A HYBRID PROPOSED IMPERIALIST COMPETITIVE ALGORITHM WITH CONJUGATE GRADIENT APPROACH FOR LARGE SCALE GLOBAL OPTIMIZATION

Ban **A**. **M**itras And **J**alal **A**. **S**ultan
Operations Research & Artificial Techniques Department, College of Computer Science &Mathematics, University of Mosul, Iraq.

(Accepted for publication: June 9, 2013)

Abstract

This paper presents a novel hybrid imperialist competitive algorithm called ICA-CG algorithm. Such an algorithm combines the evolution ideas of the imperialist competitive algorithm and the classic optimization ideas of the conjugate gradient, based on the compensation for solving the large scale optimization. In the ICA-CG algorithm, the process of every iteration is divided into two stages. In the first stage, the randomly, rapidity and wholeness of the imperialist competitive Algorithm are used. In the second stage, one of the common optimization classical techniques, that called conjugate gradient to move imperialist countries, is used. Experimental results for five well known test problems have shown the superiority of the new ICA-CG algorithm, in large scale optimization, compared with the classical GA, ICA, PSO and ABC algorithms, with regard to the convergence of speed and quality of obtained solutions.

Keywords**:** *Large scale global Optimization, Evolutionary Algorithms, Imperialist Competitive Algorithm (ICA),Conjugate Gradient (CG) .*

1 Introduction

The global optimization problem is applicable in every field of science, engineering and business. So far, many Evolutionary Algorithms (EA) [Sarimveis 2005] and [Srinivasan 2003], have been proposed for solving the global optimization problem. Inspired by the natural evolution, EA analogizes the evolution process of biological population, which can adapt the changing environments to the finding of the optimum of the optimization problem through evolving a population of candidate solutions. Some Evolutionary Algorithms for optimization problem are: the Genetic Algorithm (GA) [Goldberg 1989], at first proposed by Holland, in 1962 [Holland 1990], Particle Swarm Optimization algorithms (PSO) that at first proposed by Kennedy and Eberhart [Kennedy 1995], in 1995.In 2007, Artificial Bee Colony (ABC) that at first proposed by Karaboga and Basturk [Karaboga 2007], and in the same year , a new algorithm which is called Imperialist Competitive Algorithm (ICA) [Atashpaz 2007] has been proposed by Atashpaz-Gargari and Lucas , that has inspired from a socio-human phenomenon.

In general, The main advantages of Evolutionary algorithms are: they do not require the objective function to be differentiable or continuous, they do not require the evaluation of gradients and they can escape from local minima.

On the other hand, the Conjugate Gradient (CG) method [Andrei 2007] is a highly efficient direct minimization approach, which is currently the method of choice in wide areas of science and engineering. In computational solid state physics, for example, the CG method is used to minimize directly the total energy of the system of electrons, which is usually a function of a very large number of variables, in small number of iterations [Patel 2000]. The key features behind the great success of the CG approach is the conjugacy property of the search directions, and periodic restart of the iterative minimization procedure each certain number of CG steps.

In this paper, we presents a novel hybrid imperialist competitive algorithm called ICA-CG algorithm that combines the evolution ideas of the imperialist competitive algorithm and classic optimization ideas of the conjugate gradient based on the compensation for solving the large scale optimization.

The paper is organized as follows: In Section 2 some related work is presented. Section 3 describes a brief description of Imperialist Competitive Algorithm (ICA). In Section 4, a brief description of Conjugate Gradient (CG). In Section 5, new approach and the motivation of the ICA-CG algorithm is presented. In Section 6, results are compared with other Evolutionary Algorithms.

2. Some Related Works:

In 2007, Atashpaz and Lucas proposed an algorithm as Imperialist Competitive Algorithm (ICA) [Atashpaz 2007] and [Atashpaz 2008], that has inspired from a socio-human phenomenon. Since 2007 attempts were performed in order to increase the efficiency of the ICA. In 2009, Zhang, Wang and Peng proposed an approach based on the concept of small probability perturbation to enhance the movement of Colonies to Imperialist [Zhang 2009]. In 2010, Faez, Bahrami and Abdechiri, proposed a new method using the chaos theory to adjust the angle of Colonies movement toward the Imperialists' position (*CICA*)[Bahrami 2010a], and in other paper at the same year, they proposed another algorithm that applies the probability density function in order to adapt the angle of colonies' movement towards imperialist's position dynamically, during iterations (AICA) [Bahrami 2010b]. In 2012, Ghodrati, ,Malakooti and Soleimani, proposed a new hybrid method using the ICA and PSO by adding independent countries for large scale[Ghodrati 2012], and in the same year , Ramazani, Lotfi and Soltani proposed a new hybrid method called HEICA which combines Evolutionary algorithm and ICA [Ramezani 2012].

3. Imperialist Competitive Algorithm (ICA)

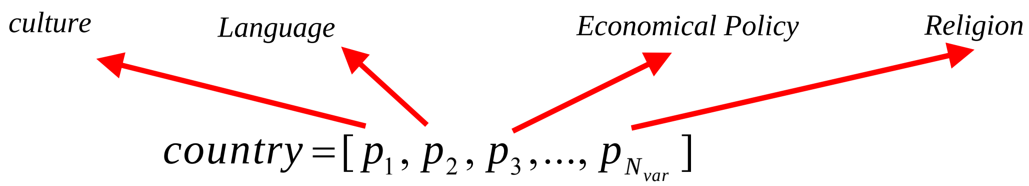
Imperialist Competitive Algorithm (ICA) is a new evolutionary algorithm in the Evolutionary Computation field based on the human's socio-political evolution. The Procedures of the ICA is presented as below:

Step 1:Creation of initial empires

The goal of optimization is to find an optimal solution in terms of the variables of the problem. We form an array of variable values to be optimized. In the GA terminology, this array is called ''chromosome'', but in ICA the term ''country'' is used for this array. In an N_{var}-dimentional optimisation problem, a country is a $1 * N_{var}$-array. This array is defined as following:

$$country = (p_1, p_2, p_3, \dots, p_{N_{var}}),$$

where p_i are the variables to be optimized. The variable values in the country are represented as floating point numbers. Each variable in the country can be interpreted as a socio-political characteristic of a country. From this point of view, all the algorithm does is to search for the best country that is the country with the best combination of socio-political characteristics such as culture, language, economical policy, and even religion. From optimization point of view this leads to find the optimal solution of the problem, the solution with least cost value. Figure (1) shows the interpretation of country using some of socio-political characteristics[Atashpaz 2007] and [Atashpaz 2008].



Figure(1): The candidate solutions of the problem, called country, consists of a combination of some socio-political characteristics such as culture, language and religion.

The cost of a country is found by evaluation of the cost function f at variables $(p_1, p_2, p_3, \dots, p_{N_{var}})$. So we have

$$cost = f(country) = f(p_1, p_2, p_3, \dots, p_{N_{var}}) \quad (1)$$

To start the optimization algorithm, initial countries of size $N_{country}$ is produced. We select N_{imp} of the most powerful countries to form the empires. The remaining N_{col} of the initial countries will be the colonies each of which belongs to an empire.

To form the initial empires, the colonies are divided among imperialists based on their power. That is, the initial number of colonies of an empire should be directly proportionate to its power. To proportionally divide the colonies among imperialists, the normalized cost of an imperialist is defined by

$$C_n = c_n - \max\{c_i\} \quad (2)$$

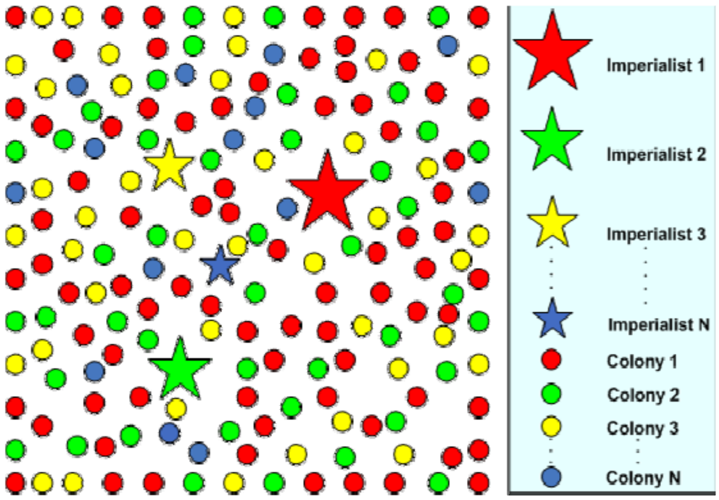
where c_n is the cost of the n[th] imperialist and C_n is its normalized cost. Having the normalized cost of all imperialists, the normalized power of each imperialist is defined by

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (3)$$

The initial colonies are divided among empires based on their power. Then the initial number of colonies of the nth empire will be

$$N.C\_n = round\{p\_n * N\_col\} \quad (4)$$

where $N.C_n$ is the initial number of colonies of the n[th] empire and N_{col} is the total number of initial colonies. To divide the colonies, $N.C_n$ of the colonies are randomly chosen and given to the n[th] imperialist. These colonies along with the nth imperialist form the nth empire. Figure (2) shows the initial empires. As shown in figure (2), bigger empires have greater number of colonies while weaker ones have less. In this figure imperialist 1 has formed the most powerful empire and consequently has the greatest number of colonies [Atashpaz 2007] and [Atashpaz 2008].



Figure(2): Generating the initial empires: The more colonies an imperialist possess, the bigger is its relevant (☆) mark.

Step2. Assimilation: movement of colonies toward the imperialist

Pursuing assimilation policy, the imperialist states tried to absorb their colonies and make them a part of themselves. More precisely, the imperialist states made their colonies to move toward themselves along different socio-political axis such as culture, language and religion. In the ICA, this process is modeled by moving all of the colonies toward the imperialist along different optimization axis. Figure (3) shows this movement. Considering a 2-dimensional optimization problem, in this figure the colony is absorbed by the imperialist in the culture and language axes. Then colony will get closer to the imperialist in these axes. Continuation of assimilation will cause all the colonies to be fully assimilated into the imperialist.

In the ICA, the assimilation policy is modeled by moving all the colonies toward the imperialist. This movement is shown in figure (3) in which a colony moves toward the imperialist by *x* units. The new position of colony is shown in a darker color. The direction of the movement is the vector from the colony to the imperialist.

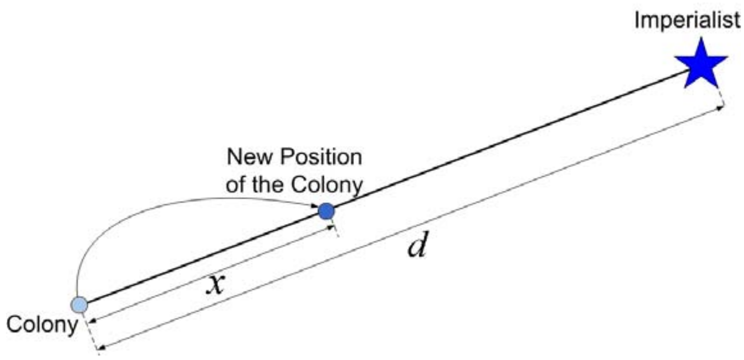


Figure (3):. Movement of colonies toward their relevant imperialist

In this figure x is a random variable with uniform (or any proper) distribution. Then

$$x \sim U(0, \beta * d) \qquad (5)$$

where β is a number greater than 1 and d is the distance between the colony and the imperialist state. β> 1 causes the colonies to get closer to the imperia list state from both sides.

Assimilating the colonies by the imperialist states did not result in direct movement of the colonies toward the imperialist. That is, the direction of movement is not necessarily the vector from colony to the imperialist. To model this fact and to increase the ability of searching more area around the imperialist, a random amount of deviation is added to the direction of movement. Figure (4) shows the new direction. In this figure θ is a parameter with uniform (or any proper) distribution. Then

$$\theta \sim U(-\gamma, \gamma) \qquad (6)$$

where γ is a parameter that adjusts the deviation from the original direction. Nevertheless the values of β and γ are arbitrary, in most of implementations a value of about 2 for β and about $\pi/4$ (Rad) for γ results in good convergence of countries to the global minimum [Atashpaz 2007] and [Atashpaz 2008].

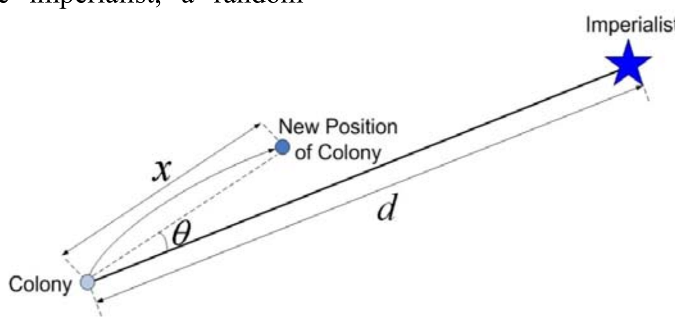


Figure (4): Movement of colonies toward their relevant imperialist in a randomly deviated direction.

Step 3. Revolution; a sudden change in socio-political characteristics of a country

Revolution is a fundamental change in power or organizational structures that takes place in a relatively short period of time. In the terminology of ICA, revolution causes a country to suddenly change its socio-political characteristics. That is, instead of being assimilated by an imperialist, the colony randomly changes its position in the socio-political axis. Figure (5) shows the revolution in Culture-Language axis. The revolution increases the exploration of the algorithm and prevents the early convergence of countries to local minimums. The revolution rate in the algorithm indicates the percentage of colonies in each colony which will randomly change their position. A very high value of revolution decreases the exploitation power of algorithm and can reduce its convergence rate. In general, the revolution rate is 0.3. That is 30 percent of colonies in the empires change their positions randomly [Atashpaz 2007] and [Atashpaz 2008].

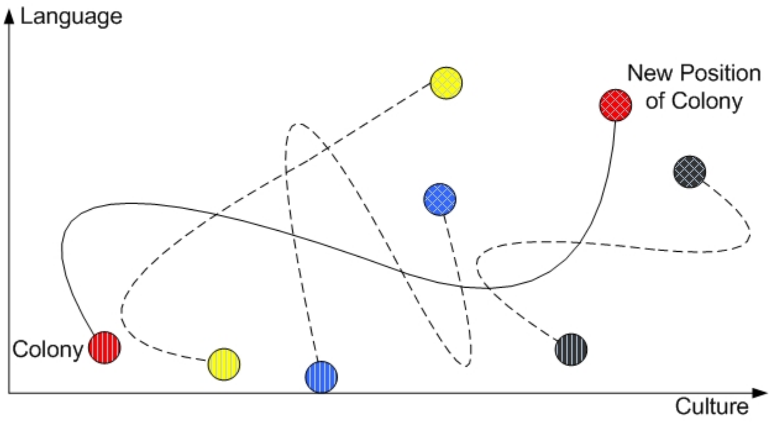
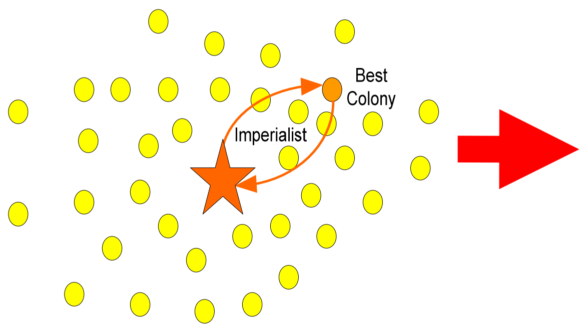


Figure (5): Revolution; a sudden change in socio-political characteristics of a country.

Step 4. Exchanging positions of the imperialist and a colony

While moving toward the imperialist, a colony might reach to a position with lower cost than the imperialist. In this case, the imperialist and the colony change their positions. Then the algorithm will continue by the imperialist in the new position and the colonies will be assimilated by the imperialist in its new position. Figure(6a) depicts the position exchange between a colony and the imperialist. In this figure the best colony of the empire is shown in a darker color. This colony has a lower cost than the imperialist. Figure (6b) shows the empire after exchanging the position of the imperialist and the colony [Atashpaz 2007] and [Atashpaz 2008].



Figure(6a): Exchanging the positions of a colony and the imperialist

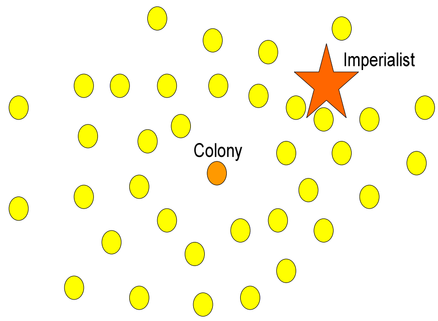


Figure (6b): The entire empire after position exchange

Step 5.Total power of an empire

Total power of an empire is mainly affected by the power of imperialist country. However the power of the colonies of an empire has an effect, albeit negligible, on the total power of that empire. This fact is modeled by defining the total cost of an empire by

$$T.C_n = cost(imperialist) + \varepsilon * mean\{\, cost(colonies\ of\ empire)\} \qquad (7)$$

Where $T.C_n$ is the total cost of the n[th] empire and ε is a positive small number. A little value for ε causes the total power of the empire to be determined by just the imperialist and increasing it will increase to the role of the colonies in determining the total power of an empire. The value of 0.1 for n has shown good results in most of the implementations [Atashpaz 2007] and [Atashpaz 2008].

Step 6. Imperialistic competition

All empires try to take the possession of colonies of other empires and control them. The imperialistic competition gradually brings about a decrease in the power of weaker empires and an increase in the power of more powerful ones. The imperialistic competition is modeled by just picking some (usually one) of the weakest colonies of the weakest empire and making a competition among all empires to possess these (this) colonies. Figure (7) shows a big picture of the modeled imperialistic competition. Based on their total power, in this competition, each of empires will have a likelihood of taking possession of the mentioned colonies. In other words, these colonies will not definitely be possessed by the most powerful empires, but these empires will be more likely to possess them.

To start the competition, first a colony of the weakest empire is chosen and then the possession probability of each empire is found. The possession probability P_P is proportionate to the total power of the empire. The normalized total cost of an empire is simply obtained by

$$N.T.C_n = T.C_n - max\{T.C_n\} \quad (8)$$

Where, $T.C_n$ and $N.T.C_n$ are the total cost and the normalized total cost of nth empire, respectively. Having the normalized total cost, the possession probability of each empire is given by

$$p_n = \left| \frac{N.T.C_n}{\sum_{i=1}^{N_{imp}} N.T.C_n} \right| \quad (9)$$

To divide the mentioned colonies among empires vector P is formed as following

$$P = [p_{p_1}, p_{p_2}, p_{p_3}, \dots, p_{p_{N_{imp}}}] \quad (10)$$

Then the vector R with the same size as P whose elements are uniformly distributed random numbers is created,

$$R = \left[r_1, r_2, r_3, \dots, r_{N_{imp}}\right], r_1, r_2, r_3, \dots, r_{N_{imp}} \sim U(0,1) \quad (11)$$

Then vector D is formed by subtracting R from P

$$D = P - R = \left[D_1, D_2, D_3, \dots, D_{N_{imp}}\right] = \left[p_1 - r_1, p_2 - r_2, p_3 - r_3, \dots, p_{N_{imp}} - r_{N_{imp}}\right] \quad (12)$$

Referring to vector D the mentioned colony (colonies) is handed to an empire whose relevant index in D is maximized.

The process of selecting an empire is similar to the roulette wheel process which is used in selecting parents in GA. But this method of selection is much faster than the conventional roulette wheel. Because it is not required to calculate the cumulative distribution function and the selection is based on only the values of probabilities. Hence, the process of selecting the empires can solely substitute the roulette wheel in GA and increase its execution speed [Atashpaz 2007] and [Atashpaz 2008].

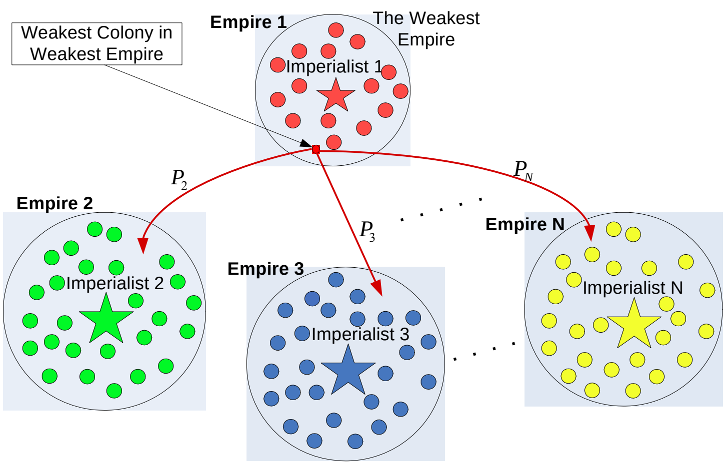


Figure (7): Imperialistic competition: The more powerful an empire is, the more likely it will possess the weakest colony of the weakest empire.

4. Conjugate Gradient Algorithm:

Conjugate gradient (CG) methods represent an important class of unconstrained optimization algorithm. The main advantages of the CG methods are its low memory requirements, its convergence speed and its poses a quadratic termination property in which the method is able to locate the minimize of quadratic function in a known finite number of iterations [Andrei 2007].

A nonlinear conjugate gradient method generates a sequence x_k, k is integer number, $k \geq 0$. Starting from an initial point x_0, the value of x_k calculate by the following equation:

$$x_{k+1} = x_k + \lambda_k d_k \qquad (13)$$

where the positive step size $\lambda_k > 0$ is obtained by a line search, and the directions d_k are generated as:

$$d_k = -g_k + \beta d_{k-1} \qquad (14)$$

where $d_0 = -g_0$, the value of β is determine according to the algorithm of Conjugate Gradient (CG), and its known as a conjugate gradient parameter, $s_k = x_{k+1} - x_k$ and $g_k = \nabla f(x_k) = f'(x_k)$, consider $\|.\|$ is the Euclidean norm and $y_k = g_{k+1} - g_k$. The termination conditions for the conjugate gradient line search are often based on some version of the Wolfe conditions. The standard Wolfe conditions:

$$f(x_k + \lambda_k d_k) - f(x_k) \leq \rho \lambda_k g_k^T d_k, \qquad (15)$$

$$g(x_k + \lambda_k d_k)^T d_k \geq \sigma g_k^T d_k, \qquad (16)$$

where d_k is a descent search direction and $0 < \rho \leq \sigma < 1$, where β_k is defined by one of the following formulas:

$$\beta_k^{(HS)} = \frac{y_k^T g_{k+1}}{y_k^T d_k} \text{ (Hestenese and Stiefel)} \qquad (17)$$

$$\beta_k^{(FR)} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \text{ (Fletcher and Reeves)} \qquad (18)$$

$$\beta_k^{(PRP)} = \frac{y_k^T g_{k+1}}{g_k^T g_k} \text{ (Polak – Ribiere ; and Polyak)} \qquad (19)$$

$$\beta_k^{(CD)} = -\frac{g_{k+1}^T g_{k+1}}{g_k^T d_k} \text{ (Conjugatedescent)} \qquad (20)$$

$$\beta_k^{(LS)} = -\frac{y_k^T g_{k+1}}{g_k^T d_k} \text{ (Liu and Stoery)} \qquad (21)$$

$$\beta_k^{(DY)} = \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} \text{ (Dai and Yuan)} \qquad (22)$$

Outlines of the CG methods:

Step1: Start with an arbitrary initial point $\boldsymbol{X_1}$.
Step 2: Set the first search direction $d_1 = -g_1 = -\nabla f(X_1)$
Step 3: Find the point $\mathbf{X}_2$ according to the relation

$$\boldsymbol{X_2 = X_1 + \lambda_1 d_1}$$

where $\boldsymbol{\lambda_1}$ is the optimal step length in the direction $\boldsymbol{d_1}$. Set $k = 2$ and go to the next step.
Step 4 Find $g_k = -\nabla f(X_k)$, and set

$$\boldsymbol{d_k = -g_k + \beta d_{k-1}}$$

where β is the conjugancy coefficient equal to $\frac{\|g_k\|^2}{\|g_{k-1}\|^2}$.
Step5: Compute the optimum step length $\boldsymbol{\lambda_k}$in the direction d_k, and find the new point

$$\boldsymbol{X_{k+1} = X_k + \lambda_k d_k}$$

Step6: Test for the optimality of the point$\boldsymbol{X_{k+1}}$. If $\boldsymbol{X_{k+1}}$ is optimum, stop the process. Otherwise, set the value of $k = k + 1$ and go to step 4 [Andrei 2007].

5. Hybrid ICA with CG Algorithm:

In this Section, we proposed a new hybrid ICA with conjugate gradient algorithm (CG), called ICA-CG algorithm. In the proposed hybrid algorithm, the process in every iteration is divided into two stages. In the first stage, we use the randomly, rapidity and wholeness of the imperialist competitive Algorithm. In the second stage we used FR-CG algorithm to move imperialist countries.

The important property in evolutionary algorithms is to find the best solution without needing to calculate the derivatives, while the classical methods of optimization especially conjugate gradient method, we need to calculate the derivatives because the solution moves in the negative gradient direction ($d_k = -g_k$) and therefore, we suggest here, to further improve the efficiency of classical ICA is to use calculations derivatives instead of using derivatives to avoid the loss of one of the most important properties of evolutionary algorithms.

The gradient direction g_k can be substituted by

$$\frac{f(x + \Delta x) - f(x)}{\Delta x} \qquad (23)$$

where Δx is a small positive perturbation (i.e. 1.00E-05).

The revolution rate (P_r) in the algorithm indicates the percentage of colonies in each colony which will randomly change their position. A very high value of revolution rate decreases the exploitation power of algorithm and can reduce its convergence rate. In our algorithms the revolution rate is not constant number as original ICA, but its depending on the number of variables (N_{var}). The experimental results shows that the best revolution rate is shown in Table (1).

Table (1):represent the best revolution rate for ICA-CG

N_{var}	Revolution rate
1-10	0.001-0.01
11-20	0.01-0.05
21-50	0.05-0.08
51-100	0.08-0.1
101-1000	0.1-0.3
more than 1000	0.3-0.4

Moreover, It is worth stressing again that no explicit calculations of the true gradient are performed, and the main purpose of this work is to provide an improved ICA method. In this algorithm when CG approach make bad imperialist, we back to the old imperialist before CG approach. Figure(8) shows the flowchart of the proposed algorithm .The Procedures of the ICA-CG is presented as below:

Procedure ICA-CG

Step 1: Initializing parameters;

Step 2:

2.1 Define the optimization problem;

2.2:Generate some random countries;

2.3:Select the most powerful countries as empires;

2.4:Randomly allocate remain countries to different empires equality;

Step 3:Decade loop k=k+1 % ICA-CG operators%

Step 4:For i=1,2,…,N_{imp} do %**Imperialist Competitive Algorithm**%

4.1:Assimilate colonies toward their imperialist;

4.2:Countries revolution;

4.3:Exchange imperialist with best colony if is necessary;

4.4:Calculate total cost of empires;

4.5:Imperialistic competition;

4.6:Eliminate the powerless empires;

Step 5:For i=1,2,…,N_{imp} do %Conjugate Gradient%

5.1: If k=1 then $d_k=-g_k$, Otherwise $d_k=-g_k+\beta_{k-1}d_{k-1}$

5.2:Calculate the new imperialist as $x_{k+1}=x_k+\lambda d_k$, where λ is a step size which is equal to 0.001

Step 6: Terminating Criterion Control; Repeat Steps 3-6 until a terminating criterion is satisfied;

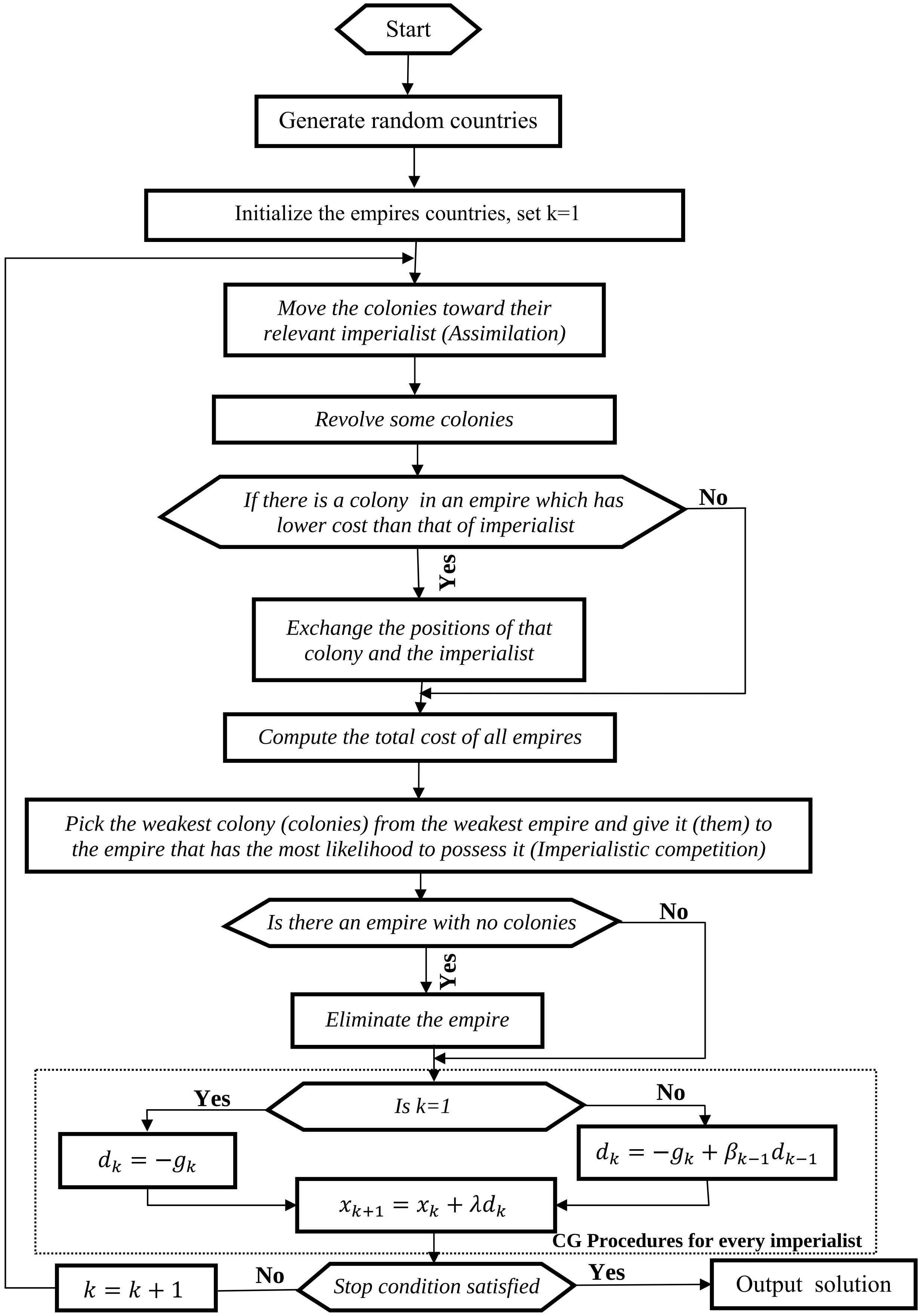


Figure (8): Flowchart of the proposed ICA-CG algorithm

5. Numerical Results

The proposed ICA-CG is tested using 5 benchmark functions [Andrei 2008]. For comparison, GA, PSO and ABC [Karaboga 2007] are also executed on these 5 functions. The parameter settings of ICA-CG algorithm are described as follows: Assimilation coefficients are set to 2.0, Revolutionary rates are set as table (1). Table (2) shows the details of test functions . The algorithm is conducted 20 runs for each test function.

Table (2): Benchmark functions (F1-F5)

Benchmarks	Function	Range
F1(Sphere)	$\sum_{i=1}^{n} x_i^2$	$[-5.12,5.12]$
F2(Rastrigin)	$\sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12,5.12]$
F3(Griewank)	$1 + \sum_{i=1}^{n} (\frac{x_i^2}{4000}) - \prod_{i=1}^{n} \left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$	$[-600,600]$
F4(Ackly)	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 - e$	$[-30,30]$
F5(SumSquares)	$\sum_{i=1}^{n} i^2 x_i^2$	$[-1,1]$

In Table (3), the performance of ICA-CG algorithm is compared with GA,ICA, PSO and ABC [13] for high dimensional problem .

Table (3): Comparing the Performance ICA-CG with GA , PSO and ABC (high dimensions)

F.	N_{var}	Pop	Gen	P_r	GA	ICA	PSO	ABC	9
F1	100	500	1000	0.1	9.70E-01	1.71E+02	1.99E+00	8.59E−05	**2.02E-16**
	500	600	1500	0.25	2.02E+01	2.94E+03	3.90E+03	2.26E+02	**6.49E-10**
	1000	800	2000	0.3	6.40E+01	6.72E+03	8.00E+03	1.46E+03	**5.29E-09**
F2	100	500	1000	0.1	5.81E+01	9.17E+02	6.05E+02	5.39E+01	**2.46E-05**
	500	600	1500	0.25	1.01E+03	7.14E+03	8.59E+03	1.93E+03	**1.31E+02**
	1000	800	2000	0.3	2.97E+03	1.54E+04	1.74E+04	6.05E+03	**5.08E+02**
F3	100	500	1000	0.1	1.32E-02	7.56E+02	5.24E−01	9.04E−03	**3.95E-08**
	500	600	1500	0.25	6.19E+01	1.03E+04	5.10E+01	9.48E+02	**2.20E-01**
	1000	800	2000	0.3	1.14E+03	2.38E+04	2.98E+02	4.86E+03	**1.50E+00**
F4	100	500	1000	0.1	2.74E+00	1.90E+00	3.37E+00	2.12E+00	**1.24E-07**
	500	600	1500	0.25	9.01E+01	3.49E+01	2.09E+01	1.49E+01	**1.96E-05**
	1000	800	2000	0.3	7.80E+03	7.26E+03	1.74E+04	1.77E+01	**2.80E-05**
F5	100	500	1000	0.1	8.69E+01	1.75E+04	1.39E+02	1.11E−03	**4.35E-08**
	500	600	1500	0.25	9.02E+02	7.41E+06	1.12E+07	5.23E+05	**9.96E+00**
	1000	800	2000	0.3	3.39E+04	7.46E+07	9.50E+07	1.84E+08	**9.77E+02**

Conclusion

This paper proposes hybrid algorithm consisting of ICA and CG. The performance of this algorithm is evaluated using various test functions. These functions are set of well-known multi-dimensional benchmark functions. The simulations indicate that the proposed algorithm has outstanding performance in speed of convergence and precision of the solution for global optimization. This is meaning that it has the capability to come up with non-differentiable objective functions with a multitude number of local optima through reasonable time limit. The results show the efficiency and capabilities of the new hybrid algorithm in finding the optimum. Also, the performance of such algorithm is better than other algorithms such as GA, ICA, PSO and ABC. Indeed, the performance achieved of this study is quite satisfactory and promising for all test functions.

References

Andrei, N.(2007) .Scaled conjugate gradient algorithms for unconstrained optimization.Compute. Optima. Appl. 38. pp.401-416.

Andrei, N. (2008). An Unconstrained Optimization Test Function Collection. Adv. Model. Optimize. Vol. 10, pp. 147-161.

Atashpaz-Gargari, E., Lucas, C.(2007). Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition. In: IEEE Congress on Evolutionary Computation (CEC 2007). pp. 4661–4667

Atashpaz-Gargari, E., Hashemzadeh, F., Rajabioun, R., Lucas, C.(2008). Colonial Competitive Algorithm: A novel approach for PID controller design in MIMO distillation column process. International Journal of Intelligent Computing and Cybernetics (IJICC) 1(3).pp.337–355.

Bahrami, H., Feaz, K., Abdechiri, M. (2010), Imperialist Competitive Algorithm using Chaos Theory for Optimization (CICA). In: Proceedings of the 12th International Conference on Computer Modelling and Simulation .

Bahrami, H., Feaz, K., Abdechiri, M.(2010). Adaptive Imperialist Competitive Algorithm (AICA).In: Proceedings of The 9th IEEE International Conference on Cognitive Informatics. ICCI 2010.

Ghodrati , A., Malakooti, V., Soleimani, M.(2012). A Hybrid ICA/PSO Algorithm by Adding Independent Countries for Large Scale Global Optimization. In: Proceedings of The Forth Asian Conference of Intelligent Information and Database Systems . ACIIDS 2012. Part III. LNAI 7198, pp. 99–108**7.**

Goldberg, D.E.(1989). Genetic Algorithms in Search. Optimization & Machine Learning. Addison-Wesley. Reading.

Holland, J.H.(1990). ECHO: Explorations of Evolution in a Miniature World. In: Farmer, J.D.,Doyne, J. (eds.) Proceedings of the Second Conference on Artificial Life .

Karaboga, D., Basturk, B.(2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39(3). pp.459–471

Kennedy, J., Eberhart, R.C.(1995). Particle Swarm Optimization. Proceedings of IEEE. pp 1942–1948 .

Patel, R.V. (2000). System Optimization. Dept. of Electrical and Computer Engineering University of Western Ontario London. Ontario. Canada.

Ramezani, F.,Lotfi , S., Soltani, M.A.(2012). A Hybrid Evolutionary Imperialist Competitive Algorithm (HEICA) . In: Proceedings of The Forth Asian Conference of Intelligent Information and Database Systems. ACIIDS 2012.Part I. LNAI 7196. pp. 359–368

Sarimveis, H., Nikolakopoulos, A.(2005). A Life Up Evolutionary Algorithm for Solving Nonlinear Constrained Optimization Problems. Computer & Operation Research 32(6). pp.1499–1514.

Srinivasan, D., Seow, T.H.(2003). Evolutionary Computation. In: CEC 2003.December 8-12. vol. 4. pp. 2292–2297. Canberra. Australia.

Soltani, M.A., Lotfi ,S.,Ramezani, F.(2012). Quad Countries Algorithm (QCA). In: Proceedings of The Forth Asian Conference of Intelligent Information and Database Systems . ACIIDS 2012. Part III. LNAI 7198. pp. 119–129.

Sun, L. and Zhang, W. (2006). An Acceleration Micro Genetic Algorithm for Numerical Optimization. Lecture Notes in computer Science. Vol. 4247.pp 277-283.

Mühlenbein, H., Schomisch, M., Born, J.(1991). The Parallel Genetic Algorithm as Function Optimizer. In: Proceedings of The Forth International Conference on Genetic Algorithms, pp. 270–278. University of California. San diego.

Zhang, Y., Wang, Y., Peng, C.(2009).Improved Imperialist Competitive Algorithm for Constrained Optimization. International Forum on Computer Science-Technology and Applications.

خوارزمية تنافسية امبريالية مقترحة مهجنة مع خوارزمية الاتجاه المترافق لحل مسائل الامثلية ذات القياس الكبير

الخلاصة :

في هذا البحث تم اقتراح خوارزمية مهجنة جديدة تدعى اختصاراً خوارزمية $ICA-CG$، وذلك بربط الافكار التطورية للخوارزمية التنافسية الامبريالية مع افكار الامثلية التقليدية متمثلة في خوارزمية الاتجاه المترافق لحل مسائل الأمثلية الكبيرة، في هذه الخوارزمية المقترحة تقسم العملية في كل تكرار إلى مرحلتين، في المرحلة الاولى تم استخدام العشوائية والسرعة لخوارزمية الـ ICA، وفي المرحلة الثانية تم استخدام واحدة من تقنيات الامثلية التقليدية المعروفة والتي تدعى خوارزمية $FR-CG$ لتحريك البلدان الامبريالية. وقد أظهرت التجارب العددية لخمسة مسائل اختبار قياسية تفوق خوارزمية $ICA-CG$ الجديدة في حل مسائل الأمثلية الكبيرة مقارنة بالخوارزميات التقليدية كالخوارزمية الجينية GA وخوارزمية أمثلة أسراب الطيور PSO وخوارزمية مستعمرات النحل ABC من ناحية سرعة التقارب ونوعية الحلول المكتسبة.