# Quad Countries Algorithm (QCA)

M.A. Soltani-Sarvestani[1,*], Shahriar Lotfi [2], and Fatemeh Ramezani[3]

[1] Computer Engineering Departmant, College of Nabi Akram, Tabriz, Iran
[2] Computer Science Departmant, University of Tabriz, Tabriz, Iran
[3] Computer Engineering Departmant, College of Nabi Akram, Tabriz, Iran
shahriar_lotfi@tabrizu.ac.ir,
{soltani_mohammadamin,f60_ramezani}@yahoo.com

**Abstract.** This paper introduces an improved evolutionary algorithm based on the *Imperialist Competitive Algorithm* (ICA), called *Quad Countries Algorithm* (QCA). The Imperialist Competitive Algorithm is inspired by socio-political process of imperialistic competition in the real world and has shown its reliable performance in optimization problems. In the ICA, the countries are classified into two groups: Imperialists and Colonies. However, in the QCA, two other kinds of countries including Independent and Seeking Independence are added to the countries collection. In the ICA also the Imperialists' positions are fixed, while in the QCA Imperialists may move. The proposed algorithm was tested by well-known benchmarks, and the compared results of the QCA with results of ICA, GA [12], PSO [12], PS-EA [12] and ABC [11] show that the QCA has better performance than all mentioned algorithms. Among them, the QCA, ABC and PSO have better performance respectively in 50%, 41.66% and 8.33% of all cases.

**Keywords:** Optimization, Imperialist Competitive Algorithm (ICA), Independent countries, countries Seeking Independence and Quad Countries Algorithm (QCA).

## 1 Introduction

Evolutionary Algorithms (EA) [1, 2] are algorithms that inspire from nature and have many applications to solve NP problems in various fields of science. Some of the proposed Evolutionary Algorithms for optimization problems are: the Genetic Algorithm (GA) [2 ,3 ,4], which at first proposed by Holland, in 1962 [3] , Particle Swarm Optimization algorithm (PSO) [5] first proposed by Kennedy and Eberhart [5], in 1995. In 2007, Atashpaz and Lucas proposed an algorithm as Imperialist Competitive Algorithm (ICA) [6, 7], that has inspired from a socio-human phenomenon. Since 2007 attempts were performed in order to increase the efficiency of the ICA. Zhang, Wang and Peng proposed an approach based on the concept of small probability perturbation to enhance the movement of Colonies to Imperialist, in 2009 [8]. In 2010, Faez, Bahrami and Abdechiri, proposed a new method using the

[*] Corresponding author.

J.-S. Pan, S.-M. Chen, N.T. Nguyen (Eds.): ACIIDS 2012, Part III, LNAI 7198, pp. 119–129, 2012.

chaos theory to adjust the angle of Colonies movement toward the Imperialists' position (*Imperialist Competitive Algorithm using Chaos Theory for Optimization : CICA*)[9], 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(Adaptive Imperialist Competitive Algorithm: AICA) [10].

In the Imperialist Competitive Algorithm (ICA), there are only two different types of countries, Imperialists and Colonies, which Imperialists absorb their Colonies. While in the real world, there are some Independent countries which are neither Imperialists, nor Colonies. In the ICA, only the Colonies' movements toward Imperialists are considered while in the real world, each Imperialist moves in order to promote its political and cultural position. In the Quad Countries Algorithm (QCA), countries are divided into four categories: Imperialist, Colony, Seeking Independence and Independent that each category has its special movement compared to the others. In the Quad Countries Algorithm, like in the real world, an Imperialist will move if reaches to a better position compared to its current position.

The following part of this paper is arranged as follows. Section two describes a brief description of Imperialist Competitive Algorithm. Section three will explain the proposed algorithm. In section four, performance of algorithms will be analyzed and evaluated. In the section five, a conclusion will be presented.

## 2 The Imperialist Competitive Algorithm (ICA)

Imperialist Competitive Algorithm (ICA) at the first time proposed by Atashpaz and Lucas, in 2007 [6]. The ICA is a new evolutionary algorithm in the Evolutionary Computation (EC) field based on the human's socio-political evolution. The algorithm starts with an initial random population called countries, then some of best countries in the population select to be the Imperialists and the rest of them form the Colonies of these Imperialists. The number of initial population is $N_{pop}$ including $N_{col}$ Colonies and $N_{imp}$ Imperialist. The Colonies divide among Imperialists. The initial number of Colonies of an Imperialist will be $NC_n$. The $NC_n$ is initial number of Colonies of $n^{th}$ Imperialist. To distribute the Colonies among Imperialists, according to the number of $NC_n$, they are randomly selected and assigned to the $n^{th}$ Imperialist. The Imperialist countries absorb the Colonies towards themselves using the Absorption policy. The Absorption policy makes the main core of this algorithm and causes the countries move towards to their minimum optima. This policy is shown in Fig.1. In the Absorption policy, the Colony moves towards the Imperialist by $x$ unit. The direction of movement is the vector from Colony to Imperialist, as shown in Fig.1. In this figure, the distance between the Imperialist and Colony is shown by $d$, and $x$ is a random variable with uniform distribution. In the ICA, in order to search different points around the Imperialist, a random amount of deviation is added to the direction of Colony movement towards the Imperialist. In Fig.1, this deflection angle is shown as $\Theta$, which is chosen randomly and with a uniform distribution.

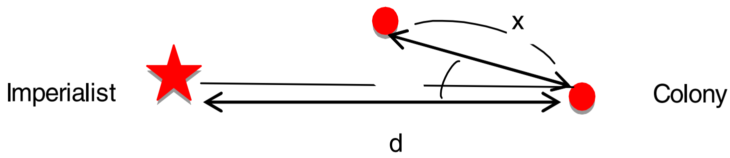



**Fig. 1.** Moving Colonies toward their Imperialist [6]

While Colonies moving toward the imperialist countries, a colony may reach to a better position than its imperialist, so the Colony position exchanges with position of the Imperialist.

In the ICA, the imperialistic competition has an important role. During the imperialistic competition, the weak Imperialist will lose their power and their colonies. After a while all the Imperialists except the most powerful one, will be collapse and all the Colonies will be under the control of this unique Imperialist.

## 3 Quad Countries Algorithm (QCA)

In this paper, a new Imperialist Competitive Algorithm is proposed which is called Quad Countries Algorithm that two new categories of countries are added to the collection of countries; Independent and Seeking Independence countries. In addition, in the new algorithm Imperialists can also move like the other countries.

### 3.1 Independent Country

In the real world, there are permanently countries which have been neither Colonies, nor Colonial. These Countries may perform any movements in order to take their advantages and try to improve their current situation. In the proposed algorithm, some countries are defined as Independent countries which explore search space randomly. As an illustration in figure 2, if during the search process, an Independent country achieves a better position compared to an Imperialist, they will definitely exchange their positions. The Independent countries change to a new Imperialist and will be the owner of old Imperialist’s Colonies and instead of the Imperialist will changes to an Independent country and will start to explore the search space like these kinds of countries.

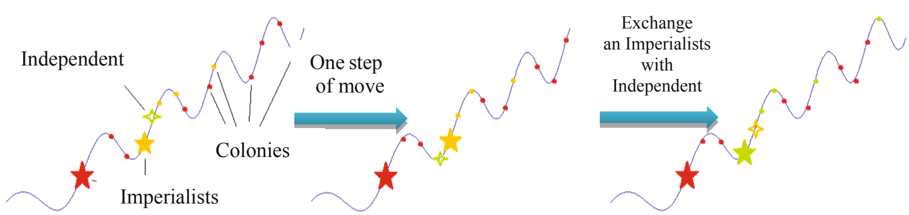
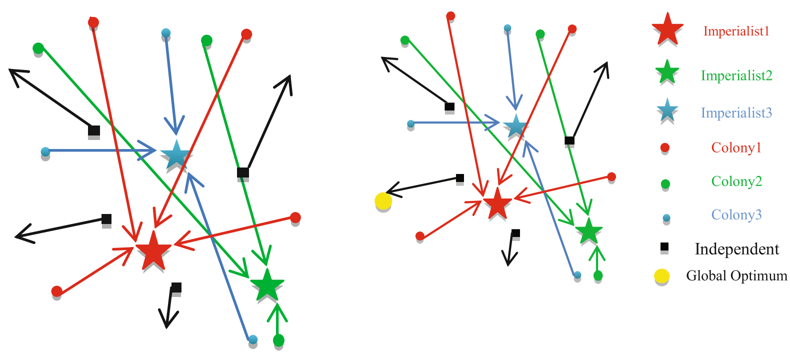



**Fig. 2.** Replace an Imperialist with an Independent

### 3.2 Seeking Independence Country

Seeking Independence Countries are countries which have challenges to the Imperialists and try to be away from them. In the main ICA, the only movement is the Colonies' movements toward Imperialists and in fact, there is only Absorption policy. While by defining the Seeking Independence Countries in proposed algorithm, there is also Repulsion policy versus Absorption policy.

Fig.3 illustrates the Repulsion Policy. As can be seen in Fig.3.a, there is only the Absorption policy that matches with the ICA. As it shows, the only use of applying Absorption policy causes that countries' positions to gets closer to each other and their surrounded space will decrease gradually and the global optima might be lost. In Fig.3.a the algorithm is converge to a local optimum. Fig.3.b illustrates the process of the proposed algorithm. The black Squares represent the Seeking Independence Countries and as can be seen, these countries can steer the search process to a direction which the other countries don't cover. It shows that, using Absorption and Repulsion policy together, will leads to better coverage of search space.




(a) Absorption policy (b) Absorption and Repulsion policy

**Fig. 3.** Different movement policy

To apply the Repulsion policy in the QCA, first the sum of differences between the Seeking Independence Countries and the Imperialists positions is calculated as a vector like (1) named *Center*, that is a $1 \times N$ vector.

$$Center_i = \sum_{j=1}^{N_{imp}} (a_i - p_{ji}), \quad i = 1,\ 2,\ \ldots,\ N \tag{1}$$

where $Center_i$ is the sum of $i^{th}$ component of all Imperialists, $p_{ji}$ is $i^{th}$ component of $j^{th}$ Imperialist, $a_i$ is $i^{th}$ component of Seeking Independence Country and $N$ indicates the problem dimensions. Then the Seeking Independence Countries will move in the direction of obtained vector as (2).

$$D = \delta \times Center, \quad \delta \in (0, \ 1) \tag{2}$$

where δ is relocation factor and *D* is relocation vector that its components, peer to peer sum to the Seeking Independence Country's components and obtains new position of the Seeking Independence Country.

### 3.3 Imperialists Movement

In the real world, all countries including Imperialists perform ongoing efforts to improve their current situation. While in the main ICA, Imperialists never move and this fixed situation sometimes leads to lose global optima or prevent to achieve better consequences. Fig.4 could be a final state of running the ICA, when only one Imperialist has remained. Since in the ICA, Imperialists have no motion, result 1 is the answer that the ICA returns. In the proposed approach, a movement, opposite to the central of gravity of its colonies is assumed for Imperialists, and the cost of this hypothetical position will be calculated. If the cost of the hypothetical position is less than the cost of the current one, the Imperialist will move to the hypothetical position, otherwise the Imperialist will not move. As can be seen in Fig.4, using this method leads to result 2 which is a better result than 1.

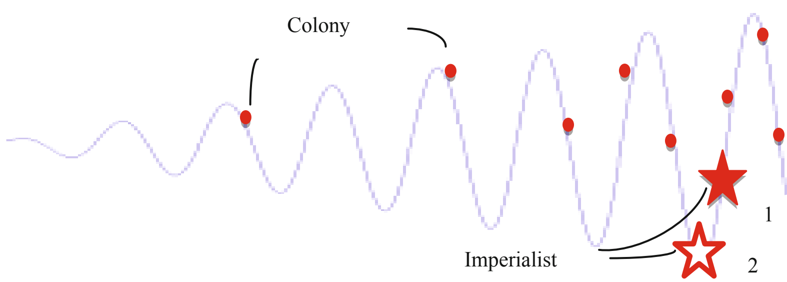



**Fig. 4.** A final state of ICA and QCA. Result 1 may be a final state of ICA and Result 2 may be a final state of QCA.

The movement of Imperialist is shown in equation (3).

$$\begin{aligned}
&imp_dir_i = \sum_{j=1}^{Ncol} colony_{j,i} \\
&New_position_i = imp_i - imp + dir_i \times ieta \times rand() \\
&if\,(\cos t(New_position_i) < \cos t(perevious_position_i)) \\
&Then \qquad perevious_position_i = New_position_i
\end{aligned} \tag{3}$$

$Imp\_dir_i$ is the Imperialist direction of movement of $i^{th}$ Imperialist, $Colony_{j,i}$ is the $j^{th}$ colony of $i^{th}$ Imperialist, ieta is a positive value less than 1, $New\_position_i$ is hypothetical position for $i^{th}$ Imperialist, Cost() is Cost function, and $Perevious\_position_i$ is the previous position of $i^{th}$ Imperialist.

## 4 Evaluation and Experimental Results

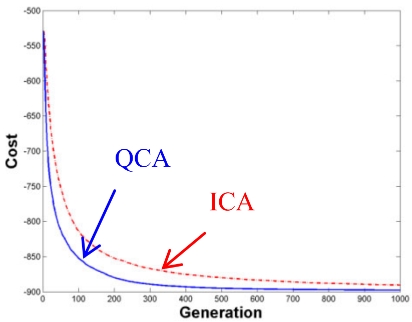
In this paper, a new algorithm based on the Imperialist Competitive Algorithm (ICA), called Quad Countries Algorithm (QCA) is introduced and was applied to some well-known benchmarks in order to verify its performance, and compare to ICA. These benchmarks functions are presented in table1.The QCA parameters set as follow: *population*=125, *ieta*=0.005, *eta*=0.01, and $\delta$=0.01.

Each algorithm runs 100 times, and The observed results of applying the algorithms on the benchmarks are shown in table 3. *Griewank Inverse* is a hill-like function and its global optima are located on the corner of search space. Fig.5 averagely, illustrates the graph of stability and convergence of *Griewank Inverse* with 10 and 50 dimensions. It can be seen from Fig 5 that the quality of the results, the convergence of the QCA is faster than the ICA. Figures 5.c and 5.d illustrates stability graph of *Griewank Inverse* with 10 and 50 dimensions.

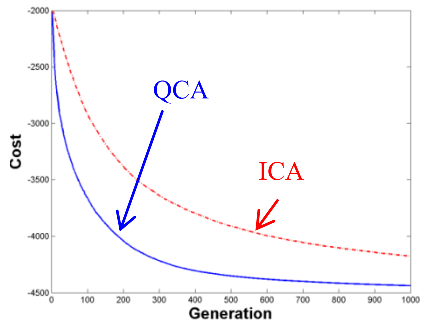
**Table 1.** Benchmarks for simulation

| Bnechmarks | Mathematical Reprasantation | Range |
|---|---|---|
| Ackley | $f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos 2\pi x_i\right) + 20 - e$ | [-32.768, 32.768] |
| Griewank | $f(x) = \frac{1}{4000} \times \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | [-600,600] |
| Rastrigin | $f(x) = \sum_{i=1}^{D}\left(x_i^2 - 10\times\cos\left(2\pi x_i\right)\right) + 10D$ | [-15,15] |
| Sphere | $f(x) = \sum_{i=1}^{D}\left(x_i^2\right)$ | [-600,600] |
| Rosenbrock | $f(x) = \sum_{i=1}^{D-1}\left(100\times\left(x_{i-1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right)$ | [-15,15] |
| Griewank Inverse | $f(x) = -\frac{1}{4000} \times \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | [-600,600] |
| Rastrigin Inverse | $f(x) = -\sum_{i=1}^{D}\left(x_i^2 - 10\times\cos\left(2\pi x_i\right) + 10\right)$ | [-600,600] |
| Sphere Inverse | $f(x) = -\sum_{i=1}^{D}\left(x_i^2\right)$ | [-600,600] |

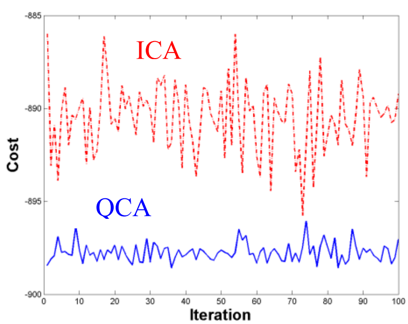
In the other comparison, the results are compared to Genetic Algorithm (GA), Particle Swarm Optimization(PSO), PS-EA and Artificial Bee Colony(ABC) in table 4. As can be seen, the results of the proposed algorithm are better than GA and PS-EA in 100 percent of all cases. But in the comparison with the QCA, the ABC and PSO, in 50 percent of cases the QCA has better performance by comparison with ABC and PSO. The ABC and PSO are 41.66 and 8.33 percent of all cases respectively, which are shown better performance.



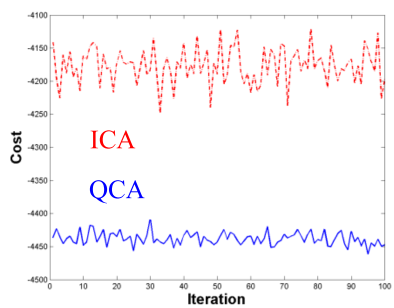

(a) the average of convergence of 100 iterations up to 1000 generation with 10 dimensions




(b) the average of convergence of 100 iterations up to 1000 generation with 50 dimensions




(c) Stability graph of 10 dimensions




(d) Stability graph of 50 dimensions

**Fig. 5.** The convergence and stability graphs of ICA and QCA on *Griewank Inverse*

**Table 2.** The result of applying benchmark on the QCA and the ICA with 2, 10, 30 and 50 dimensions

| Benchmark | Alg. DIM. | Optimum | QCA | | | ICA | | | Imp. |
|---|---|---|---|---|---|---|---|---|---|
| | | | Best result | Results Average | SD | Best result | Results Average | SD | |
| Sphere | 2 | 0 | 1.6384 E -26 | 7.4682 E -20 | 2.7799 E -19 | 2.0568 E -20 | 1.3710 F -10 | 1.1761 E -9 | ≈ 100% |
| | 10 | 0 | 4.6801 E -15 | 1.8719 E -11 | 3.9881 E -11 | 2.5493 E -12 | 3.0484 E -8 | 6.4450 E -5 | 99.94% |
| | 30 | 0 | 2.5590 E -9 | 7.1833 E -7 | 2.2583 E -6 | 1.0972 E -6 | 3.2491 E -5 | 3.6956 E -5 | 98.37% |
| | 50 | 0 | 7.3234 E -7 | 3.9662 E -5 | 1.0098 E -4 | 2.6172 E -4 | 0.0031 | 0.003 | 99.07% |
| Sphere Inv | 2 | -7.20 E +5 | -7.2 E +5 | -7.2000 E +5 | 0.2526 | -7.2000 E +5 | -7.1998 E +5 | 14.8687 | *0.003%* |
| | 10 | -3.60 E +6 | -3.5995 E +6 | -3.5983 E +6 | 783.7695 | -3.5821 E +6 | -3.5689 E +6 | 6.2142 E +3 | 0.82% |
| | 30 | -1.08 E +7 | -1.0761 E +7 | -1.0734 E +7 | 1.4222 E +4 | -1.0506 E +7 | -1.0358 E +7 | 5.4485 E +4 | 3.63% |
| | 50 | -1.80 E +7 | -1.7866 E +7 | -1.7755 E +7 | 4.0419 E +4 | -1.6950 E +7 | -1.6706 E +7 | 9.4520 E +4 | 6.29% |
| Rastrigin | 2 | 0 | 0 | 0 | 0 | 0 | 1.1358 E -13 | 6.6520 E -13 | 100% |
| | 10 | 0 | 0 | 1.3269 E -14 | 4.0851 E -14 | 4.4640 E -12 | 5.5944 E -9 | 1.5154 E -8 | 99.99% |
| | 30 | 0 | 3.6981 E -10 | 1.4274 E -8 | 2.4467 E -8 | 1.6195 E -4 | 0.3899 | 0.5083 | ≈ 100% |
| | 50 | 0 | 7.5566 E -7 | 0.0599 | 0.2362 | 1.0452 | 5.3211 | 1.7154 | 99.62% |
| Rastrigin Inv | 2 | -7.20 E +5 | -7.2000 E +5 | -7.2000 E +5 | 0.3104 | -7.2000 E +5 | -7.1999 E +5 | 13.0936 | 0.002% |
| | 10 | -3.60 E +6 | -3.5995 E +6 | -3.5983 E +6 | 906.4164 | -3.5861 E +6 | -3.5692 E +6 | 6.4272 E +3 | 0.82% |
| | 30 | -1.08 E +7 | -1.0767 E +7 | -1.0732 E +7 | 1.3192 E +4 | -1.0486 E +7 | -1.0348 E +7 | 4.7617 E +4 | 3.71% |
| | 50 | -1.80 E +7 | -1.7830 E +7 | -1.7757 E +7 | 3.4901 E +4 | -1.7019 E +7 | -1.6707 E +7 | 1.0234 E +5 | 6.28% |
| Griewank | 2 | 0 | 0 | 0 | 0 | 0 | 3.7356 E-13 | 2.7586 E-12 | 100% |
| | 10 | 0 | 8.9106 E -13 | 9.3103 E -9 | 2.2949 E -8 | 01.4433 E -9 | 6.8886 E -6 | 1.9415 E -5 | 99.86% |
| | 30 | 0 | 4.8241 E -4 | 0.0144 | 0.0220 | 0.0040 | 0.0721 | 0.0522 | 80.03% |
| | 50 | 0 | 0.0747 | 0.3832 | 37.9352 | 0.1402 | 0.4227 | 41.8484 | 17.2% |
| Griewank Inv | 2 | -180.0121 | -179.0827 | -178.9388 | 0.0877 | -179.0774 | -178.8674 | 0.0832 | 0.04% |
| | 10 | -901 | -898.5777 | -897.6309 | 0.5023 | -893.9284 | -890.7051 | 1.6002 | 0.79% |
| | 30 | -2701 | -2688 | -2681 | 3.6252 | -2620 | -2589 | 10.4053 | 3.6% |
| | 50 | -4501 | -4460 | -4439 | 9.2071 | -4255 | -4178 | 28.4462 | 6.28% |
| Ackley | 2 | 0 | 8.8818 E -16 | 8.4754 E -13 | 2.0295 E -12 | 3.4195 E -13 | 5.2040 E -8 | 4.5546 E -7 | 99.99% |
| | 10 | 0 | 1.2632 E -9 | 2.5552 E -8 | 4.8522 E -8 | 1.4476 E -7 | 2.4681 E -6 | 5.4074 E -6 | 98.99% |
| | 30 | 0 | 1.0459 E -6 | 4.3273 E -6 | 2.5904 E -6 | 3.9508 E -5 | 1.7145 E -4 | 1.0189 E -4 | 97.54% |
| | 50 | 0 | 3.7308 E -5 | 9.9126 E -5 | 4.1411 E -5 | 4.5648 E -4 | 0.0014 | 7.0191 E -4 | 93.54% |
| Schwefel | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Both algorithms are run 100 times, and indicate the results that are obtained after 1,000 cycle with a population having 125 individuals

**Table 3.** The results of GA, PSO, PS-EA, ABC, ICA and QCA

| BENCH MARK | Alg DIM | GA[12] | | PSO[12] | | PS-EA[12] | | ABC[11] | | ICA | | QCA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| Griewank | 10 | 0.0502 | 0.0295 | 0.0794 | 0.03345 | 0.22237 | 0.0781 | 0.00087 | 0.00254 | 6.889E-6 | 1.942E-5 | **9.31E-9** | **2.2949E-8** |
| | 20 | 1.0139 | 0.027 | 0.0306 | 0.02542 | 0.59036 | 0.2030 | **2.01E-08** | **6.76E-08** | 0.0052 | 0.0079 | 1.206E-4 | 1.9890E-4 |
| | 30 | 1.2342 | 0.11 | 0.0112 | 0.01422 | 0.8211 | 0.1394 | **2.87E-09** | **8.45E-10** | 0.0721 | 0.0522 | 0.0144 | 0.0220 |
| Rastrigin | 10 | 1.3928 | 0.763 | 2.6559 | 1.3896 | 0.43404 | 0.2551 | **0** | **0** | 5.594E-9 | 1.515E-8 | 1.327E-14 | 4.085E-14 |
| | 20 | 6.0309 | 1.4537 | 12.059 | 3.3216 | 1.8135 | 0.2551 | 1.45E-08 | 5.06E-08 | 2.154E-4 | 0.0016 | **3.31E-11** | **6.26E-11** |
| | 30 | 10.439 | 2.6386 | 32.476 | 6.9521 | 3.0527 | 0.9985 | 0.03388 | 0.18156 | 0.3899 | /-4/72 | **1.427E-8** | **2.4467E-8** |
| Ackley | 10 | 0.5927 | 0.2248 | **9.85E-13** | **9.62E-13** | 0.19209 | 0.1951 | 7.8E-11 | 1.16E-09 | 2.468E-6 | 5.407E-6 | 2.555E-8 | 4.8522E-8 |
| | 20 | 0.924 | 0.226 | 1.178E-6 | 1.584E-6 | 0.32321 | 0.09735 | **1.6E-11** | **1.9E-11** | 3.033E-5 | 1.916E-5 | 4.311E-7 | 3.5441E-7 |
| | 30 | 1.0989 | 0.2496 | 1.491E-6 | 1.861E-6 | 0.3771 | 0.09876 | **3E-12** | **5E-12** | 1.715E-4 | 1.019E-4 | 4.327E-6 | 2.5904E-6 |
| Schwefel | 10 | 1.9519 | 1.3044 | 161.87 | 144.16 | 0.32037 | 1.6185 | 1.27E-09 | 4E-12 | 0 | 0 | **0** | **0** |
| | 20 | 7.285 | 2.9971 | 543.07 | 360.22 | 1.4984 | 0.84612 | 19.8397 | 45.1234 | 0 | 0 | **0** | **0** |
| | 30 | 13.535 | 4.9534 | 990.77 | 581.14 | 3.272 | 1.6185 | 146.857 | 82.3144 | 0 | 0 | **0** | **0** |

All algorithms indicate the results that are obtained after 500, 750 and 1,000 cycle with a population having 125 individuals

## 5 Conclusion and Future Works

In this paper, an improved Imperialist algorithm is introduced which is called the Quad Countries Algorithm (QCA). In the QCA, we define four categories of country including Colonial, Colony, Independent, and Seeking Independence country. Therefore, each group of countries have special motion differently compared to the others. While, in the primary ICA, there are only two categories, Colony and Colonial, and the only motion is the colonies movement toward Imperialists which is applied with absorption policy. Whereas by adding Independent countries in the QCA, a new policy which is called repulsion policy, is also added. The empirical results were found by applying the proposed algorithm to some famous benchmarks indicate that the quality of global optima solutions and the convergence speed towards the optima have remarkably increased in the proposed algorithm in comparison to the primary ICA.

Through increasing the problem dimensions, the performance of the QCA increases considerably in comparison with the ICA. Compared to the QCA and GA, PSO, PS-EA and ABC, it observed that, in 100 percent of all cases the proposed algorithm has better performance than GA and PS-EA, but in comparison with ABC and PSO, in 50 percent of cases the QCA has better performance than ABC and PSO. ABC and PSO have better performance about 41.66 and 8.33percent of cases.

Overall, the performed experiments showed that, the QCA has considerably better performance in comparison with the primary ICA and also the other evolutionary algorithms such as GA, PSO, PS-EA and ABC. The Quad Countries Algorithm (QCA) has a proper performance to solve optimization problems. However, by changing the countries' movements and defining new moving policies, its performance will increase. In fact, by define new movement policies the ability of exploration and algorithm performance will increase.

## References

1. Sarimveis, H., Nikolakopoulos, A.: A Life Up Evolutionary Algorithm for Solving Nonlinear Constrained Optimization Problems. Computer & Operation Research 32(6), 1499–1514 (2005)
2. Mühlenbein, H., Schomisch, M., Born, J.: 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 (1991)
3. Holland, J.H.: ECHO: Explorations of Evolution in a Miniature World. In: Farmer, J.D., Doyne, J. (eds.) Proceedings of the Second Conference on Artificial Life (1990)
4. Melanie, M.: An Introduction to Genetic Algorithms. MIT Press, Massachusett's (1999)
5. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. Proceedings of IEEE, 1942–1948 (1995)
6. Atashpaz-Gargari, E., Lucas, C.: Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition. In: IEEE Congress on Evolutionary Computation (CEC 2007), pp. 4661–4667 (2007)

7. Atashpaz-Gargari, E., Hashemzadeh, F., Rajabioun, R., Lucas, C.: 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), 337–355 (2008)
8. Zhang, Y., Wang, Y., Peng, C.: Improved Imperialist Competitive Algorithm for Constrained Optimization. International Forum on Computer Science-Technology and Applications (2009)
9. Bahrami, H., Feaz, K., Abdechiri, M.: Imperialist Competitive Algorithm using Chaos Theory for Optimization (CICA). In: Proceedings of the 12th International Conference on Computer Modelling and Simulation (2010)
10. Bahrami, H., Feaz, K., Abdechiri, M.: Adaptive Imperialist Competitive Algorithm (AICA). In: Proceedings of The 9th IEEE International Conference on Cognitive Informatics, ICCI 2010 (2010)
11. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39(3), 459–471 (2007)
12. Srinivasan, D., Seow, T.H.: Evolutionary Computation. In: CEC 2003, Canberra, Australia, December 8-12, vol. 4, pp. 2292–2297 (2003)