A Hybrid Evolutionary Imperialist Competitive Algorithm (HEICA)

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

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

Abstract. This paper proposes a new approach by combining the Evolutionary Algorithm and Imperialist Competitive Algorithm. This approach tries to capture several people involved in community development characteristic. People live in different type of communities: *Monarchy*, *Republic* and *Autocracy*. People dominion is different in each community. Research work has been undertaken to deal with curse of dimensionality and to improve the convergence speed and accuracy of the basic ICA and EA algorithms. Common benchmark functions and large scale global optimization have been used to compare HEICA with ICA, EA, PSO, ABC, SDENS and jDElsgo. HEICA indeed has established superiority over the basic algorithms with respect to set of functions considered and it can be employed to solve other global optimization problems, easily. The results show the efficiency and capabilities of the new hybrid algorithm in finding the optimum. Amazingly, its performance is about 85% better than others. The performance achieved is quite satisfactory and promising.

Keywords: Evolutionary Algorithm (EA), Imperialist Competitive Algorithm (ICA) and a Hybrid Evolutionary Imperialist Competitive Algorithm (HEICA).

1 Introduction

Recently there has been considerable amount of attention devoted to bio-inspiration and bio-mimicry, for solving computational problems and constructing intelligent systems. In the scope of computational intelligence it seems there are at least six main domains of intelligence in biological systems and wild life: Swarming, Communication and Collaboration, Reproduction and Colonization, Learning and Experience, Competition and Evolution [1].

Evolutionary algorithms, such as Genetic Algorithm (GA) [2], Simulated Annealing (SA), Particle Swarm Optimization (PSO) [3-4] and Ant Colony Optimization (ACO) [5] are computer simulation of natural processes such as natural evolution and annealing processes in materials.

The Imperialist Competitive Algorithm (ICA) was proposed by Atashpaz, in 2007 [6]. ICA is the mathematical model and the computer simulation of human social

J.-S. Pan, S.-M. Chen, N.T. Nguyen (Eds.): ACIIDS 2012, Part I, LNAI 7196, pp. 359–368, 2012.

evolution. It is one of the recent meta-heuristic algorithms based on a socio-politically, proposed to solve optimization problems [6]. ICA can be thought of as the social counterpart of GA because uses imperialism and imperialistic competition, socio-political evolution process, as source of inspiration.

People get together in organized groups or similar close aggregations. In each country people try to reach better position in society such as job, religious, economic and culture promotion. In other hand, it occurs among different countries. People who live together in country work with the leader to be better among other countries. Therefore it is a good idea to combine EA and ICA algorithms to increase performance. We use EA to imagery a point of view, person and competition between people, and ICA to imagery another point of view, country and competition between countries.

The paper is organized as follows. Section 2, provides a brief literature overview of the EA and ICA. In Section 3, some related work is presented. In Section 4, new approach and the motivation of the HEICA is presented. In Section 5, results are compared with other algorithms.

2 Background

As an alternative to the conventional mathematical approaches, the meta-heuristic optimization techniques have been widely utilized and improved to obtain engineering optimum design solutions. Many of these methods are created by the simulation of the natural processes. Evolutionary Algorithm aims to simulate natural selection with survival of fittest in mind. Imperialist Competitive Algorithm simulates the social political process of imperialism and imperialistic competition.

2.1 Imperialist Competitive Algorithm (ICA)

This algorithm starts by generating a set of candidate random solutions in the search space of the optimization problem. The generated random points are called the initial population (countries in the world). Countries are divided into two groups: imperialists and colonies. The more powerful imperialist, have greater number of colonies. The cost function of the optimization problem determines the power of each country. Based on their power, some of the best initial countries (the countries with the least cost function value), become Imperialists and start taking control of other countries (called colonies) and form the initial Empires [6].

Three main operators of this algorithm are Assimilation, Revolution and Competition. This algorithm uses the assimilation policy. Based on this policy the imperialists try to improve the economy, culture and political situations of their colonies. This policy makes the colony's enthusiasm toward the imperialists. Assimilation makes the colonies of each empire get closer to the imperialist state in the space of socio-political characteristics (optimization search space). Revolution brings about sudden random changes in the position of some of the countries in the search space. During assimilation and revolution a colony might reach a better position and has the chance to take the control of the entire empire and replace the current imperialist state of the empire.

In competition operator, imperialists attempt to achieve more colonies and the colonies start to move toward their imperialists. All the empires try to win and take possession of colonies of other empires. The power of an empire depends on the power of its imperialist and its colonies. In each step of the algorithm, all the empires have a chance to take control of one or more of the colonies of the weakest empire based on their power. Thus during the competition the powerful imperialists will be improved and the weak ones will be collapsed. After a while, the weaker empires will lose all their colonies and their imperialists will transform to the colonies of the other empires; at the end, all the weak empires will be collapsed and only one powerful empire will be left. All the colonies are randomly divided among the imperialists. More powerful imperialists take possession of more colonies [6].

Algorithm continues until a stop condition is satisfied such as just one imperialist will remain. In this stage the position of imperialist and its colonies will be the same.

2.2 Evolutionary Algorithm (EA)

There are many different variants of Evolutionary Algorithms (EAs). The common underlying idea behind all these methods is the same. The idea of EA is based on survival of fittest and it causes a rise in the fitness of the population in different generations. Based on the fitness function some of the better candidates are chosen, they are seed the next generation by applying recombination and mutation. Execution of these operators leads to a set of new candidates, the offspring. Replacement operator replaces new offspring in next generation, based on their fitness.

3 Related Works

Abderchiri and Meybodi [7] proposed two algorithms for Solving SAT problems: First, a new algorithm that combines ICA and LR. Secondly, a hybrid Hopfield network (HNN)-Imperialist Competitive Algorithm (ICA). The proposed algorithm (HNNICA) has a good performance for solving SAT problems.

Vahid Khorani, Farzad Razavi and Ahsan Ghoncheh [8] proposed R-ICA-GA (Recursive-ICA-GA) based on the combination of ICA and GA. A new method improves the convergence speed and accuracy of the optimization results. They run ICA and GA consecutively. Results show that a fast decrease occurs while the proposed algorithm switches from ICA to GA.

Jain and Nigam [9] proposed a hybrid approach by combining the evolutionary optimization based GA and socio-political process based colonial competitive algorithm (CCA). They used CCA–GA algorithm to tune a PID controller for a real time ball and beam system.

Razavi and others [10] studied the ability of evolutionary Imperialist Competitive Algorithm (ICA) to coordinate over current relays. The ICA was compared to the GA. The algorithms were compared in terms of the mean convergence speed, mean convergence time, convergence reliability, and the tolerance of convergence speed in obtaining the absolute optimum point.

4 A Hybrid Evolutionary Imperialist Competitive Algorithm

In this section, combines two algorithms to present a novel hybrid algorithm. The pseudo-code of the HEICA is presented in follow:

```
Procedure HEICA
Step 1: Initialization;
   Generate some random people;
   Randomly allocate remain people to others countries;
   Select more powerful leaders as the empires;
Step 2: Evolutionary Algorithm
   Roulette Wheel Selection;
   Crossover;
   Mutation;
   Replacement;
Step 3: Imperialist Competitive Algorithm
   People Assimilation; Move the people of each country
   toward their relevant leaders.
   People Revolutionary;
   Countries Assimilation; Move the leaders of each
   country toward their empires and move the people of
   each country as the same as their leaders.
   Countries Revolutionary;
   Imperialistic Competition; Pick the weakest country
   from the weakest empire and give it to the empire
   that has the most likelihood to possess it.
   Elimination; Eliminate the powerless empires.
Step 4: Terminating Criterion Control; Repeat Steps 2-3
   until a terminating criterion is satisfied.
```

4.1 Population

This algorithm starts by generating a set of candidate random solutions in the search space of the optimization problem. The generated random points are called the initial population which consists of persons. *Persons* in this algorithm are the counterpart of *Chromosomes* in GA and *Particles* in PSO which are array of candidate solutions. In human society, groups of people form community and are involved in community development.

4.2 Types of Community

There are different kinds of community:

- *Republic*: A republic is led by representatives of the voters. Each is individually chosen for a set period of time. This type community has president. Best of people select as candidate and each people vote to their president. The votes have been counted and candidate with highest elected.

- *Autocracy*: This type of community does not have leader. The people are free and there is no force. They do what they want.
- *Monarchy*: A monarchy has a king or queen, who sometimes has absolute power. Power is passed along through the family. This type of community has a monarch; People should follow her. The powerful person is selected as the monarch in each country. Different monarchy countries exist in empires; the best monarch of this type of country selects as empire.

4.3 Initialization

The algorithm starts with an initial random population called person. Some of the best person in the population selected to be the leaders and the rest form the people of these countries. Each country has an equal population.

The total power of a country depends on both the power of the leaders and the power of its people. This fact is modeled by defining the total power of a country as the power of the leader of the country plus a percentage of mean power of its people. The power of the people has an effect on the total power of that country:

$$T.P_{C_i} = Cost(Leader_i) + \xi mean\{Cost(People\ of\ country_i)\} \tag{1}$$

$T.P_{C_i}$ is the total power of i-th country.

Based on monarchy countries power, some of the best initial countries (the countries with the least cost function value), become Imperialists and start taking control of other countries and form the initial Empires. The best leaders of the countries determine the empire of the Empires. We select N_{Imp} of the most powerful countries to form the empires. To divide the countries among imperialists proportionally, we use the normalized cost of an imperialist by [6]:

$$N.C_{I_i} = C_{I_i} - \underbrace{Max}_{j}\{C_{I_j})\} \tag{2}$$

C_{I_i} is the cost of i-th imperialist and $N.C_{I_i}$ is its normalized cost. Having the normalized cost of all imperialists, the normalized power of each imperialist is defined by [6]:

$$N.P_{I_i} = \left| \frac{C_{I_i}}{\sum_{j=1}^{N_{Imp}} C_{I_j}} \right| \tag{3}$$

The normalized power of an imperialist is the portion of colonies that should be possessed by that imperialist. Then the initial number of countries of an empire will be [6]:

$$N.C_{I_i} = round(N.P_{I_i} * N_{Monarchy}) \tag{4}$$

$N.C_{I_i}$ is the initial number of countries of i-th empire and $N_{Monarchy}$ is the number of all monarchy countries. To divide the countries, for each imperialist we randomly choose $N.C_{I_i}$ of the monarchy countries and give them to it. These countries along with the imperialist will form i-th empire.

4.4 Assimilation and Revolution

After Evolutionary Algorithm operator accomplish, imperialist started to improve their countries and countries started to improve their people. HEICA has modeled these facts by moving all the countries toward the imperialist and all the people toward the leaders:

- *External*: External operations are among the countries. Assimilation occurs just among monarchy countries of each imperialist they move toward the empires. The country moves toward the imperialist it means all the people among these countries move in the same way, toward the empires. Just monarchy countries have an empire therefore assimilation is toward the empire. Revolution occurs in all countries. All the people of one country should move toward the same way because revolution is against the empire in monarchy countries. In other countries they try to improve their countries therefore they move with each other.
- *Internal*: Internal operations are among the people of the countries. Assimilation occurs in all countries they move toward the leaders. Revolution occurs in all countries, people try to get the position.

5 Evaluation and Experimental Results

For evaluating performance of the proposed algorithm, the simulation results are compared with results of EA, ICA, PSO and ABC [11]. M is mean of best results. Performance calculated from Equation 5 as follow for minimum optimization:

$$P_{Algorithm} = 100\% * (1 - \frac{M_{HEICA}}{M_{Algorithm}}) \tag{5}$$

5.1 Benchmark Functions

Used common benchmark functions are listed in Table 1.The comparison results of F1-F10 are shown in Table 2. Best stability and convergence diagrams of different functions are shown in Fig. 1 and Fig. 2. In Table 3, the performance of HEICA algorithm is compared with PSO and ABC for high dimensional problem. In Table 4, the proposed HEICA algorithm was tested on benchmark functions provided by CEC2010 Special Session on Large Scale Global Optimization [12]. HEICA performs better than SDENS [13]. HEICA performs better on 1.2E5 and 6E5 FEs and jDElsgo[14] performs better on 3E6 FEs. To show the efficiency of HEICA in solving different function, logarithmic scale diagram is used. A logarithmic scale is a scale of measurement using the logarithm of a physical quantity instead of the quantity itself. Since, in this study, the values cover a wide range; logarithmic scale makes it easy to compare values.

5.2 Discussion

This paper proposes a novel hybrid approach consisting EA and ICA and its performance is evaluated using various test functions. Performance parameter shows HEICA perform better optimization than EA, ICA, PSO and ABC in all test functions. It shows that hybrid algorithm perform better optimization.

Table 1. Benchmark function (F1-F10)

Title	Function	Range
F1(Sphere)	$\sum_{i=1}^{D} x_i^2$	$-5.12 \le x_i \le 5.12$
F2(Rosenbrock)	$\sum_{i=1}^{D-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2$	$-2.048 \le x_i \le 2.048$
F3(Rastrigin)	$\sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	$-5.12 \le x_i \le 5.12$
F4(Griewangk)	$1 + \sum_{i=1}^{D} \left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{2} \left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$	$-600 \le x_i \le 600$
F5(Schwefel)	$\sum_{i=1}^{D} 418.9829 - x_i sin\,(\sqrt{\lvert x_i \rvert})$	$-500 \le x_i \le 500$
F6	$\sum_{i=1}^{D} 10^{i-1} x_i^2$	$-10 \le x_i \le 10$
F7(Schaffer)	$0.5 + \frac{sin^2\sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2}$	$-100 \le x, y \le 100$
F8(Schwefel 1.2)	$\sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	$-100 \le x, y \le 100$
F9(SumSquares)	$\sum_{i=1}^{D} i^2 x_i^2$	$-1 \le x_i \le 1$
F10(Ackley)	$-20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)} + 20 + e$	$-30 \le x_i \le 30$

6 Conclusion and Future Works

This paper proposed a novel hybrid approach consisting EA (Evolutionary Algorithm) and ICA (Imperialist Competitive Algorithm). The efficiency of HEICA is surveyed by comparing it evolutionary algorithm through a 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, i.e. it has the capability to come up with non-differentiable objective functions with a multitude number of local optima in a reasonable time limit. We are working on using LEM (Learnable Evaluation Model) to improve results.

Table 2. The results achieved by HEICA on F1-F10

F	F1		F2		F3		F4		F5	
D.	10	50	10	30	10	50	10	50	10	50
Gen.	1000	2000	2000	4000	1000	2000	1000	3000	1000	2000
Pop.	150	500	250	600	150	400	450	750	150	300
EA	9.74E-03	5.15E+0	6.50E+00	7.42E+01	1.27E+00	7.81E+01	2.41E-02	1.06E-02	8.52E+00	1.58E+03
ICA	6.04E-04	8.18E-03	8.49E-01	1.11E+01	1.23E-01	2.46E+00	1.77E-03	2.15E-02	9.57E-01	9.59E+01
PSO	2.30E-35	2.14E-12	1.33E+00	3.32E+01	1.69E+00	6.72E+01	5.68E-02	1.00E-02	6.88E+02	7.76E+03
ABC	7.30E-17	1.14E-15	2.60E-03	5.79E+00	**0**	5.79-12	1.67E-17	**2.22E-17**	2.90E-02	3.19E+02
HEICA	**1.36E-39**	8.88E-23	**3.23E-03**	**7.06E+00**	**0**	**0**	**0**	3.33E-16	**1.27E-04**	**6.36E-04**
$P_{EA}(\%)$	100	100	100	99	100	100	100	100	100	100
$P_{PSO}(\%)$	100	100	100	97	100	100	100	100	100	100
$P_{ABC}(\%)$	100	100	85	85	-	100	100	-823	100	100

F	F6		F7	F8		F9		F10	
D.	10	50	2	10	50	10	50	10	50
Gen.	1000	2500	1000	1000	1500	1000	2000	1000	2000
Pop.	150	450	150	150	200	150	500	150	200
EA	3.63E+05	1.55E+45	2.09E-02	1.09E+01	3.73E+04	1.01E-02	1.12E+02	1.52E+00	6.03E+00
ICA	1.78E+01	1.20E+28	1.34E-08	1.00E+00	1.19E+02	4.79E-04	9.53E-02	2.74E-01	7.15E-01
PSO	8.18E-16	2.16E+11	3.45E-04	-	-	2.61E-35	7.73E-11	4.26E-15	1.12E-03
ABC	6.32E-17	3.19E+02	6.88E-06	-	-	7.26E-17	1.57E-15	6.93E-15	2.02E-07
HEICA	**5.80E-37**	**1.11E-01**	**0**	**2.79E-39**	**9.06E-12**	**5.76E-39**	**8.77E-23**	**4.00E-15**	**6.51E-11**
$P_{EA}(\%)$	100	100	100	100	100	100	100	100	100
$P_{PSO}(\%)$	100	100	100	-	-	100	100	6	100
$P_{ABC}(\%)$	100	100	100	-	-	100	100	42	100

Table 3. Comparing HEICA with ABC and PSO (high dimensions)

	D.	Pop.	Gen.	ABC	PSO	HEICA	$P_{ABC}(\%)$	$P_{PSO}(\%)$
F1	500	600	1500	2.26E+02	3.90E+03	**3.83E+01**	83	99
	1000	800	2000	1.46E+03	8.00E+03	**9.29E-01**	100	100
F2	500	600	1500	8.44E+03	1.99E+05	**4.53E+03**	46	98
	1000	800	2000	5.09E+04	4.21E+05	**4.03E+03**	92	99
F3	500	600	1500	1.93E+03	8.59E+03	**1.14E+03**	41	87
	1000	800	2000	6.05E+03	1.74E+04	**5.20E+02**	91	97
F4	500	600	1500	9.48E+02	5.10E+01	**1.23E+00**	100	98
	1000	800	2000	4.86E+03	2.98E+02	**5.10E+00**	100	98
F9	500	600	1500	5.23E+05	1.12E+07	**7.15E+04**	86	99
	1000	800	2000	1.84E+08	9.50E+07	**6.60E+03**	100	100
F10	500	600	1500	1.49E+01	2.09E+01	**1.88E+00**	87	91
	1000	800	2000	1.77E+01	1.74E+04	**4.89E+00**	72	100

Table 4. The results achieved by HEICA on the test suite

FEs	Algorithm	Shifted Elliptic	Shifted Rastrigin	Single-group Shifted and m-rotated Elliptic	Single-group Shifted m-dimensional Rosenbrock	$\frac{D}{2m}$-group Shifted and m-rotated Elliptic	$\frac{D}{2m}$-group Shifted and m-rotated Elliptic	$\frac{D}{2m}$-group Shifted and m-rotated Ackley	Shifted Schwefel
1.2E5	PSO	1.26E+10	2.36E+04	**4.06E+13**	**5.51E+08**	3.74E+10	4.31E+10	4.29E+02	9.92E+11
	SDENS[13]	5.01E+09	1.19E+04	5.10E+13	7.71E+08	1.56E+10	1.84E+10	**4.15E+02**	2.61E+11
	jDElsgo[14]	3.70E+09	1.09E+04	1.40E+14	2.39E+09	1.64E+10	2.32E+10	4.17E+02	7.99E+10
	HEICA	**9.76E+08**	**6.29E+03**	4.10E+13	8.07E+09	**3.54E+09**	**1.36E+10**	4.21E+02	**2.67E+10**
6E5	PSO	5.78E+08	2.36E+04	1.31E+13	1.19E+08	8.82E+09	1.52E+10	4.25E+02	1.00E+11
	SDENS	7.87E+06	7.09E+03	1.72E+13	7.41E+07	2.23E+09	5.14E+09	4.13E+02	2.69E+08
	jDElsgo	**8.99E+04**	3.95E+03	1.39E+13	6.82E+07	1.66E+09	4.10E+09	**2.99E+02**	1.01E+06
	HEICA	2.49E+06	**5.55E+02**	**8.42E+12**	**6.33E+07**	**9.55E+08**	**3.61E+09**	4.13E+02	**9.61E+05**
3E6	PSO	1.99E+07	2.36E+04	5.72E+12	7.16E+07	1.87E+09	6.45E+09	4.09E+02	8.55E+09
	SDENS	5.73e–06	2.21E+03	5.11E+12	5.12E+07	5.63E+08	1.88E+09	4.08E+02	**9.90E+02**
	jDElsgo	**8.86E-20**	**1.25E-01**	**8.06E+10**	**3.15E+06**	**3.11E+07**	**1.69E+08**	**1.44E+02**	1.53E+03
	HEICA	3.49E+04	2.62E+00	3.10E+12	5.24E+07	3.00E+08	1.09E+09	4.01E+02	3.73E+03

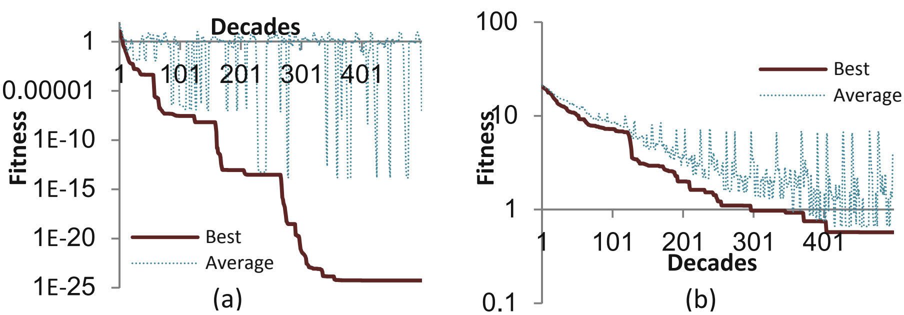

Fig. 1. Convergence diagram of HEICA: (a) F1 (D = 10) (b) F10 (D = 50)

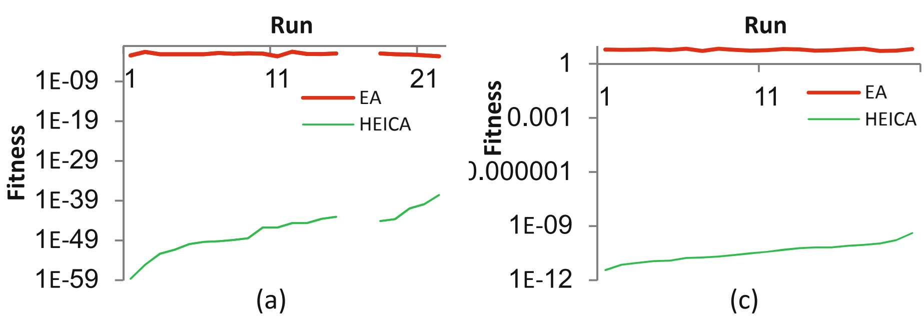

Fig. 2. Best stability diagram of HEICA: (a) F1 (D = 10) (b) F10 (D = 50)

References

1. Hajimirsadeghi, H., Lucas, C.: A Hybrid IWO/PSO Algorithm for Fast and Global Optimization. In: International IEEE Conference Devoted to 150 Anniversary of Alexander Popov (EUROCON 2009), Saint Petersburg, Russia, pp. 1964–1971 (2009)
2. Holland, J.H.: Adoption in Natural and Artificial Systems. University of Michigan, Ann Arbor (1975)
3. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Australia, pp. 1942–1948 (1995)
4. Eberhart, R., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS 1995), pp. 39–43 (1995)
5. Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man and Cybernetics, 1–13 (1996)
6. Atashpaz-Gargari, E., Lucas, C.: Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired By Imperialistic Competition. In: Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, pp. 4661–4667 (2007)
7. Abdechiri, M., Meybodi, M.R.: A Hybrid Hopfield Network-Imperialist Competitive Algorithm for Solving the SAT Problem. In: 3rd International Conference on Signal Acquisition and Processing (ICSAP 2011), pp. 37–41 (2011)
8. Khorani, A.V., Razavi, B.F., Ghoncheh, C.A.: A new Hybrid Evolutionary Algorithm Based on ICA and GA: Recursive-ICA-GA. In: The 2010 International Conference on Artificial Intelligence, pp. 131–140 (2010)
9. Jain, T., Nigam, M.J.: Synergy of Evolutionary Algorithm and Socio-Political Process for Global Optimization. Expert Systems with Applications 37, 3706–3713 (2010)
10. Razavi, F., Khorani, V., Ghoncheh, A., Abdollahi, H., Askarian, I.: Using Evolutionary Imperialist Competitive Algorithm (ICA) to Coordinate Overcurrent Relays. In: The 2011 International Conference on Genetic and Evolutionary Methods, GEM 2011 (2011)
11. Karaboga, D., Basturk, B.: Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 789–798. Springer, Heidelberg (2007)
12. Tang, K., et al.: Benchmark Functions for the CEC 2010 Special Session and Competition on Large-Scale Global Optimization. s.l.: Nature Inspired Computation and Applications Laboratory, Technical report (2010), `http://nical.ustc.edu.cn/cec10ss.php`
13. Wang, H., Wu, Z., Rahnamayan, S., Jiang, D.: Sequential DE Enhanced by Neighborhood Search for Large Scale Global Optimization. In: WCCI 2010 IEEE World Congress on Computational Intelligence, pp. 4056–4062 (2010)
14. Brest, J., Zamuda, A., Fister, I., Maucec, M.S.: Large Scale Global Optimization using Self-adaptive Differential Evolution Algorithm. In: WCCI 2010 IEEE World Congress on Computational Intelligence, pp. 3097–3104 (2010)