. The competition operation of ICA simply moves a colony from the weakest empire to another empire. From this aspect, ICA resembles Island Model Genetic Algorithm [27]. Overall, ICA is essentially an integration of Island Model and a primitive form of PSO.

The impact of the competition operation is weaker than that of the assimilation operation since in each generation only one colony is directly affected by the competition operation but all colonies are moved by the assimilation operation. Previous work on ICA focuses mostly on improving the assimilation operation or replacing the assimilation operation with more powerful meta-heuristics, but none focuses on improving the competition operation or, more generally, on enhancing the interaction among the empires in ICA. The quality of interaction via the competition operation is poor since the interaction only involves moving an inferior solution (*i.e.*, a colony from the weakest empire) to another empire. The objective of this paper is to enhance the interaction among the empires in ICA. Two methods are proposed to achieve this objective. The first method utilizes all imperialists to create a new artificial imperialist, and the second method uses a crossover operation on the imperialists to allow effective interaction among all empires. Performance study using a set of well-known benchmark functions supports the effectiveness of the proposed methods.

The rest of this paper is organized as follows. Section 2 reviews ICA and its variants. Section 3 describes our proposed methods. Section 4 gives experimental results. Section 5 concludes this paper and gives directions for future research.

2. Literature Review

2.1. ICA Basic Concept

This section reviews the original ICA [10]. However, as the original ICA has some design problems, we adopt Atashpaz-Gargari's Matlab implementation of ICA [28] whenever appropriate. The original ICA is shown in Figure 1. It proceeds in two stages: initialization and evolution.

Figure 1. The original Imperialist Competitive Algorithm (ICA) algorithm.

```

1. Initialization
2. While the termination criteria is not satisfied Do
3. For each empire  $i$  Do
4. Perform assimilation for all colonies in empire  $i$ ;
5. Calculate the cost of each colony in empire  $i$ ;
6. Update the imperialist in empire  $i$ ;
7. End
8. Calculate the cost of each empire;
9. Compete among empires;
10. End
  
```

During the initialization stage (line 1 in Figure 1), a user-specified number of randomly generated solutions $\{\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^N\}$ are created, where each solution $\mathbf{p}^i$ ($1 \leq i \leq N$) is a $1 \times n$ array and is called a *country*, and N denotes the number of countries in the population. Then, the cost of each country $\mathbf{p}^i$ is calculated as $f(\mathbf{p}^i)$. A user-specified number of countries with the lowest cost in the population are chosen as *imperialists*, and the remaining countries as *colonies*. Let N_{imp} denote the number of imperialists. Next, N_{imp} empires are formed by first assigning an imperialist to each empire, and then randomly assigning all colonies to these empires such that the number of colonies of an empire is proportional to the power of its imperialist. That is,

$$P_i = \begin{cases} 1.3(\max_{1 \leq j \leq N_{imp}} \{c_j\}) - c_i & \text{if } \max_{1 \leq j \leq N_{imp}} \{c_j\} > 0 \\ 0.7(\max_{1 \leq j \leq N_{imp}} \{c_j\}) - c_i & \text{if } \max_{1 \leq j \leq N_{imp}} \{c_j\} \leq 0 \end{cases} \quad (2)$$

$$NC_i = \text{round} \left(\frac{P_i}{\sum_{j=1}^{N_{imp}} P_j} \times (N - N_{imp}) \right) \quad (3)$$

where c_i and P_i denote the *cost* and *power* of the imperialist of empire i , respectively, and NC_i denotes the number of colonies assigned to empire i , where $1 \leq i \leq N_{imp}$. Notably, Equation (2), proposed by Atashpaz-Gargari [28], calculates the power of an imperialist in a way different from that in the

original ICA [10] to avoid the problem of assigning zero colony to the weakest empire in the original ICA.

During the evolution stage (lines 2–10 in Figure 1), assimilation within each empire and competition among all empires occur in every generation until the termination condition (e.g., all countries have converged or a user-specified number of generations has been reached) is satisfied. Assimilation within an empire is achieved by moving all colonies of the empire toward their imperialist (line 4 in Figure 1). Given a colony $\mathbf{p}^c$ and its imperialist $\mathbf{p}^i$, the assimilation operation moves $\mathbf{p}^c$ as follows [28]:

$$\mathbf{p}^c = \mathbf{p}^c + \beta * \boldsymbol{\delta} .* (\mathbf{p}^i - \mathbf{p}^c) \quad (4)$$

where β is a parameter with default value 4, $\boldsymbol{\delta}$ is a $1 \times n$ array whose elements are random values between 0 and 1, and “ $.*$ ” denotes element-by-element multiplication between two $1 \times n$ arrays. For optimization problem with bounded variables, Equation (4) could generate values outside the search space. If this happens, the out-of-bound value, e.g., x_i on the i -th dimension, is simply replaced by its nearest boundary (i.e., U_i or L_i).

After moving all colonies to their new positions through the assimilation operation, their costs are recalculated (line 5 in Figure 1) and then compared against the cost of their imperialists (line 6 in Figure 1). If the cost of a colony is smaller than the cost of its imperialist, the colony and the imperialist swap roles to ensure that the imperialist of an empire is always the country with the lowest cost in the empire.

Next, the cost (denoted by Q_i) and the power (denoted by E_i) of each empire i are calculated using the cost of its imperialist (denoted by c_i) and the average cost of the colonies in empire i , as shown in Equations (5) and (6), respectively (line 8 in Figure 1).

$$Q_i = c_i + \xi \text{ mean}\{\text{cost}(\text{colonies of empire } i)\} \quad (5)$$

$$E_i = (\max_{1 \leq j \leq N_{imp}} \{Q_j\}) - Q_i \quad (6)$$

Notably, ξ is a parameter with suggested value 0.1 in the original ICA, but with default value 0.02 in [28]. This study uses $\xi = 0.02$. Competition among all empires (line 9 in Figure 1) is achieved by taking the weakest colony away from the weakest empire and giving it to a chosen empire, where the probability of empire i been chosen is calculated as follows.

$$p_i = \left| \frac{E_i}{\max_{1 \leq j \leq N_{imp}} E_j} \right| \quad (7)$$

When the last colony of an empire is taken away by another empire, the former empire is eliminated, and its imperialist also becomes a colony of the latter empire.

2.2. Variants of ICA

In the literature, many new operations have been proposed to enhance the basic operations in ICA. For example, Atashpaz-Gargari [28] introduced three new features into the original ICA. In each generation, a portion of the colonies are replaced by the same number of randomly generated new candidate solutions, through the *revolution* operation. When two imperialist are too close to each other,

their empires are united as one, through the *unite* operation. Finally, instead of competition in every generation, the frequency of competition among empires is reduced to avoid converging too quickly.

Instead of using the revolution operation to replace a portion of colonies in every generation, Duan *et al.* [29] suggested monitoring the minimal cost and average cost of all countries, and if the minimal cost and the difference between the average cost and the minimal cost are below their respective user-specified thresholds, then a random number of countries in the population are replaced by new countries, generated using a chaotic sequence. Jain and Nigam [30] suggested using a genetic algorithm to generate new countries for the revolution operation. Khorani *et al.* [31] suggested applying genetic algorithm and ICA recursively to benefit from the search power of both algorithms.

The assimilation operation of ICA often converges to a local optimum prematurely [29,30,32], and thus many attempts have been proposed to replace it with more powerful operations. For examples, Bahrami *et al.* [33] used a chaotic map to determine the moving direction for the assimilation operation. Zhang *et al.* [34] defined the assimilation operation as a two-step process. First, each colony is moved precisely toward their imperialists. Then, some randomly chosen colonies are moved further in all dimensions to improve diversity. Lin *et al.* [35] replaced Equation (4) with Equation (8) below to allow some colonies moving away from their imperialists:

$$\mathbf{p}^c = \mathbf{p}^c + (\beta \times \delta - 1) \times (\mathbf{p}^i - \mathbf{p}^c) \quad (8)$$

They also suggested bouncing the out-of-bound values (e.g., x_i on the i -th dimension) into the feasible space using Equation (9), instead of sticking them to the nearest boundary.

$$x_i = \begin{cases} x_i & \text{if } L_i \leq x_i \leq U_i \\ 2U_i - (x_i - \left\lfloor \frac{x_i - U_i}{U_i - L_i} \right\rfloor (U_i - L_i)) & \text{if } x_i > U_i \\ 2L_i - (x_i + \left\lfloor \frac{L_i - x_i}{U_i - L_i} \right\rfloor (U_i - L_i)) & \text{if } x_i \leq L_i \end{cases} \quad (9)$$

Lin *et al.* [26] indicated that the assimilation operation is a primitive form of Particle Swarm Optimization (PSO), and thereby suggested replacing the assimilation operation with PSO. Similarly, Karimi *et al.* [13] replaced the assimilation operation with an Electromagnetism-like Algorithm. Nozarian and Jahan [36] applied ICA as the local search mechanism in a memetic algorithm. Bahrami *et al.* [37] proposed an assimilation operation adaptive to the distribution of colonies to balance between the exploration and exploitation abilities of ICA.

3. Interaction Enhanced ICA

From the literature review in Section 2.2, it is obvious that previous work on ICA mostly focuses on improving the assimilation operation (*i.e.*, the evolution within each empire), and pays little or no attention on the competition operation (*i.e.*, the interaction among all empires). Ironically, the competition operation is what distinguishes ICA from other evolutionary algorithms, as its name implies. With ICA, each empire evolves almost separately, and the only interaction among empires is through the competition operation. However, the competition operation just removes a colony from the

weakest empire and then adds the colony to another empire. Intuitively, the weakest empire does not benefit from this operation. The empire winning a colony does not benefit much from this operation either, because the quality of the colony is poor. Notably, the growth and decline of the numbers of colonies in these two empires rebalance the computational effort towards the stronger empire. Furthermore, in each generation, the assimilation operation affects all colonies, while the competition operation affects only one colony. Therefore, the competition operation has much less impact on the performance than the assimilation operation.

To strengthen the power of the competition operation, or more generally to enhance the quality of interaction among empires, we add an interaction step to ICA such that the niche of each empire can be shared by other empires. Since the imperialist of an empire is the best country in the empire, exchanging information about the imperialist with other empires posts a greater benefit than exchanging information about a colony, as did in the original competition operation. The resulting algorithm, called Interaction Enhanced ICA, is shown in Figure 2. It differs from the original ICA (see Figure 1) in three aspects. Firstly, the assimilation steps of Lin *et al.* [35] are adopted to improve the diversity of the assimilation operation (lines 4 and 5 in Figure 2). Secondly, an interaction step is added (line 10 of Figure 2). Two ways to perform the interaction step are described in detail in Sections 3.1 and 3.2. Thirdly, a new parameter ρ is introduced to control the frequency of competition (lines 11 and 12 in Figure 2) where $0 \leq \rho \leq 1$. The original ICA can be regarded as having $\rho = 1$ since competition occurs in every generation. Intuitively, a smaller ρ causes the number of empires to reduce at a slower speed. Since the preceding interaction step only works with more than one empire, a small ρ can increase the impact of the interaction step.

Figure 2. Interaction Enhanced ICA.

1. **Initialization**
2. While the termination criteria is not satisfied Do
3. For each empire i Do
4. Perform **assimilation** for all colonies in empire i using Equation (8);
5. Move all colonies of empire i with out-of-bound values using Equation (9);
6. Calculate the cost of each colony in empire i ;
7. Update the imperialist in empire i ;
8. End
9. Calculate the cost of each empire using Equation (5);
10. If there exist more than one empires then perform **interaction** operation;
11. Generate a random value $\mu \sim U(0,1)$;
12. If $\mu \leq \rho$ then perform **competition** among empires; // ρ is a parameter
13. End

3.1. Artificial Imperialist

This section proposes the idea of artificial imperialist, and describes how it is used in Interaction Enhanced ICA. Let $\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^m$ denote the imperialists in current population, listed in ascending order of their costs. An artificial imperialist $\mathbf{p}^a$ is constructed as the weighted sum of all imperialists, as shown below.

$$\mathbf{p}^a = \sum_{i=1}^m w_i \mathbf{p}^i \quad (10)$$

$$w_i = \frac{(0.9)^i}{\sum_{j=1}^m (0.9)^j} \quad (11)$$

The weights of these imperialists form a geometric sequence with ratio 0.9. A better imperialist has a larger weight. Intuitively, the artificial imperialist combines the niche of all imperialists, and allows exploration of global optimum efficiently. To employ artificial imperialists in Interaction Enhanced ICA, the steps in Figure 3 are used for the interaction step in Figure 2. If the artificial imperialist $\mathbf{p}^a$ is better than the weakest imperialist $\mathbf{p}^m$, then $\mathbf{p}^a$ becomes the new imperialist of empire m , and $\mathbf{p}^m$ is simply discarded. We denote the Interaction Enhanced ICA using Artificial Imperialist as ICAAI.

Figure 3. Interaction using Artificial Imperialist.

1. Sort all imperialists in ascending order of their cost as $\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^m$;
2. Construct artificial imperialist $\mathbf{p}^a$ using Equations (10) and (11);
3. If $f(\mathbf{p}^a) < f(\mathbf{p}^m)$ then
4. $\mathbf{p}^a$ replaces $\mathbf{p}^m$ as the imperialist of empire m ;
5. End If

ICAAI only incurs one more calculation of the objective function per generation than the original ICA does. However, the time complexity remains the same for both methods.

3.2. Crossover Imperialists

With ICA, the imperialist of an empire cannot move unless it is no longer the best country in the empire. This situation is similar to Electromagnetic-like algorithm [4] where all particles except the best particle are allowed to move. However, with ICA, there exist several empires, and consequently, several imperialists. Thus, this motivates the idea of exchanging information among the imperialists to improve the convergence speed of ICA. Based on this motivation, this section proposes crossover imperialists (see Figure 4) for the interaction step in Interaction Enhanced ICA.

Figure 4. Interaction using Crossover Imperialists.

1. For $k = 1$ to $\lfloor vm \rfloor$ Do
2. Randomly choose two imperialists $\mathbf{p}^i$ and $\mathbf{p}^j$ from all imperialists, assuming $f(\mathbf{p}^i) \leq f(\mathbf{p}^j)$;
3. Uniform crossover $\mathbf{p}^i$ and $\mathbf{p}^j$ to generate two new countries $\mathbf{q}^1$ and $\mathbf{q}^2$;
4. Assign the best of $\mathbf{p}^i, \mathbf{p}^j, \mathbf{q}^1$ and $\mathbf{q}^2$ as the imperialist of the empire i ;
5. Assign the 2nd best of $\mathbf{p}^i, \mathbf{p}^j, \mathbf{q}^1$ and $\mathbf{q}^2$ as the imperialist of the empire j ;
6. End

Notably, v in step 1 of Figure 4 is a parameter between 0 and 1 to control the crossover ratio, and m is the number of imperialists in the current population. Given two countries $\mathbf{p}^i$ and $\mathbf{p}^j$, the steps to generate two new countries $\mathbf{q}^1$ and $\mathbf{q}^2$ using uniform crossover are shown in Figure 5.

Figure 5. Uniform crossover of two countries.

```

1. For  $d=1$  to  $n$  Do // for each dimension
2. Generate a random number  $r \sim U(0,1)$ ;
3. If ( $r>0.5$ ) then
4.  $q_d^1 = p_d^i$  and  $q_d^2 = p_d^j$ ;
5. Else
6.  $q_d^1 = p_d^j$  and  $q_d^2 = p_d^i$ ;
7. End if
8. End

```

Denote Interaction Enhanced ICA using Crossover Imperialists as ICACI. ICACI incurs $[vm]$ more calculations of the objective function per generation than ICA does. Notably, m equals N_{imp} initially and is non-monotonic decreasing afterwards. The time complexity of ICACI remains the same as that of ICA.

4. Experimental Results

4.1. Experiment Settings

A set of 13 well-known benchmark functions, listed in Table 1, was used in this experiment [33,38]. The number of dimensions (*i.e.*, n in Table 1) of the search space is set to 30. Functions f_1 – f_5 are unimodal, and functions f_6 – f_{13} are multimodal. Notably, the function u in functions f_{12} and f_{13} is defined as follows.

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a; \\ 0, & -a \leq x_i \leq a; \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$$

This performance study contains two experiments. In experiment 1, PSO and three ICA variants (Perturbed ICA [35], ICAAI and ICACI) were tested. For the three ICA methods, the number of countries, the number of imperialists, β in Equations (4) and (8), and ξ in Equation (5) are set to 88, 8, 4 and 0.02, respectively. For ICACI, the crossover ratio v is set to 0.8. For both ICAAI and ICACI, ρ in line 12 of Figure 2 is set to 1. For PSO, the number of particles, inertia weight w , and two acceleration coefficients c_1 and c_2 are set to 88, 0.7, 2 and 2, respectively.

Experiment 2 studied the effect of competition frequency on both ICAAI and ICACI. All experimental settings are the same as those in experiment 1 except that the value of ρ varied from 0 to 1 in step of 0.1. Notably, $\rho = 0$ means that no competition occurs among empires.

For each setting, 30 runs were conducted, and their average performance was obtained. Each run stops until the maximum number of generations (1000 or 3000 in experiment 1, and 1000 in experiment 2) is reached. For the same run, the same set of initial countries (or particles in PSO) was generated for all four methods.

Table 1. Benchmark functions.

f	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^n x_i^2$	$x_i \in [-100, 100]$	$f_1(\mathbf{0}) = 0$
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$x_i \in [-10, 10]$	$f_2(\mathbf{0}) = 0$
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j^2 \right)$	$x_i \in [-100, 100]$	$f_3(\mathbf{0}) = 0$
$f_4(x) = \max_i \{x_i 1 \leq i \leq n\}$	$x_i \in [-100, 100]$	$f_4(\mathbf{0}) = 0$
$f_5(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	$x_i \in [-100, 100]$	$f_5(\mathbf{p}) = 0, -0.5 \leq p_i < 0.5$
$f_6(x) = -\sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	$x_i \in [-500, 500]$	$f_6(\mathbf{420.97}) = -418.9829n$
$f_7(x) = \sum_{i=1}^{n-1} (100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$x_i \in [-100, 100]$	$f_7(\mathbf{1}) = 0$
$f_8(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$x_i \in [-10, 10]$	$f_8(\mathbf{0}) = 0$
$f_9(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$x_i \in [-600, 600]$	$f_9(\mathbf{0}) = 0$
$f_{10}(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)}$	$x_i \in [-32, 32]$	$f_{10}(\mathbf{0}) = 0$
$f_{11}(x) = -\sum_{i=1}^n \sin(x_i) \left[\sin \frac{ix_i^2}{\pi} \right]^{20}$	$x_i \in [0, \pi]$	$f_{11} > -n$
$f_{12}(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$, where $y_i = 1 + (x_i + 1)/4$	$x_i \in [-50, 50]$	$f_{12}(\mathbf{-1}) = 0$
$f_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	$x_i \in [-50, 50]$	$f_{13}(\mathbf{1}) = 0$

4.2. Performance Comparison

4.2.1. Experiment 1: Impact of Interaction Operation

This experiment compares ICAAI and ICACI against PSO and Perturbed ICA [35]. Wilcoxon signed-rank test has been conducted between every two methods, and p -values less than 0.05 are deemed statistically significant. Table 2 shows the average and the standard deviation of the best objective values obtained by each method after 1000 generations. Since ICAAI and ICACI are essentially the Perturbed ICA with the addition of an interaction step (line 10 of Figure 2), the impact of the interaction step can be shown by comparing the performance of ICAAI and ICACI against that of the Perturbed ICA. Wilcoxon signed-rank test shows that ICAAI statistically significantly outperforms the Perturbed ICA on benchmark functions $f_1, f_2, f_5, f_7, f_8, f_9, f_{10}, f_{12}$ and f_{13} , and that there is no significant difference between the two methods on the rest of the benchmark functions. Similarly, Wilcoxon signed-rank test shows that ICACI statistically significantly outperforms the Perturbed ICA on benchmark functions f_1, f_7, f_8, f_{12} and f_{13} , and that the Perturbed ICA statistically significantly

outperforms ICACI only on benchmark function f_9 . Therefore, the addition of an interaction step in ICAAI and ICACI improves the Perturbed ICA on some benchmark functions.

Wilcoxon signed-rank test between ICAAI and ICACI shows that ICAAI statistically significantly outperforms ICACI on benchmark functions $f_1, f_4, f_5, f_7, f_8, f_9, f_{10}, f_{12}$ and f_{13} , and that there is no significant difference between the two methods on the rest of the benchmark functions. Therefore, the interaction step using artificial imperialist appears to be more effective than that using crossover imperialists.

Wilcoxon signed-rank test between ICAAI and PSO shows that ICAAI statistically significantly outperforms PSO on benchmark functions f_3, f_6 and f_9 , and that PSO statistically significantly outperforms ICAAI on benchmark functions f_1, f_2, f_5, f_7, f_8 and f_{13} . Similarly, Wilcoxon signed-rank test between ICACI and PSO shows that ICACI statistically significantly outperforms PSO also on benchmark functions f_3, f_6 and f_9 , and that PSO statistically significantly outperforms ICACI on benchmark functions $f_1, f_2, f_4, f_5, f_7, f_8, f_{10}, f_{12}$ and f_{13} . Thus, PSO appears to be better than both ICAAI and ICACI, according to Wilcoxon signed-rank test. However, later in this section, we show that ICAAI becomes competitive to PSO after 3000 generations, according to Wilcoxon signed-rank test.

Experiment using Atashpaz-Gargari's ICA [28] has also been conducted, but its performance results are much poorer than the four methods discussed herein and thus are not included in Table 2. Wilcoxon signed-rank test shows that both ICAAI and ICACI statistically significantly outperform Atashpaz-Gargari's ICA on all benchmark functions.

Table 2. Average and standard deviation of the best objective values over 30 runs after 1000 generations. The standard deviation is shown in parentheses.

F	PSO	Perturbed ICA [35]	ICAAI	ICACI
f_1	1.57×10^{-16} (8.6×10^{-16})	8.312×10^{-6} (1.3×10^{-5})	3.757×10^{-10} (2×10^{-9})	2.1×10^{-7} (9.7×10^{-7})
f_2	1.015×10^{-3} (5.6×10^{-3})	3.559×10^{-4} (7.48×10^{-4})	1.103×10^{-7} (2.38×10^{-7})	5.08×10^{-5} (2.1×10^{-4})
f_3	1.5×10^{-18} (5.8×10^{-18})	2.687×10^{-4} (4.5×10^{-4})	1.53×10^{-10} (6.4×10^{-10})	2.835×10^{-6} (7×10^{-6})
f_4	14.44 (3.7)	6.607 (2.2)	1.989×10^{-1} (0.2)	8.134 (3.1)
f_5	6.67×10^{-2} (0.25)	19.57 (37.7)	0.3 (0.79)	46.27 (146.88)
f_6	-1.135×10^4 (367)	-1.140×10^4 (280)	-1.142×10^4 (256)	-1.143×10^4 (304)
f_7	31.2 (16.3)	230.3 (295.9)	100.2 (131.3)	126.7 (145)
f_8	38.5 (10.14)	5.945 (3.03)	5.172 (2.94)	6.008 (2.89)
f_9	1.53×10^{-2} (0.02)	2.284×10^{-2} (0.03)	1.23×10^{-2} (0.017)	3.81×10^{-2} (0.037)
f_{10}	8.4×10^{-7} (4.55×10^{-6})	1.203×10^{-3} (1.3×10^{-3})	4.139×10^{-6} (8.6×10^{-6})	1.063×10^{-3} (0.002)
f_{11}	-23.757 (1.22)	-27.72 (0.59)	-27.68 (0.89)	-27.58 (0.89)
f_{12}	1.037×10^{-2} (0.03)	6.913×10^{-3} (0.026)	1.037×10^{-2} (0.032)	6.91×10^{-3} (0.026)
f_{13}	1.1×10^{-3} (3.35×10^{-3})	1.810×10^{-3} (4.04×10^{-3})	1.83×10^{-3} (4.16×10^{-3})	1.83×10^{-3} (4.16×10^{-3})

To see the possibility of further converging to a better solution, the same experiment is extended from 1000 to 3000 generations, and the results are shown in Table 3. According to Wilcoxon signed-rank test, ICACI statistically significantly outperforms the Perturbed ICA on benchmark functions f_1, f_7, f_8 and f_{12} , and the Perturbed ICA statistically significantly outperforms ICACI only on benchmark function f_3 . ICAAI remains more effective than the Perturbed ICA after 3000 generations. According to Wilcoxon signed-rank test, ICAAI statistically significantly outperforms the Perturbed

ICA on benchmark functions $f_1, f_5, f_7, f_8, f_9, f_{10}$ and f_{12} , and there is no significant difference between ICAAI and the Perturbed ICA on the rest of the benchmark functions.

ICAAI is still more effective than ICACI after 3000 generations. According to Wilcoxon signed-rank test, ICAAI statistically significantly outperforms ICACI on benchmark functions $f_1, f_4, f_5, f_7, f_8, f_9$ and f_{10} , and there is no significant difference between ICAAI and ICACI on the rest of the benchmark functions.

Both ICAAI and ICACI are more competitive to PSO after 3000 generations than after 1000 generations. According to Wilcoxon signed-rank test, ICAAI statistically significantly outperforms PSO on benchmark functions f_3, f_5, f_6, f_7 and f_9 , and PSO statistically significantly outperforms ICAAI on benchmark functions f_1, f_8, f_{10}, f_{12} and f_{13} . Similarly, Wilcoxon signed-rank test shows that ICACI statistically significantly outperforms PSO on benchmark functions f_3, f_6, f_7 and f_9 , and that PSO statistically significantly outperforms ICACI on benchmark functions $f_1, f_4, f_5, f_8, f_{10}, f_{12}$ and f_{13} .

Table 3. Average and standard deviation of the best objective values over 30 runs after 3000 generations. The standard deviation is shown in parentheses.

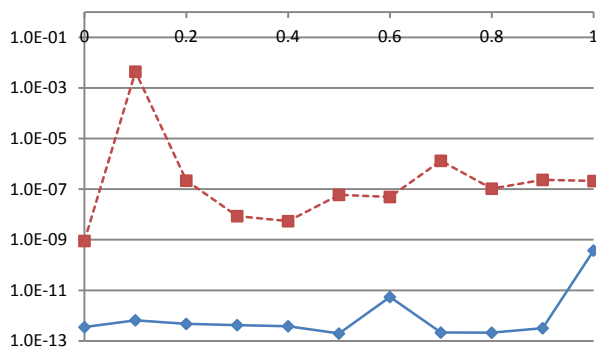
f	PSO	Perturbed ICA [35]	ICAAI	ICACI
f_1	1.89×10^{-54} (9.95×10^{-54})	8.199×10^{-24} (3.3×10^{-23})	2.89×10^{-28} (1.58×10^{-27})	2.23×10^{-26} (9.5×10^{-26})
f_2	1.015×10^{-3} (5.56×10^{-3})	1.726×10^{-15} (3.33×10^{-15})	2.671×10^{-19} (6.1×10^{-19})	9.29×10^{-18} (2.2×10^{-17})
f_3	3.17×10^{-56} (1.5×10^{-55})	1.493×10^{-22} (4.56×10^{-22})	1.665×10^{-25} (9.1×10^{-25})	1.1×10^{-24} (3.6×10^{-24})
f_4	6.67 (2.77)	2.770×10^{-1} (0.19)	5.242×10^{-3} (6.17×10^{-3})	3.396×10^{-1} (0.2)
f_5	0 (0)	19.57 (37.7)	0.3 (0.79)	46.1 (146.9)
f_6	-1.136×10^4 (368.7)	-1.14×10^4 (280.32)	-1.142×10^4 (263)	-1.143×10^4 (303.6)
f_7	29.96 (16.19)	87.34 (118.4)	49.39 (59.02)	59.07 (84.66)
f_8	28.73 (10.34)	3.681 (2.22)	4.676 (3.02)	5.373 (3.44)
f_9	1.53×10^{-2} (0.02)	2.282×10^{-2} (0.031)	1.230×10^{-2} (0.017)	3.807×10^{-2} (0.037)
f_{10}	1.8×10^{-14} (4.2×10^{-15})	6.05×10^{-13} (1.5×10^{-12})	1.68×10^{-13} (3.22×10^{-13})	8.79×10^{-13} (1.8×10^{-12})
f_{11}	-24.42 (1.16)	-28.02 (0.74)	-27.93 (0.84)	-27.59 (0.91)
f_{12}	6.9×10^{-3} (0.026)	6.911×10^{-3} (0.026)	1.037×10^{-2} (0.0317)	6.911×10^{-3} (0.0263)
f_{13}	7.325×10^{-4} (0.0028)	1.099×10^{-3} (3.35×10^{-3})	1.831×10^{-3} (4.17×10^{-3})	1.831×10^{-3} (4.16×10^{-3})

4.2.2. Experiment 2: Impact of Competition Frequency

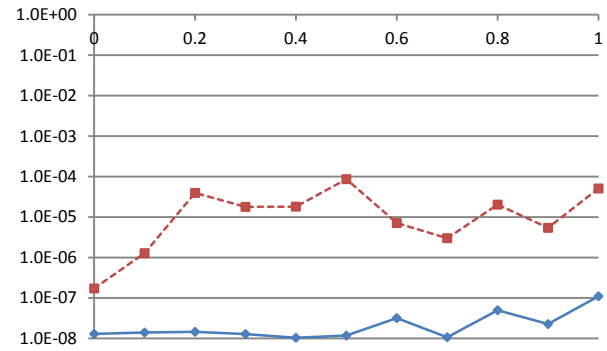
This experiment compares ICAAI and ICACI under various competition frequencies, which are controlled by the value of parameter ρ . Figure 6 shows that ICAAI consistently outperforms ICACI on all of the five unimodal benchmark functions $f_1 \sim f_5$. For $f_1 \sim f_4$, ICAAI often yields better results with smaller competition frequency (i.e., smaller ρ).

Figure 7 shows the results for the eight multimodal benchmark functions $f_6 \sim f_{13}$. ICAAI consistently outperforms ICACI on f_9 and f_{10} . For f_7, f_8 and f_{11} , ICAAI underperforms ICACI only when $\rho = 0$. For f_6 and f_{12} , ICAAI often yields better results than ICACI does. Overall, ICACI performs better than ICAAI only on f_{13} . Smaller competition frequency does not guarantee better performance for both ICAAI and ICACI, however, experimenting with $\rho < 1$ can often find better solutions. For example, Figure 6 and Figure 7 show that the best ρ for ICAAI is always less than 1 on all benchmark functions.

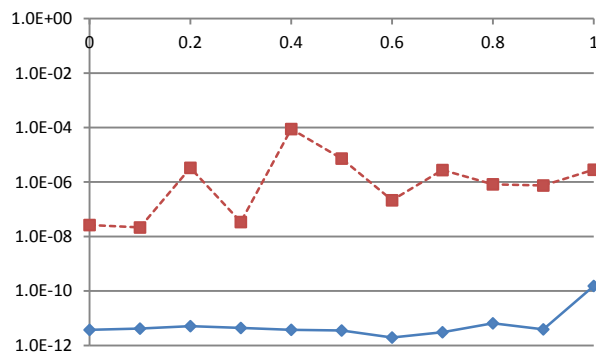
Figure 6. ICAAI (solid line) and ICACI (dashed line) for unimodal benchmark functions $f_1 \sim f_5$. The horizontal axis is ρ , and the vertical axis is the average of the best objective values.



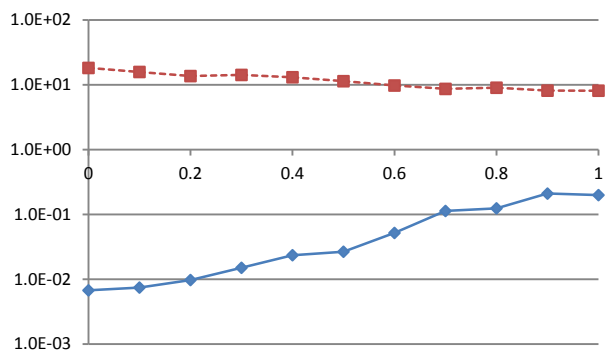
(a) bechmark $f_1(x)$



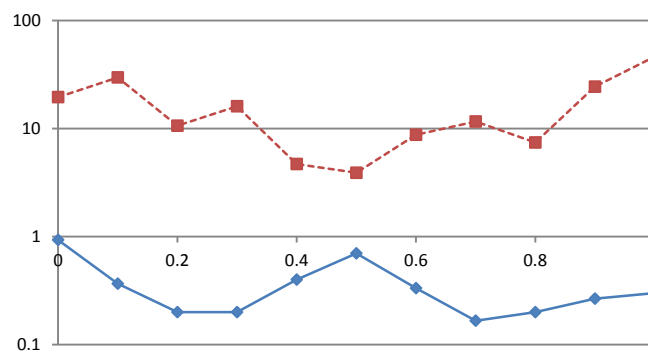
(b) bechmark $f_2(x)$



(c) bechmark $f_3(x)$

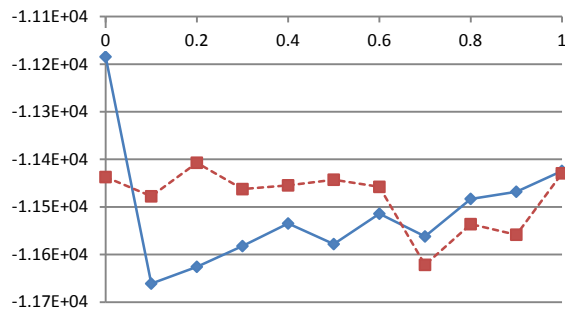


(d) bechmark $f_4(x)$

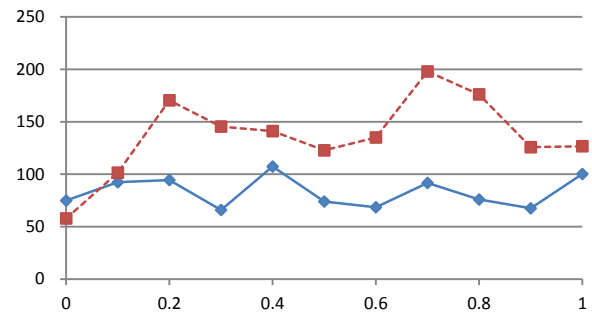


(e) bechmark $f_5(x)$

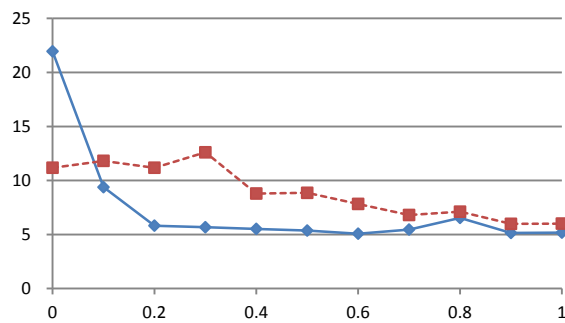
Figure 7. ICAAI (solid line) and ICACI (dashed line) for multimodal benchmark functions $f_6 \sim f_{13}$. The horizontal axis is ρ , and the vertical axis is the average of the best objective values.



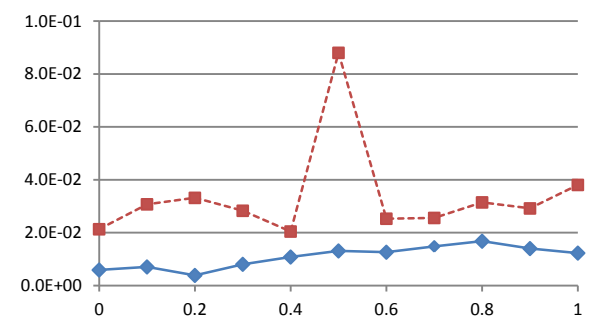
(a) bechmark $f_6(x)$



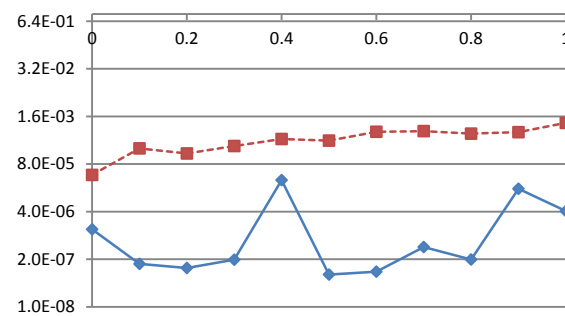
(b) bechmark $f_7(x)$



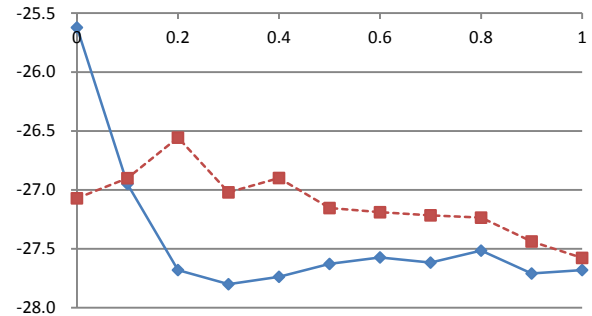
(c) bechmark $f_8(x)$



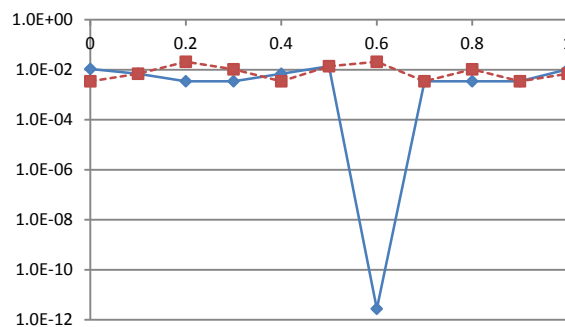
(d) bechmark $f_9(x)$



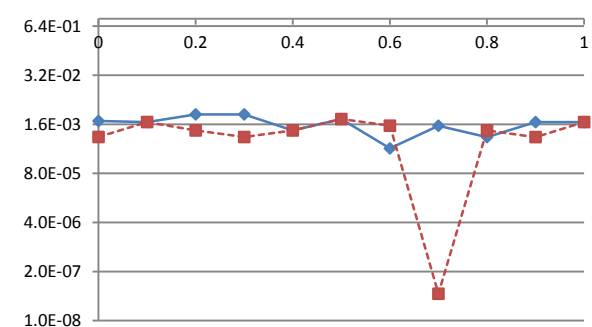
(e) bechmark $f_{10}(x)$



(f) bechmark $f_{11}(x)$



(g) bechmark $f_{12}(x)$



(h) bechmark $f_{13}(x)$

5. Conclusions

ICA often converges to a local optimum [29,30,32]. To resolve this problem, previous work focuses mostly on improving the diversity of the colonies in ICA via perturbed assimilation move [35] or random replacement (e.g., the revolution operation in [28]). This work shifts the focus to improving the interaction among the imperialists. Two new methods, ICAAI and ICACI, are proposed. Both methods do not increase the time complexity of ICA. Experiment 1 shows that both methods often yield better results than ICA, and Experiment 2 shows that a better solution can be found by experimenting ICAAI with lower competition frequency (*i.e.*, $\rho < 1$). Although ICAAI requires less computation than ICACI, ICAAI appears to be more effective than ICACI.

Two possible extensions to ICA are discussed as follows. Firstly, the multiple empires (*i.e.*, sub-populations) in ICA offer a new ground for exploration. Ideas that were originally proposed for single population method can be extended to some or all empires in ICA. For example, Electromagnetism-like Algorithm [4] applies local search on the best candidate solution in the current population to improve the solution quality. With ICA, the same local search idea can be applied to either the imperialist of the best empire or all imperialists.

Secondly, an effective evolutionary method should be equipped with a way to detect stagnation and to know what to do when stagnation occurs [39]. Previous work on this topic can also be extended to each empire or the whole population in ICA. Since the size of an empire is smaller than that of a population, the stagnation detection mechanism should be adjusted accordingly to avoid false detection.

Acknowledgments

This research is supported by National Science Council under Grant NSC 99-2221-E-155-048-MY3.

References

1. Holland, J.H. *Adaptation in Natural and Artificial Systems*; MIT Press: Cambridge, MA, USA, 1992.
2. Chen, S.-H.; Chen, M.-C.; Chang, P.-C.; Chen, Y.-M. Ea/g-ga for single machine scheduling problems with earliness/tardiness costs. *Entropy* **2011**, *13*, 1152–1169.
3. Clerc, M.; Kennedy, J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolut. Comput.* **2002**, *6*, 58–73.
4. Birbil, S.I.; Fang, S.C. An electromagnetism-like mechanism for global optimization. *J. Global Optim.* **2003**, *25*, 263–282.
5. Dasgupta, D.; Yu, S.H.; Nino, F. Recent advances in artificial immune systems: Models and applications. *Appl. Soft Comput.* **2011**, *11*, 1574–1587.
6. Gao, W.F.; Liu, S.Y.; Huang, L.L. A global best artificial bee colony algorithm for global optimization. *J. Comput. Appl. Math.* **2012**, *236*, 2741–2753.
7. Zhang, Y.; Wu, L. Optimal multi-level thresholding based on maximum tsallis entropy via an artificial bee colony approach. *Entropy* **2011**, *13*, 841–859.

8. Dorigo, M.; Maniezzo, V.; Coloni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern.* **1996**, *26*, 29–41.
9. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **1997**, *11*, 341–359.
10. Atashpaz-Gargari, E.; Lucas, C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *Proceedings of IEEE Congress on Evolutionary Computation*, Singapore, 25–28 September 2007; pp. 4661–4667.
11. Behnamian, J.; Zandieh, M. A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. *Expert Syst. Appl.* **2011**, *38*, 14490–14498.
12. Forouharfard, S.; Zandieh, M. An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. *Int. J. Adv. Manuf. Tech.* **2010**, *51*, 1179–1193.
13. Karimi, N.; Zandieh, M.; Najafi, A.A. Group scheduling in flexible flow shops: A hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism. *Int. J. Prod. Res.* **2011**, *49*, 4965–4977.
14. Shokrollahpour, E.; Zandieh, M.; Dorri, B. A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem. *Int. J. Prod. Res.* **2011**, *49*, 3087–3103.
15. Lian, K.; Zhang, C.; Gao, L.; Shao, X. A modified colonial competitive algorithm for the mixed-model u-line balancing and sequencing problem. *Int. J. Prod. Res.* **2012**, doi:10.1080/00207543.2011.653453.
16. MousaviRad, S.J.; Akhlaghian Tab, F.; Mollazade, K. Application of imperialist competitive algorithm for feature selection: A case study on bulk rice classification. *Int. J. Comput. Appl.* **2012**, *40*, 41–48.
17. Karami, S.; Shokouhi, S.B. Application of imperialist competitive algorithm for automated classification of remote sensing images. *Int. J. Comput. Theory Eng.* **2012**, *4*, 137–143.
18. Bagher, M.; Zandieh, M.; Farsijani, H. Balancing of stochastic u-type assembly lines: An imperialist competitive algorithm. *Int. J. Adv. Manuf. Tech.* **2011**, *54*, 271–285.
19. Coelho, L.D.S.; Afonso, L.D.; Alotto, P. A modified imperialist competitive algorithm for optimization in electromagnetics. *IEEE Trans. Magn.* **2012**, *48*, 579–582.
20. Kaveh, A.; Talatahari, S. Optimum design of skeletal structures using imperialist competitive algorithm. *Comput. Struct.* **2010**, *88*, 1220–1229.
21. Kazemi, S.; Ghorbani, A.; Hashemi, S.N. Deployment of the meta heuristic colonial competitive algorithm in synthesis of unequally spaced linear antenna array. *IEICE Electron. Express* **2011**, *8*, 2048–2053.
22. Lucas, C.; Nasiri-Gheidari, Z.; Tootoonchian, F. Application of an imperialist competitive algorithm to the design of a linear induction motor. *Energy Convers. Manag.* **2010**, *51*, 1407–1411.
23. Nazari-Shirkouhi, S.; Eivazy, H.; Ghodsi, R.; Rezaie, K.; Atashpaz-Gargari, E. Solving the integrated product mix-outsourcing problem using the imperialist competitive algorithm. *Expert Syst. Appl.* **2010**, *37*, 7615–7626.
24. Soltanpoor, H.; Nozarian, S.; VafaeiJahan, M. Solving the graph bisection problem with imperialist competitive algorithm. *Int. Conf. Sys. Eng. Model.* **2012**, *34*, 136–140.

25. Rezaei, E.; Karami, A.; Shahhosseini, M. The use of imperialist competitive algorithm for the optimization of heat transfer in an air cooler equipped with butterfly inserts. *Aust. J. Basic Appl. Sci.* **2012**, *6*, 293–301.
26. Lin, J.-L.; Yu, C.-Y.; Tsai, Y.-H. PSO-based imperialist competitive algorithm. *J. Phys. Conf. Ser.* **2012**, in press.
27. Niwa, T.; Tanaka, M. Analysis on the island model parallel genetic algorithms for the genetic drifts. *Simul. Evolut. Learn.* **1999**, *1585*, 349–356.
28. Atashpaz-Gargari, E. Imperialist competitive algorithm (ICA). Available online: <http://www.mathworks.com/matlabcentral/fileexchange/22046-imperialist-competitive-algorithm-ica> (accessed on 10 January 2012).
29. Duan, H.B.; Xu, C.F.; Liu, S.Q.; Shao, S. Template matching using chaotic imperialist competitive algorithm. *Pattern Recogn. Lett.* **2010**, *31*, 1868–1875.
30. Jain, T.; Nigam, M.J. Synergy of evolutionary algorithm and socio-political process for global optimization. *Expert Syst. Appl.* **2010**, *37*, 3706–3713.
31. Khorani, V.; Razavi, F.; Ghoncheh, A. A new hybrid evolutionary algorithm based on ICA and GA: Recursive-ICA-GA. In *IC-AI*; Arabnia, H.R., de la Fuente, D., Kozerenko, E.B., Olivas, J.A., Chang, R., LaMonica, P.M., Liuzzi, R.A., Solo, A.M.G., Eds.; CSREA Press: Las Vegas, NV, USA, 12–15 July 2010; pp. 131–140.
32. Talatahari, S.; Azar, B.F.; Sheikholeslami, R.; Gandomi, A.H. Imperialist competitive algorithm combined with chaos for global optimization. *Commun. Nonlinear Sci.* **2012**, *17*, 1312–1319.
33. Bahrami, H.; Faez, K.; Abdechiri, M. Imperialist competitive algorithm using chaos theory for optimization (cica). In *Proceedings of the 2010 12th International Conference on Computer Modelling and Simulation (UKSim)*, Cambridge, UK, 24–26 March 2010; IEEE Computer Society Conference Publishing Service; pp. 98–103.
34. Zhang, Y.; Wang, Y.; Peng, C. Improved imperialist competitive algorithm for constrained optimization. In *Proceedings of the International Forum on Computer Science-Technology and Applications, 2009*, Chongqing, China, 25–27 December 2009; Volume 1, pp. 204–207.
35. Lin, J.-L.; Cho, C.-W.; Chuan, H.-C. Imperialist competitive algorithms with perturbed moves for global optimization. *Appl. Mech. Mater.* **2012**, in press.
36. Nozarian, S.; Jahan, M.V. A novel memetic algorithm with imperialist competition as local search. *IPCSIT* **2012**, *30*, 54–59.
37. Bahrami, H.; Abdechiri, M.; Meybodi, M.R. Imperialist competitive algorithm with adaptive colonies movement. *Int. J. Intell. Syst. Appl.* **2012**, *2*, 49–57.
38. Ao, Y.; Chi, H. Differential evolution using opposite point for global numerical optimization. *J. Intell. Learn. Syst. Appl.* **2012**, *4*, 1–19.
39. Worasuchee, C. A particle swarm optimization with stagnation detection and dispersion. In *Proceedings of IEEE World Congress on Computational Intelligence*, Hong Kong, China, 1–6 June 2008; pp. 424–429.