A Hybrid ICA/PSO Algorithm by Adding Independent Countries for Large Scale Global Optimization

Amirhossein Ghodrati[1], Mohammad V. Malakooti[2], and Mansooreh Soleimani[2]

[1] Computer Engineering Department, College of Nabi Akram, Tabriz, Iran
[2] Islamic Azad University, UAE Branch, Department of Computer Engineering
{a.h.ghodrati,m.soleimani65}@gmail.com, malakooti@iau.ae

Abstract. This paper presents the hybrid approach of Imperialist Competitive Algorithm (ICA) and Particle Swarm Optimization (PSO) for global optimization. In standard ICA, there are only two types of countries: imperialists and colonies. In the proposed hybrid algorithm (ICA/PSO) we added another type of country, '*Independent*'. Independent countries do not fall into the category of empires, and are anti-imperialism. In addition, they are united and their shared goal is to get stronger in order to rescue colonies and help them join independent countries. These independent countries are aware of each other positions and make use of swarm intelligence in PSO for their own progress. Experimental results are examined with benchmark functions provided by CEC2010 Special Session on Large Scale Global Optimization (LSGO) and the results are compared with some previous LSGO algorithms, standard PSO and standard ICA.

Keywords: ICA, Global optimization, PSO, Hybrid evolutionary algorithm and Swarm intelligence.

1 Introduction

Computing optimal solutions is intractable for many optimization problems of industrial and scientific importance [1]. The complexity of the problem of interest makes it impossible to search every possible solution or combination [2]. However, due to their complexity, the use of approximation algorithms for solving them has become almost popular in the past years. Among these optimization algorithms, modern metaheuristics are becoming increasingly popular, leading to a new branch of optimization, called metaheuristic optimization. Most metaheuristic algorithms are nature-inspired [3], some of which have been proposed for optimization problems.

PSO was formulated by Kennedy and Eberhart in 1995 [4]. It is an evolutionary computation technique which is inspired by social behavior of swarms. This algorithm is the simulation of the social behavior of birds, like the choreography of a bird flock. Each individual in the population is a particle and gets a random value in the initialization. Each particle despite the position vector contains the best personal experience and a velocity vector. The particle's velocity and its best personal

J.-S. Pan, S.-M. Chen, N.T. Nguyen (Eds.): ACIIDS 2012, Part III, LNAI 7198, pp. 99–108, 2012.

experience and the global best position all together determines a particle next movement. PSO has variety of usages such as [5] and [6].

ICA algorithm has been proposed by Atashpaz-Gargari and lucas in 2007 that has been inspired by a socio-human phenomenon [7]. This algorithm like other evolutionary algorithms starts with an initial population. Each individual of a given population is considered a country. There are two types for countries: colonies and imperialists that all together form some empires. Imperialistic competition among these empires forms the basis of the ICA evolutionary algorithm. During this competition, weak empires collapse and powerful ones take possession of their colonies. Imperialistic competition hopefully converges to a state in which there exists only one empire and its colonies are in the same position and have the same cost as the imperialists [7].

ICA and PSO are nature inspired algorithms and work well for optimization problems. In this paper, we have tried to get closer to the reality and improve the ICA by adding a new category named ‘independent countries’. Despite the original idea of ICA which is based on imperialism and colonies, we considered a group of independent peaceful countries. These countries are united and they communicate with each other using swarm intelligence. They are against the imperialism countries and are in competition with them. The experimental results show that we are obtaining better results. Standard ICA is discussed in section 2. Section 3 provides an overview of PSO approach. Section 4 gives a hybrid approach of ICA with PSO and Section 5 presents the detailed experimental results to compare the performance of the proposed algorithm with other algorithms.

2 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. This algorithm like other algorithms starts with some initial random solutions which each one is named country. Some of the best countries in the population are selected to be the imperialists and the rest form the colonies of these imperialists. Each imperialist enclose colonies based on their power. Thus, the powerful imperialists will have more colonies than the weaker ones. In an N dimensional optimization problem a country is defined as below:

$$Country = [P_1, P_2, \dots, P_N] . \tag{1}$$

The cost of each country is evaluated with the cost function f at variables $(P_1, P_2, \dots, P_N)$ as the following:

$$c_i = f(country_i) = f(P_{i1}, P_{i2}, \dots, P_{iN}) . \tag{2}$$

When the imperialists form their empire, the imperialist countries absorb their colonies towards themselves using the absorption policy. The absorption policy shown in Fig. 1 makes the main core of this algorithm and causes the countries to move towards their minimum optima. In ICA algorithm, 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 θ, which is chosen randomly and with an uniform distribution. In Our implementation γ is ${}^{\pi}/_{4}\,(Rad)$.

$$\theta \sim U(-\gamma, \gamma)\ . \tag{3}$$

In the absorption policy, the colony moves towards the imperialist by x unit. In Fig. 1 the distance between the imperialist and colony shown by d and x is a random variable with uniform distribution. β is greater than 1 and is near to 2. Therefore, a proper choice can be $\beta = 2$.

$$x \sim U(0, \beta \times d)\ . \tag{4}$$

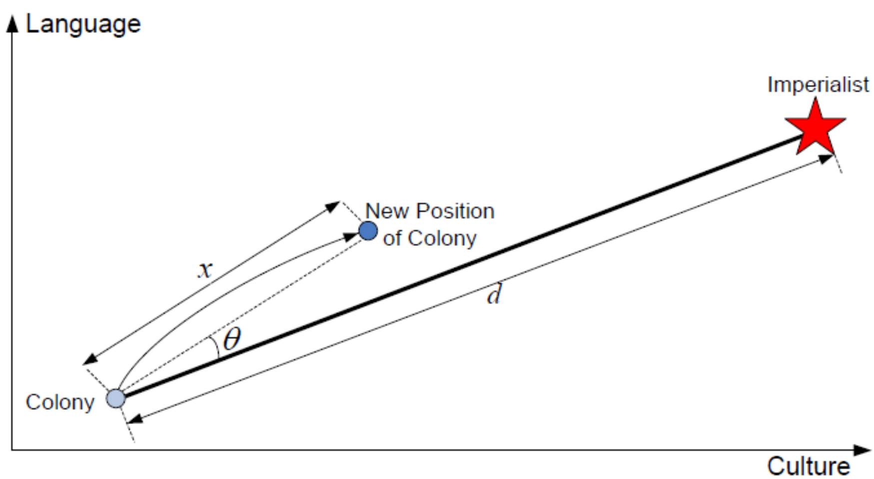

Fig. 1. Moving colonies toward their imperialist

After the absorption process, we will have the revolution operator. It is a known fact that revolution takes place in some countries, so in this algorithm revolution occurs with a probability. Revolution makes a sudden change in one or more parameters of the problem. After revolution and absorption, a colony may reach a better position, so the colony position changes according to the position of the imperialist.

For imperialistic competition first we calculate the total cost of each empire as below:

$$TC_n = cost(imperialist_n) + \xi mean\{cost(colonies\ \ of\ empire_n)\}\ . \tag{5}$$

In imperialistic competition, the weakest colony of the weakest empire will be chosen for a competition among all empires. Each empire has a chance to win that colony based on their power. Consequently, the stronger empire will have a greater chance and the weaker one will have a smaller chance.

3 Particle Swarm Optimization

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates [8].

PSO is an evolutionary algorithm which is inspired by the feeding birds or fish and proposed by Kennedy and Eberhart in 1955. This algorithm like other evolutionary algorithms starts with random initial solutions and begins the process of finding the global optimum. In this algorithm, we call each solution a *particle.* Each particle moves around in the search space with a velocity. The best position explored for a particle so far is recorded and is called *pbest*. Moreover, each particle knows the best *pbest* among all the particles which is called *gbest.* By considering *pbest, gbest* and the velocity of each particle the update rule for their position is as the following equations:

$$V_{t+1} = W_t * V_t + C_1 * rand(\quad) * (pbest - x_t) + C_2 * rand(\quad) * (gbest - x_t)\ . \tag{6}$$

$$x_{t+1} = x_t + V_{t+1}\ . \tag{7}$$

Where W is inertia weight which shows the effect of previous velocity vector (V_t) on the new vector, C_1 and C_2 are acceleration constants and $rand(\quad)$ is a random function in the range [0, 1] and x_t is current position of the particle.

4 Hybrid ICA/PSO Algorithm

In this section, we explore the details of the hybrid algorithm by adding independent countries to ICA. In standard ICA, we have colonies and imperialists only. In this paper, we have tried to get closer to the nature's reality. We have added a group of independent peaceful countries to the standard ICA. These independent countries are anti-imperialism and are united and communicate with each other. This communication is obtained from the swarm intelligence in the PSO. In the initial population, the most powerful countries are selected as independent. These countries are chosen to be powerful because the imperialists could not treat them as a colony for themselves. In fact, these countries are anti-imperialist, so they will not constitute an empire. In each iteration of this algorithm, three more steps have been added:

4.1 Step 1. Move Independent Countries

Each country will take a new position based on three parameters:

- *pbest*: The best personal experience of that country,
- *gbest*: The best *pbest* among all the countries,
- *velocity:* The countries current velocity.

All the independent countries will move in the search space based on the following equations:

$$V_{t+1} = W_t * V_t + C_1 * rand(\quad) * (pbest - country_t) + C_2 * rand(\quad) * (gbest - country_t)\ . \tag{8}$$

$$country_{t+1} = country_t + V_{t+1}\ . \tag{9}$$

Where W is inertia weight which shows the effect of previous velocity vector (V_t) on the new vector, C_1 and C_2 are acceleration constants and $rand(\quad)$ is a random function in the range [0, 1] and $country_t$ is current position of the country. In PSO particles move toward their best experience and global best according to equation 6 and 7, however in ICA/PSO each country update its position based on its best experience and global best of independent countries.

4.2 Step 2. Competition for Independency

As mentioned earlier, independent countries are anti-imperialism and these countries are in competition with imperialist countries. Independent countries' aim is to free the colonies from the empires and let them join independent countries to cause the collapse of all empires. In this part of algorithm, we calculate the total cost of independent countries with the mean of each ones cost.

$$TC_I = mean\{cost(independent\ countries)\}\ . \tag{10}$$

We selected the weakest colony of each empire that was weaker than independent countries. This was obtained by comparing the total cost of independent countries with the empires total cost. Then, we moved the selected colony toward the independent countries according to equation 8 and 9. This normally happens due to the fact that the colonies are not interested to be a colony, and they try to separate themselves from their empires. On the other hand, the imperialist country of that empire does not want to lose its colonies, but independent countries are more powerful than this empire and the empires are not able to stand against independent countries. The reason for selecting the weakest colony of that empire is that empires will not attempt so much to keep the weakest colony. If the weakest colony gets stronger than its imperialist country after the movement toward independent countries, then they will leave that empire and join independent countries.

4.3 Step 3. Competition to Colonize Independent Countries

If the power of independent countries is less than that of all empires, then one of the independent countries will be available for competition between all empires like the "imperialistic competition" stage in ICA. This independent country is not interested to be a colony, but the independent countries are not powerful enough to protect that country. By 'the power of independent countries', we mean the total cost in equation 10. The pseudo-code of the ICA/PSO is presented as bellow:

```
Procedure ICA/PSO
Step 1: Initialization
   Generate some random countries;
   Select the most powerful countries as independent;
   Initialize remaining countries as empires;
Step 2: Hybrid ICA/PSO Algorithm
   Move independent countries;
   Assimilate colonies toward their imperialist;
   Countries revolution;
   Exchange imperialist with best colony if is
   necessary;
   Calculate total cost of empires;
   Competition for independency;
   Competition for colonizing independent countries;
   Imperialistic competition;
   Eliminate the powerless empires;
Step 3: Terminating Criterion Control; Repeat Step 2
```

5 Experimental Results

In this section, we show experimental results which evaluate our hybrid ICA/PSO proposed algorithm. In the field of evolutionary computation, it is common to compare different algorithms using benchmark functions, especially when the test involves function optimization. Thus, the proposed algorithm is tested on benchmark functions provided by CEC2010 Special Session on Large Scale Global Optimization [9]. This test suite is composed by five types of high dimensional problems. We picked eight functions and at least one from each type, and then we compared the results with our implementation of standard ICA, our implementation of standard PSO, jDElsgo [10], SDENS [11] and DECC-DML [12]. The mean of best values produced by the algorithms have been recorded and demonstrated for comparison in table 1. Fig. 1 shows the logarithmic scale convergence plot of ICA/PSO. Figure 2 shows the stability diagram after 3.00E+06 function evaluation for functions F1 and F6. The selected eight functions are as follows:

Table 1. Experimental results with ICA/PSO

FEs	Alg.	F1	F2	F3	F4	F5	F6	F7	F8
1.20E+05	ICA/PSO	3.15E+03	1.69E+13	9.68E+05	3.92E+02	4.5E+09	1.66E+01	2.2E+09	1.85E+06
	ICA	2.30E+04	5.48E+14	1.60E+07	4.22E+02	5.61E+12	2.14E+01	2.35E+11	2.86E+07
	PSO	2.36E+04	4.06E+13	6.24E+06	4.29E+02	9.92E+11	2.13E+01	3.74E+10	6.34E+06
	jDElsgo	1.09E+04	1.40E+14	3.15E+06	4.17E+02	7.99E+10	1.87E+01	1.64E+10	4.85E+06
	SDENS	1.19E+04	5.10E+13	2.95E+06	4.15E+02	2.61E+11	2.01E+01	1.56E+10	4.31E+06
	DECC-DML	5.75E+03	6.76E+13	4.70E+06	3.75E+02	3.84E+09	9.51E+00	4.89E+09	8.81E+06
6.00E+05	ICA/PSO	1.63E+02	4.38E+12	2.69E+05	3.88E+02	4.65E+03	1.38E-01	3.1E+08	6.46E+05
	ICA	2.24E+04	5.45E+14	1.60E+07	4.21E+02	5.61E+12	2.14E+01	2.35E+11	2.86E+07
	PSO	2.36E+04	1.31E+13	2.10E+06	4.25E+02	1.00E+11	2.05E+01	8.82E+09	3.81E+06
	jDElsgo	3.95E+03	1.39E+13	9.39E+05	2.99E+02	1.01E+06	1.22E+00	1.66E+09	1.95E+06
	SDENS	7.09E+03	1.72E+13	1.32E+06	4.13E+02	2.69E+08	6.12E+00	2.23E+09	2.07E+06
	DECC-DML	2.64E+03	1.61E+13	4.19E+06	4.47E+01	1.69E+03	1.81E-02	3.73E+08	7.27E+06
3.00E+06	ICA/PSO	1.17E-13	1.25E+12	2.52E+04	3.88E+02	6.79E+02	1.19E-09	5.5E+07	8.62E+04
	ICA	2.18E+04	5.45E+14	1.60E+07	4.21E+02	5.61E+12	2.14E+01	2.35E+11	2.86E+07
	PSO	2.36E+04	5.72E+12	7.37E+05	4.09E+02	8.55E+09	1.99E+01	1.87E+09	3.50E+06
	jDElsgo	1.25E-01	8.06E+10	1.21E+04	1.44E+02	1.53E+03	3.81E-12	3.11E+07	1.02E+05
	SDENS	2.21E+03	5.11E+12	4.13E+05	4.08E+02	9.90E+02	2.70E-05	5.63E+08	1.08E+06
	DECC-DML	2.17E+02	3.58E+12	3.80E+06	5.08E-02	9.91E+02	1.18E-13	5.92E+07	6.54E+06

- The first function $F1$ is Shifted Rastrigin's Function. The function is multimodal, shifted, separable and scalable.
- The second function $F2$ is Single-group Shifted and m-rotated Elliptic Function. The function is unimodal, shifted, single-group m-rotated and single-group m-nonseparable.
- The third function ($F3$) is $\frac{D}{2m}$ - group Shifted m-dimensional Schwefel's Problem. This Schwefel's problem is unimodal, shifted and $\frac{D}{2m}$ - group m-nonseparable.
- The forth function ($F4$) is $\frac{D}{m}$ - group Shifted and m-rotated Ackley's Function. The function is multimodal, shifted, $\frac{D}{m}$ - group m-rotated and $\frac{D}{m}$ - group m-nonseparable.
- The fifth function ($F5$) is Shifted Rosenbrock's Function. The function is multimodal, shifted and fully-nonseparable.
- The sixth function ($F6$) is Shifted Ackley's Function. The function is multimodal, shifted, separable and scalable.
- The seventh function ($F7$) is $\frac{D}{2m}$ - group Shifted m-rotated Elliptic Function. The function is unimodal, shifted, $\frac{D}{2m}$ - group m- rotated and $\frac{D}{2m}$ - group m-nonseparable.
- The eighth function ($F8$) is $\frac{D}{m}$ - group Shifted m-dimensional Schwefel's Problem. This Schwefel's problem is unimodal, shifted and $\frac{D}{m}$ - group m-nonseparable.

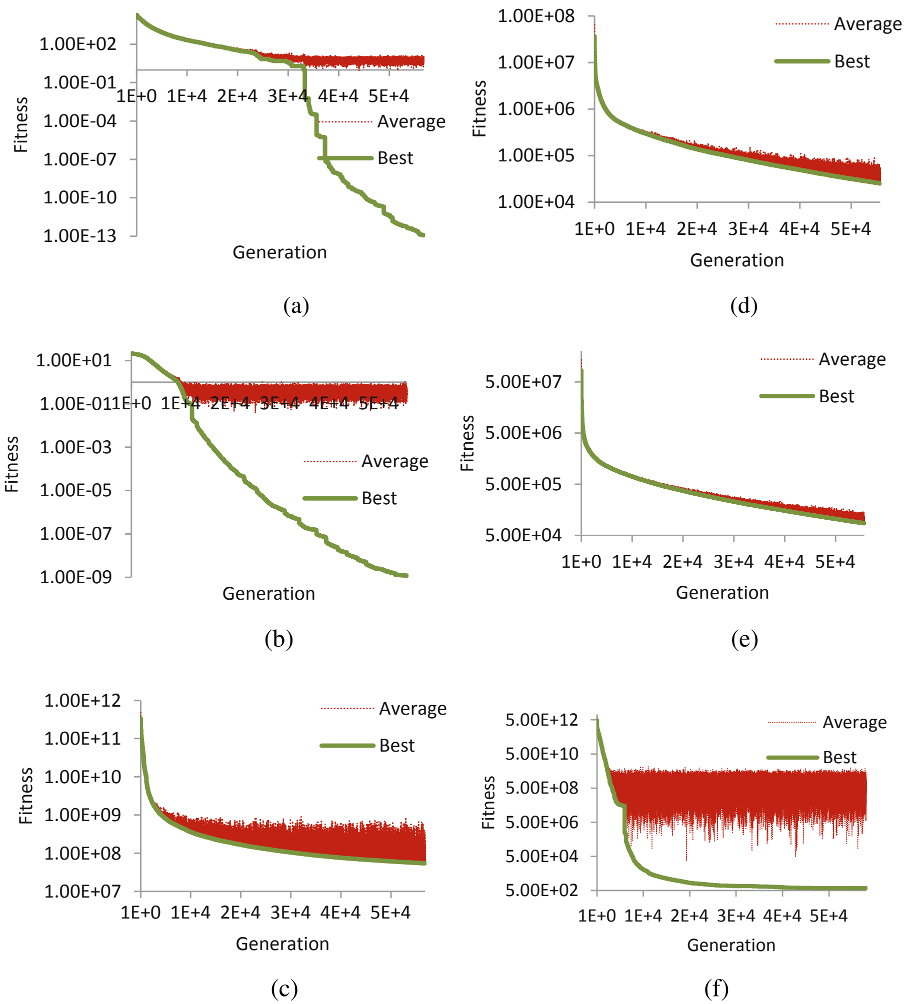

Fig. 2. Convergence diagram for functions: (a) *F1*; (b) *F3*; (c) *F5*; (d) *F6*; (e) *F7* and (f) *F8*

The control parameter used to define the degree of separability of a given function, in the given test suite is set as $m = 50$. The parameter settings of ICA/PSO are described as follows: The population size is set to 50, the number of empires is set to 5 and the number of independent countries is 10. Acceleration constants $C1$, $C2$ are set to 1.5 and the inertia weight is 0.7. The maximum number of Functions Evaluations (FEs) is set to 3e+6 for all test functions, and the study has been carried out with dimension D=1000 and 20 runs of algorithm were needed for each function.

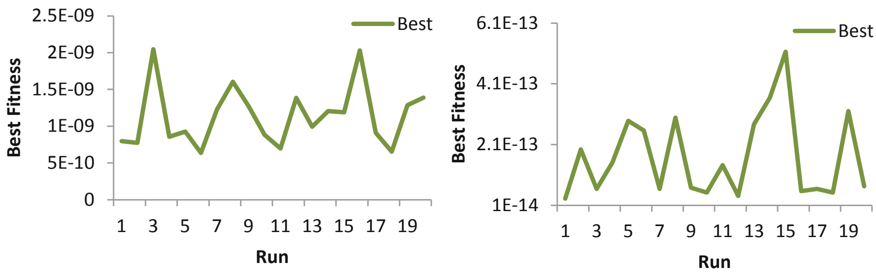

Fig. 3. Stability diagram after 3.00E+06 function evaluation for functions: (a) *F1* and (b) *F6*

6 Conclusion

This paper introduced a new hybrid ICA/PSO algorithm. Besides the colonies and imperialists in the standard ICA, we added independent anti-imperialism countries which move in the search space as the particles do in PSO. In fact, independent countries and imperialists are in competition, and when one of them finds a better solution, after a while, all countries in the search space will converge to that group. Therefore, we have more chance to find better solutions and more spaces are observed by moving countries from one group to another. ICA/PSO algorithm was evaluated on 5 of the benchmark functions provided by CEC2010 Special Session on Large Scale Global Optimization. The results show that the proposed hybrid algorithm outperforms the standard ICA, standard PSO and SDENS with 100%, jDElsgo with more than 80% and DECC-DML with more than 65% of the total time in comparison. The results show an acceptable performance of the proposed algorithm.

References

1. Talbi, E.G.: Metaheuristic: from design to implementation. Wiley Publishing, Hoboken (2009)
2. Yang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley Publishing, New Jersey (2010)
3. Yang, X.-S.: Metaheuristic Optimization: Algorithm Analysis and Open Problems. In: Pardalos, P.M., Rebennack, S. (eds.) SEA 2011. LNCS, vol. 6630, pp. 21–32. Springer, Heidelberg (2011)
4. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, pp. 1942–1948 (1995)
5. Puranik, P., Bajaj, P., Abraham, A., Palsodkar, P., Deshmukh, A.: Human Perception-based Color Image Segmentation Using Comprehensive Learning Particle Swarm Optimization. Journal of Information Hiding and Multimedia Signal Processing 2(3), 227–235 (2011)
6. Chang, F.C., Huang, H.-C.: A Refactoring Method for Cache-Efficient Swarm Intelligence Algorithms. Information Sciences, doi:10.1016/j.ins.2010.02.025

7. 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)
8. Nedjah, N., Mourelle, L.M.: Swarm Intelligent Systems. Springer, New York (2006)
9. Tang, K., et al.: Benchmark Functions for the CEC 2010 Special Session and Competition on Large-Scale Global Optimization: Nature Inspired Computation and Applications Laboratory Technical report (2010), http://nical.ustc.edu.cn/cec10ss.php
10. Brest, J., Zamuda, A., Fister, I., Maučec, M.S.: Large Scale Global Optimization using Self-adaptive Differential Evolution Algorithm. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC). IEEE Press, Barcelona (2010)
11. Wang, H., Wu, Z., Rahnamayan, S., Jiang, D.: Sequential DE Enhanced by Neighborhood Search for Large Scale Global Optimization. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC). IEEE Press, Barcelona (2010)
12. Omidvar, M.N., Li, X., Yao, X.: Cooperative Co-evolution with Delta Grouping for Large Scale Non-separable Function Optimization. In: Proceedings of IEEE Congress onEvolutionary Computation (CEC), pp. 1762–1769. IEEE Press, Barcelona (2010)