Applied Soft Computing 13 (2013) 2837–2856

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Social-Based Algorithm (SBA)

Fatemeh Ramezani[a,*], Shahriar Lotfi[b]

[a] Computer Engineering Department, College of Nabi Akram, Iran
[b] Computer Science Department, University of Tabriz, Iran

ARTICLE INFO

Article history:
Received 6 September 2011
Received in revised form 13 May 2012
Accepted 16 May 2012
Available online 29 May 2012

Keywords:
Evolutionary Algorithm
Imperialist Competitive Algorithm
Social-Based Algorithm

ABSTRACT

This paper proposes a new approach by combining the Evolutionary Algorithm (EA) and socio-political process based Imperialist Competitive Algorithm (ICA). This approach tries to capture several people involved in community development characteristic. People live in different type of communities: *Monarchy*, *Republic*, *Autocracy* and *Multinational*. Leadership styles are 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. The proposed algorithm has been compared with some well-known heuristic search algorithms. The obtained results confirm the high performance of the proposed algorithm in solving various benchmark functions specially in high dimensional problem. Simulation results were reported and the SBA 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 other algorithms such as EA and ICA. The performance achieved is quite satisfactory and promising for all test functions.

1. Introduction

Recently there has been a 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].

Ants foraging, birds flocking, fish schooling and bacterial chemotactics are some examples in the category of swarming [2–8]. Communication is present among most animals and plants. The way they share their information about the quality of nutrients makes one of the fastest mediums of communication in collective systems [9,10]. Reproduction is existed in all creatures. An intellectual mechanism for breeding or reproducing like making fruiting body is existed in some kinds of bacteria [11]. Plants, trees, roots and branches have intelligent attitude toward colonization. Some creatures like human, ant and bee are capable of learning and experience. Competition can be seen in all spices to form new individuals. There is numerous evidence of evolution in nature.

Evolutionary Algorithms, such as Genetic Algorithm (GA) [12], Simulated Annealing (SA), Particle Swarm Optimization (PSO) [13,14] and Ant Colony Optimization (ACO) [15] are computer simulation of natural processes such as natural evolution and annealing processes in materials. For several problems, an EA might be good enough to find the desired solution but there are some types of problems where it could fail to obtain optimal solution [16]. Combination of algorithms has provided very powerful search algorithms. Possible reasons for hybridization are to improve the performance and the quality of the solutions obtained by the EA and to incorporate the EA as part of a larger system [16].

Over the past few years, interest in hybrid meta-heuristics has raised considerably among researchers in combinatorial optimization. The best results found for many optimization problems are obtained by hybrid algorithms. Therefore, it is clear to need hybrid EAs with other optimization algorithms, machine learning techniques, heuristic, etc. Recently, hybridization of Evolutionary Algorithms is getting popular due to their capabilities in handling several real world problems. Several techniques and heuristics/meta-heuristics have been used to improve the general efficiency of the EA [16,17] and researchers used them in solving manufacturing problems [18,19].

Many researchers have used the idea of hybridization for EAs. Niknam [20] proposed a new hybrid EA to find optimal or near optimal solutions for clustering problems of allocating N objects to k clusters. A novel hybrid evolutionary optimization method is based on a combination of ICA, SA and k-means algorithms called K-MICA. They have used the advantages of k-means and ICA, and

* Corresponding author. Tel.: +98 9183688125.
E-mail addresses: f60_ramezani@yahoo.com (F. Ramezani), shahriar_lotfi@tabrizu.ac.ir (S. Lotfi).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.05.018

reduced their disadvantages. The simulated annealing (SA) is used as a local search around the best solution found by MICA algorithm.

Shi et al. [21] proposed a novel PSO-GA-based hybrid algorithm (PGHA). PGHA executes the two algorithms simultaneously and selects *P* individuals from each algorithm for exchanging after the designated *N* iterations. The individual with larger fitness has more opportunities of being selected. Results show that the proposed PGHA possesses better ability of finding the global optimum compared to the GA and PSO algorithms.

Tseng and Liang [22] proposed a hybrid approach that combines ACO, GA and Local Search (LS). In order to provide a good start, GA has an initial population constructed by ACO. Pheromone acts as a feedback mechanism from GA phase to ACO phase. When GA phase reaches the termination criterion, control is transferred back to ACO phase. Then ACO utilizes pheromone updated by GA phase to explore solution space and produces a promising population for the next run of GA phase. The local search method is applied to improve the solutions obtained by ACO and GA.

EA can be combined with other meta-heuristics algorithm such as Imperialist Competitive Algorithm (ICA). The main idea is based on the human communities. The ICA was proposed by Atashpaz-Gargari and Lucas [23]. ICA is the mathematical model and the computer simulation of human social evolution. It is one of the recent meta-heuristic algorithms based on a socio-political idea, proposed to solve optimization problems [23]. ICA can be thought of as the social counterpart of GA because it uses imperialism and imperialistic competition, socio-political evolution process, as the source of inspiration.

People get together in organized groups or similar close aggregations. In each country people try to reach better positions in the society such as job, religious, economic and culture promotion. Likewise, it occurs among different countries. People who live together work with the leader to be better. Therefore it is a good idea to combine EA and ICA algorithms to increase performance. A new approach uses EA in low-level based on person and competition between people, and ICA in high-level based on country and competition between countries.

Abdechiri and Meybod [24] 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. Khorani et al. [25] 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 [26] 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 [27] 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.

The paper is organized as follows: Section 2 provides a brief literature overview of the EA and ICA. In Section 3, new approach and the motivation of the SBA is presented. In Section 4, results are compared with other Evolutionary Algorithms.

2. Background

In the last years, a new kind of approximate algorithm has emerged which basically tries to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space. These methods are commonly called meta-heuristics nowadays [28]. 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. ICA simulates the social political process of imperialism and imperialistic competition. ICA and EA will introduce in this section.

2.1. Evolutionary Algorithm (EA)

Evolutionary Algorithms (EAs) are population-based meta-heuristic optimization algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection, and survival of the fittest in order to refine a set of solution candidates iteratively [29]. There are numerous of EAs. They all share a common idea. 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 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.

2.2. 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* [23].

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 [23]. Algorithm continues until a stop condition is satisfied such as just one imperialist will remain.

```
Procedure SBA
Step 1: Initializing parameters;
Step 2:
  2. 1. Define the optimization problem;
  2. 2. Generate some random people;
  2. 3. Select some powerful random people as leaders;
  2. 4. Randomly allocate remain people to different countries;
  2. 5. Initialize the empires with imperialists cost function T.P_{C_i}
  2. 6. Select more powerful leaders as the empires;
Step 3: Decade loop: N_d = N_d + 1
Step 4: For i = 1,2,...,N_country do:
     4. 1. Selection;
     4. 2. Crossover;
     4. 3. Mutation;
     4. 4. Replacement;
Step 5: For i = 1,2,...,N_imp do:
     5. 1. People assimilation policy: Move the people of each
        country toward their relevant leaders, using:
                      x~U(0, Coeff_internal assimilation × d)
        d is the distance between person and leader.
     5. 2. People revolutionary;
     5. 3. Countries assimilation policy: Move the leaders of each
        country toward their empires and move the people of each country
        as the same as their leaders, using:
                      x~U(0, Coeff_external assimilation × d)
        d is the distance between leader and imperialist.
     5. 4. Countries revolutionary;
     5. 5. Change position;
     5. 6. Imperialistic Competition; Pick the weakest country from
        the weakest empire and give it to the empire that has the most
        likelihood to possess it.
     5. 7. Elimination; Eliminate the powerless empires and countries.
Step 6: Terminating Criterion Control; Repeat Steps 3-6 until a
        terminating criterion is satisfied.
```

Fig. 1. The pseudo-code of the SBA.

In this stage the position of imperialist and its colonies will be the same.

3. Social-Based Algorithm (SBA)

It can be concluded that EA and ICA have two different approaches for optimization in Section 2. In this section, combines two algorithms to present a novel hybrid algorithm. Fig. 1 presents pseudo-code of the SBA. Figs. 2 and 3 show operators among the people and countries. Initializing parameters are:

- n : the dimension of search space;
- $N_{monarchy}$: number of initial monarchy countries;
- $N_{republic}$: number of initial republic countries;
- $N_{autocracy}$: number of initial autocracy countries;
- $N_{multinational}$: number of initial multinational countries;
- $Coff_{external\,assimilation}$: external assimilation coefficient;
- $Coff_{external\,assimilation}$: internal assimilation coefficient;
- P_m: the mutation rate;
- P_c: the crossover rate;
- P_e: the external revolution rate;
- P_i: the internal revolution rate;
- N_{imp}: number of initial imperialist;
- N_{person}: number of people;
- N_d: number of decades.

The steps of the proposed solution algorithm are described in the following sub-sections.

3.1. 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. People form different communities in this approach. A person is a vector of size n in

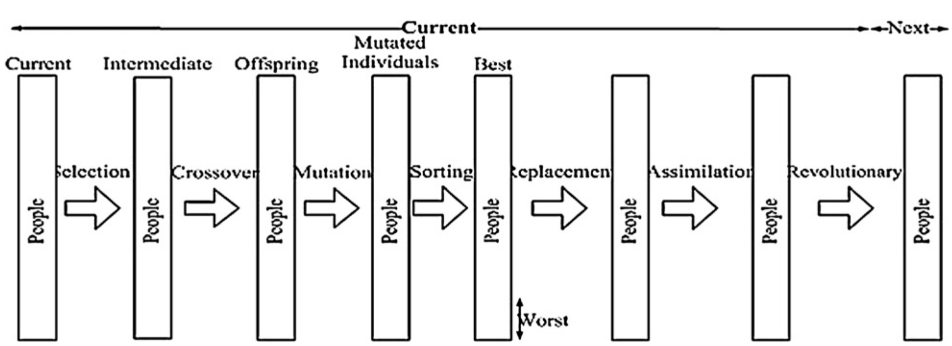

Fig. 2. A schematic of people operators.

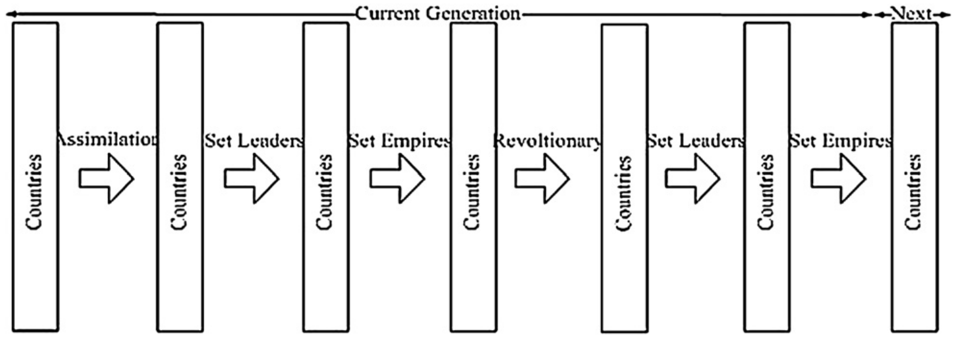

Fig. 3. A schematic of countries operators.

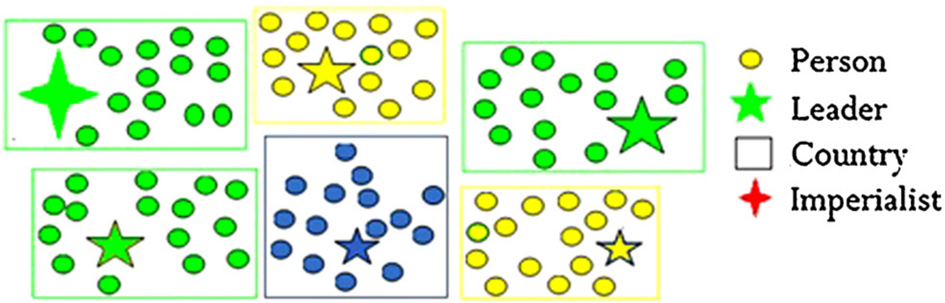

Fig. 4. Generating the initial empires and countries: the more countries an imperialist possess, the bigger is its relevant ⋆ mark.

an n dimensional optimization problem. This array is defined as Eq. (1):

$$Person = [x_1, \ldots, x_n] \tag{1}$$

The cost of a person is found by evaluating the cost function f at the variables $(x_1, \ldots, x_n)$:

$$C_{pi} = f(Person_i) \tag{2}$$

C_{pi} is the cost of ith person.

3.2. 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. $N_{Candidate}$ is number of candidates.
- *Autocracy*: This type of community does not have leader. The people are free and exists 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.
- *Multinational*: Multinational community that manages production or delivers services in more than one country [30]. The constitutions of these communities claim that all power belongs to the working class. They produce goods in different part of country, and send them to the others.

3.3. Initialization

The algorithm starts with a random population called people. N_{Person} is the total number of people. According to the parameters calculate their fitness. Then select $N_{Country}$ of the best people in the population to be the leaders and the rest form the people of these countries. Total number of countries is calculated from Eq. (3):

$$N_{Country} = N_{Monarchy} + N_{Republic} + N_{Autocracy} + N_{Multinational} \tag{3}$$

To form the initial countries, the people are divided among countries based on their power. Different types of initialization compare as follows:

- *ICA-Based*: Based on original ICA initialization, some of the best initial countries (the countries with the least cost function value),

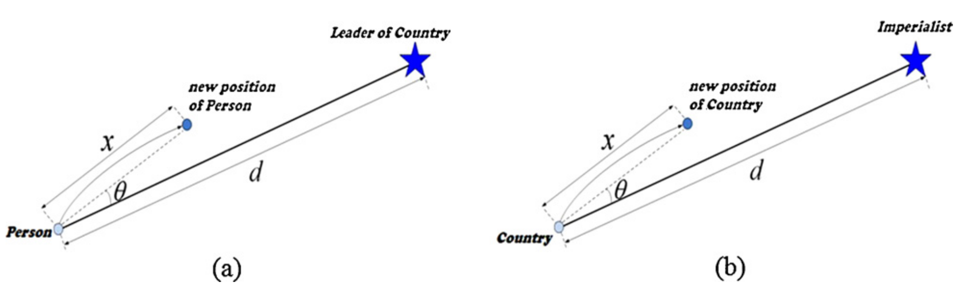

Fig. 5. Assimilation: (a) internal and (b) external.

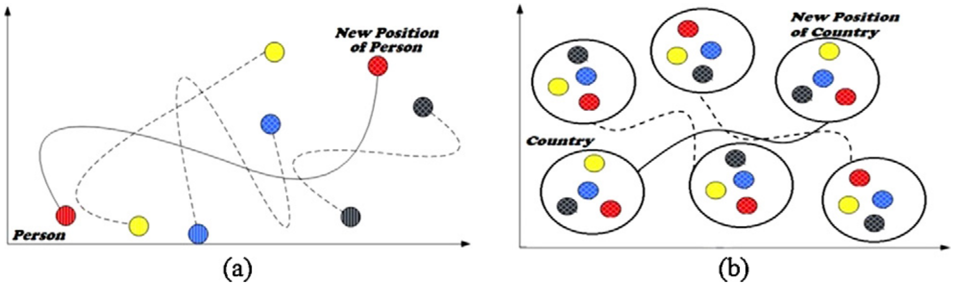

Fig. 6. Revolution: (a) internal and (b) external.

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. To form the initial countries, the people are divided among leaders based on their power. The initial number of people of a leader should be directly proportionate to its power. To proportionally divide the people among leaders, the normalized cost of a leader is defined by Eq. (4).

$$N \cdot C_i = \max_j \{Cost(Leader_j)\} - Cost(Leader_i) \tag{4}$$

In this equation $Cost(Leader_j)$ is the ith leader's cost and $N \cdot C_i$ is its normalized cost. With normalized cost of all leaders, the initial people can be divided among leader according to the following equation:

$$P \cdot C_i = round \left\{ \frac{N \cdot C_i \times N_{Person}}{\sum_{j=1}^{N_{Country}} N \cdot C_j} \right\} \tag{5}$$

where $P \cdot C_i$ is the initial number of people of the ith leader. The people are randomly chosen and given to the ith leader and the people along with the ith leader form ith country. The better leaders have greater number of people while weaker ones have less.Similarly, to form the initial empire, the countries are divided among empires based on their power. Fig. 4 shows the initial population of each empire. As shown in this figure bigger empires have greater number of countries while weaker ones have less. In this figure the most powerful empire has the greatest number of countries. It is referred to SBA-1 in results.

- *Frog-Based*: In order to partition frogs into N_m memeplexes which containing N_f members, in Shuffled Frog Leaping (SFL) algorithm [31], population is sorting a decreasing order in terms of function evaluation value, 1st frog ranking goes to memeplex$_1$, 2nd frog ranking$_2$ goes to memeplex$_2$, . . ., N_mth frog ranking goes to memeplex N_m. Then second member of each subset is assigned as: $(N_m + 1)$th frog ranking goes to memeplex1, $(N_m + 2)$th frog ranking goes to memeplex$_2$, . . ., $(N_m + N_m)$ frog ranking goes to memeplex N_m. This process continues to assign all frogs into all memeplexes. It is referred to SBA-2 in results.
- *Hybrid-ICA-Frog-Based:* In these strategies, it combines probabilistic idea of ICA with ordering idea of frog and give the ordered people to each country based on the described power. It is referred to SBA-3 in results.
- *Classification-Based*: People in the geographical position lives together and define new countries it means location is affect the countries of the people. People choose nearest country to live it based on the distance with the leader of country. In order to classify the person use two Eqs. (6) and (7):

$$distance(i, j) = \sqrt{\sum_{t=1}^{n} (person_i \cdot x_t - leader_j \cdot x_t)^2} \tag{6}$$

$$distance(i, j) = \sum_{t=1}^{n} \left| person_i \cdot x_t - leader_j \cdot x_t \right| \tag{7}$$

$leader_j \cdot x_t$ is the tth position of leader of jth country and $person_i \cdot x_t$ is the tth position of jth person. They are referred to SBA-4 and SBA-5 in results.

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 of the country has an effect on the total power of that country:

$$T \cdot P_{Ci} = Cost(Leader_i) + \xi mean\{Cost(People\ of\ country_i)\} \tag{8}$$

$T \cdot P_{Ci}$ is the total power of ith country and ξ is a positive number which is considered to be less than 1. A little value for ξ causes the total power of the leader to be determined by just the leader and increasing it will add to the role of the people in determining the total power of a leader. The value 0.1 for ξ is a proper value in most of the implementation.

After forming initial empires, in each decade's Evolutionary Algorithm operator such as selection, crossover and mutation begins on people of each country. These operators does not effect

Table 1
Benchmark functions of CEC2010.

Seperable		Single group m-nonseperable		$\frac{D}{2m}$-group m-nonseparable		$\frac{D}{2m}$-group m-nonseparable		Nonseperable	
$F1$	Shifted Elliptic	$F4$	Single-group Shifted and m-rotated Elliptic	$F9$	$\frac{D}{2m}$-group Shifted and m-rotated Elliptic	$F14$	$\frac{D}{2m}$-group Shifted and m-rotated Elliptic	$F19$	Shifted Schwefel
$F2$	Shifted Rastrigin	$F5$	Single-group Shifted m-rotated Rastrigin	$F10$	$\frac{D}{2m}$-group Shifted and m-rotated Rastrigin	$F15$	$\frac{D}{2m}$-group Shifted and m-rotated Rastrigin	$F20$	Shifted Rosenbrock
$F3$	Shifted Ackley	$F6$	Single-group Shifted m-rotated Ackley	$F11$	$\frac{D}{2m}$-group Shifted and m-rotated Ackley	$F16$	$\frac{D}{2m}$-group Shifted and m-rotated Ackley		
		$F7$	Single-group Shifted m-dimensional Schwefel	$F12$	$\frac{D}{2m}$-group Shifted m-dimensional Schwefel	$F17$	$\frac{D}{2m}$-group Shifted m-dimensional Schwefel		
		$F8$	Single-group Shifted m-dimensional Rosenbrock	$F13$	$\frac{D}{2m}$-group Shifted m-dimensional Rosenbrock	$F18$	$\frac{D}{2m}$-group Shifted m-dimensional Rosenbrock		

Table 2
The results achieved by SBA on the test suite.

FES	1.20E+05				6.00E+05				3.00E+06			
Algorithm	PSO	SDENS [31]	jDElsgo [30]	SBA	PSO	SDENS [31]	jDElsgo [30]	SBA	PSO	SDENS [31]	jDElsgo [30]	SBA
*F*1	1.26E+10	5.01E+09	3.70E+09	**9.76E+08**	5.36E+08	7.87E+06	**8.99E+04**	2.49E+06	1.99E+07	5.73E−06	**8.86E−20**	3.49E+04
	(3.61E+09)	(9.81E+08)	(5.11E+08)	**(1.82E+08)**	(1.73E+08)	(5.94E+06)	**(1.39E+04)**	(4.25E+05)	(1.74E+07)	(4.46E−06)	**(4.51E−20)**	(2.72E+03)
*F*2	2.36E+04	1.19E+04	1.09E+04	**6.29E+03**	2.36E+04	7.09E+03	3.95E+03	**5.55E+02**	2.36E+04	2.21E+03	**1.25E−01**	2.62E+00
	(4.06E+02)	(9.89 + 01)	(1.75E+02)	**(1.84E+02)**	(4.06E+02)	(6.76E+01)	(1.32E+02)	**(3.52E+01)**	(4.06E+02)	(8.95E+01)	**(3.45E−01)**	(1.69E+00)
*F*3	2.13E+01	2.01E+01	**1.87E+01**	2.04E+01	2.05E+01	6.12E+00	**1.22E+00**	1.39E+01	1.99E+01	2.70E−05	**3.81E−12**	1.62E−02
	(2.84E−02)	(1.17E−01)	**(4.46E−01)**	(4.12E−02)	(4.06E−02)	(6.30E−01)	**(1.38E−01)**	(1.04E+00)	(1.03E−02)	(1.54E−05)	**(5.02E−12)**	(3.17E−02)
*F*4	**4.06E+13**	5.10E+13	1.40E+14	4.10E+13	1.31E+13	1.72E+13	1.39E+13	**8.42E+12**	5.72E+12	5.11E+12	**8.06E+10**	3.10E+12
	(1.94E+13)	(1.46E+13)	(3.69E+13)	(1.64E+13)	(4.45E+12)	(6.68E+12)	(4.60E+13)	**(4.07E+12)**	(1.84E+12)	(2.16E+12)	**(3.08E+10)**	(1.02E+12)
*F*5	3.35E+08	3.29E+08	3.39E+08	3.58E+08	3.34E+08	**1.81E+08**	1.88E+08	3.44E+08	3.33E+08	1.18E+08	**9.72E+07**	3.44E+08
	(9.00E+07)	(1.04E+07)	(1.82E+07)	(6.18E+07)	(9.02E+07)	**(2.29E+07)**	(2.31E+07)	(5.96E+07)	(9.01E+07)	(2.88E+07)	**(1.44E+07)**	(5.96E+07)
*F*6	9.30E+06	**1.84E+06**	4.26E+06	1.34E+07	8.98E+06	**1.53E+01**	5.07E+01	7.92E+06	8.19E+06	2.02E−04	**1.70E−08**	5.79E+06
	(8.34E+06)	**(4.77E+05)**	(3.81E+05)	(3.38E+06)	(8.40E+06)	**(1.18E+00)**	(5.81E+01)	(4.92E+06)	(7.97E+06)	(4.29E−05)	**(4.03E−08)**	(4.92E+06)
*F*7	**1.15E+10**	3.75E+10	5.39E+10	7.39E+10	**8.36E+07**	9.28E+09	6.43E+09	2.73E+10	1.62E+05	1.20E+08	**1.31E−02**	2.13E+09
	(4.04E+09)	(5.46E+09)	(1.07E+10)	(2.43E+10)	**(2.46E+07)**	(3.44E+09)	(2.12E+09)	(1.67E+10)	(1.03E+05)	(6.56E+07)	**(6.38E−02)**	(2.57E+09)
*F*8	**5.55E+08**	7.71E+08	2.39E+09	8.31E+09	1.19E+08	**7.41E+07**	6.82E+07	1.16E+09	6.76E+07	5.12E+07	**3.15E+06**	5.25E+07
	(5.42E+08)	(2.27E+08)	(9.13E+08)	(1.50E+10)	(6.60E+07)	**(2.73E+07)**	(3.53E+07)	(3.08E+09)	(2.56E+07)	(2.12E+07)	**(3.27E+06)**	(3.82E+07)
*F*9	3.74E+10	1.56E+10	1.64E+10	**3.54E+09**	8.82E+09	2.23E+09	1.66E+09	**9.55E+08**	1.87E+09	5.63E+08	**3.11E+07**	3.00E+08
	(4.67E+09)	(2.77E+09)	(1.73E+09)	**(3.54E+09)**	(1.02E+09)	(3.70E+08)	(8.29E+07)	**(8.56E+07)**	(3.19E+08)	(5.78E+07)	**(5.00E+06)**	(3.22E+07)
*F*10	2.42E+04	1.39E+04	1.43E+04	**1.10E+04**	2.42E+04	1.10E+04	8.67E+03	**7.56E+03**	2.42E+04	6.87E+03	**2.64E+03**	7.28E+03
	(3.31E+02)	(2.51E+02)	(4.38E+02)	**(3.81E+02)**	(3.31E+02)	(4.59E+02)	(3.99E+02)	**(3.82E+02)**	(3.31E+02)	(5.60E+02)	**(3.19E+02)**	(3.78E+02)
*F*11	2.35E+02	2.27E+02	**2.19E+02**	2.26E+02	2.32E+02	2.26E+02	**1.17E+02**	2.20E+02	2.23E+02	2.21E+02	**2.20E+01**	2.01E+02
	(5.08E−01)	(3.49E−01)	**(5.92E+00)**	(8.19E−01)	(3.75E+00)	(3.83E−01)	**(1.87E+01)**	(5.94E−01)	(6.02E+00)	(5.09E−01)	**(1.53E+01)**	(1.63E+00)
*F*12	6.24E+06	2.95E+06	3.15E+06	**2.01E+06**	2.10E+06	1.32E+06	9.39E+05	**7.80E+05**	7.37E+05	4.13E+05	**1.21E+04**	2.46E+05
	(6.28E+05)	(2.37E+05)	(2.19E+05)	**(1.78E+05)**	(2.38E+05)	(5.98E+04)	(2.96E+04)	**(6.56E+04)**	(1.07E+05)	(4.28E+04)	**(2.04E+03)**	(1.81E+04)
*F*13	3.50E+10	1.88E+10	3.76E+09	**1.89E+07**	3.89E+09	6.43E+05	5.32E+04	**3.10E+04**	5.66E+08	2.19E+03	**7.11E+02**	1.67E+04
	(1.15E+10)	(1.07E+09)	(1.04E+09)	**(7.05E+06)**	(8.69E+09)	(1.10E+05)	(1.70E+04)	**(7.35E+03)**	(2.56E+09)	(1.03E+03)	**(1.37E+02)**	(8.18E+03)
*F*14	3.98E+10	1.84E+10	2.32E+10	**1.36E+10**	1.40E+10	5.14E+09	4.10E+09	**3.61E+09**	5.99E+09	1.88E+09	**1.69E+08**	1.09E+09
	(1.19E+10)	(3.56E+09)	(2.03E+09)	**(1.20E+09)**	(4.19 + 09)	(9.89E+08)	(2.89E+08)	**(3.44E+08)**	(1.80E+09)	(2.33E+08)	**(2.08E+07)**	(7.80E+07)
*F*15	2.45E+04	**1.43E+04**	1.54E+04	1.54E+04	2.45E+04	**1.03E+04**	1.20E+04	1.47E+04	2.45E+04	7.32E+03	5.84E+03	1.47E+04
	(3.39E+02)	**(3.72E+02)**	(3.33E+02)	(3.35E+02)	(3.39E+02)	**(2.29E+03)**	(5.30E+02)	(3.65E+02)	(3.39E+02)	(9.63 + 01)	(4.48E+02)	(3.65E+02)
*F*16	4.29E+02	**4.15E+02**	4.17E+02	4.21E+02	4.25E+02	4.13E+02	**2.99E+02**	4.13E+02	4.09E+02	4.08E+02	**1.44E+02**	4.01E+02
	(7.51E−01)	**(1.08E−01)**	(3.28E+00)	(4.95E−01)	(4.78E+00)	(3.49E−01)	**(1.91E+01)**	(7.61E−01)	(1.26E+01)	(2.53E+00)	**(3.43E+01)**	(3.50E−01)
*F*17	6.34E+06	4.31E+06	4.85E+06	**3.99E+06**	3.81E+06	2.07E+06	1.95E+06	**1.67E+06**	3.50E+06	1.08E+06	**1.02E+05**	5.13E+05
	(6.41E+05)	(4.04E+05)	(3.53E+05)	**(2.27E+05)**	(4.87E+05)	(1.17E+05)	(6.54E+04)	**(6.67E+04)**	(4.51E+05)	(1.11E+05)	**(1.26E+04)**	(2.48E+04)
*F*18	4.01E+11	2.11E+11	6.60E+10	**4.02E+10**	8.21E+10	2.02E+08	**1.03E+06**	6.80E+06	3.53E+10	3.08E+04	**1.85E+03**	4.78E+04
	(7.59E+10)	(1.27E+10)	(9.47E+09)	**(4.40E+09)**	(3.46E+10)	(5.02E+07)	**(2.08E+05)**	(2.43E+06)	(2.18E+10)	(1.22E+04)	**(3.18E+02)**	(1.47E+04)
*F*19	3.23E+07	1.67E+07	2.85E+07	**9.92E+06**	1.72E+07	**5.41E+06**	6.09E+06	5.54E+06	9.97E+06	8.80E+05	**2.74E+05**	2.99E+06
		(3.71E+06)	(3.38 + 06)	**(1.00E+06)**		**(4.31E+05)**	(1.65E+06)	(3.25E+05)		(1.59E+05)	**(2.12E+04)**	(1.26E+05)
*F*20	9.91E+11	2.61E+11	7.99E+10	**2.67E+10**	1.00E+11	2.69E+08	1.01E+06	**9.61E+05**	8.55E+09	**9.90E+02**	**1.53E+03**	3.73E+03
	(8.71E+10)	(1.49E+10)	(1.25E+10)	**(5.44E+09)**	(2.77E+10)	(7.57E+07)	(2.48 + 05)	**(4.24E+05)**	(6.63E+09)	**(1.62E+01)**	**(1.32E+02)**	(1.51E+03)

Table 3
Benchmark function ($F21$–$F33$).

Title	Function	Range
$F21$ (Sphere)	$\sum_{i=1}^{n} x_i^2$	$-5.12 \le x_i \le 5.12$
$F22$ (Rosenbrock)	$\sum_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2 + (1+x_i)^2$	$-2.048 \le x_i \le 2.048$
$F23$	$x \sin(4x) + 1.1y \sin(2y)$	$0 \le x,\ y \le 10$
$F24$ (Rastrigin)	$\sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$-5.12 \le x_i \le 5.12$
$F25$ (Griewangk)	$1 + \sum_{i=1}^{n} \left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{2} \left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$	$-600 \le x_i \le 600$
$F26$ (Schwefel)	$\sum_{i=1}^{n} 418.9829 - x_i \sin(\sqrt{\lvert x_i \rvert})$	$-500 \le x_i \le 500$
$F27$	$\sum_{i=1}^{n} 10^{i-1} x_i^2$	$-10 \le x_i \le 10$
$F28$ (Schaffer)	$0.5 + \frac{\sin^2\sqrt{x^2+y^2} - 0.5}{(1+0.001(x^2+y^2))^2}$	$-100 \le x,\ y \le 100$
$F29$	$x \cdot sgn(x)$	$-1.0 \le x \le 2.0$
$F30$ (Schwefel 1.2)	$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	$-100 \le x,\ y \le 100$
$F31$ (SumSquares)	$\sum_{i=1}^{n} i^2 x_i^2$	$-1 \le x_i \le 1$
$F32$ (Ackley)	$-20e - 0.2\sqrt{\frac{1}{D}\sum_{i=1}^{n} x_i^2} - e^{\frac{1}{D}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 + e$	$-30 \le x_i \le 30$
$F33$	$\sum_{i=1}^{n} [x_i]$	$-5.12 \le x_i \le 5.12$

Table 4
Parameters values of PSO and ABC.

	Parameter	Value	Parameter	Value
PSO	Cognitive acceleration	2	The fraction of maximum iterations	0.7
	Social acceleration	2	Maximum velocity step	100
	Neighborhood acceleration	1	Constriction factor	1
	Velocity weight at the beginning	0.95	Velocity weight at the end	0.4
ABC	The number of food sources	Population size/2	Food source[a]	Population size $\times n/2$

[a] A food source which could not be improved through "limit" trials is abandoned by its employed bee.

on countries' leaders. After that ICA starts and the countries in each empire start moving toward their relevant imperialist in monarchy countries. This movement is a simple model of assimilation policy which was pursued by some of the imperialist states. Revolution brings about sudden random changes in the position of some of the countries in the search space. After that in each country calculate the cost of the new people and whom are changed in moving process, if the person reach to a position with better cost value than its leader. Similarly it is done in each empire to select better leader as a new empire. In competition, one of the weakest countries of the

Table 5
Parameters values of compared algorithms.

Algorithm	Parameter	Value	Parameter	Value
GA and GA–BF [35]	Mutation rate	0.9	Crossover rate	0.1
	Step size	1E−007	Chemotactic step	1000
GSA [36]	G_0	100	α	20
	K_0	Population size	Population size	50
CICA and ICA [37]	Number of imperialist	8	Number of colonies	80
	α	0.5	β	2
	b	0.2		
PSO [37]	c_1	1.5	c_2	1.5
	Number of particles	80		
RGA [36]	Crossover rate	0.3	Mutation rate	0.1
	Crossover	Arithmatic	Mutation	Gaussian
	Selection	Roulette wheel		

Table 6
Comparing SBA with ABC and PSO.

F	n	Pop.	Gen.	ABC	PSO	SBA	P_{ABC}	P_{PSO}
F21	100	500	1000	8.59E−05	1.99E+00	**1.27E−10**	100	100
	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
F22	100	500	1000	2.22E+02	3.15E+04	**1.54E+02**	31	100
	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
F24	100	500	1000	5.39E+01	6.05E+02	**2.00E−07**	100	100
	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
F25	100	500	1000	9.04E−03	5.24E−01	**4.92E−03**	46	99
	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
F31	100	500	1000	1.11E−03	1.39E+02	**1.40E−08**	100	100
	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
F32	100	500	1000	2.12E+00	3.37E+00	**5.60E−05**	100	100
	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

weakest empires is picking and making a competition among all empires to possess this country. Each empire will have a likelihood of taking possession of the mentioned country, the probability of which is proportional to its power calculated like Eq. (8). A country or empire is assumed to be collapsed when they lose all of their people or countries.

3.4. Assimilation and revolution

After Evolutionary Algorithm operator accomplishes, imperialist starts to improve their countries and countries started to improve their people. SBA 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. The leaders move toward the empires by x units. The new position of leader is calculated from Eq. (9).

$$x \sim U(0, Coeff_{external\,assimilation} \times d) \quad (9)$$

x is a random variables with uniform between 0 and 0 and $Coeff_{external\,assimilation} \times d$ and $Coeff_{external\,assimilation}$ is a number greater than 1 and d is the distance between leader and empire. They do not move straight toward their empire. ICA used the angle of countries movement toward imperialist's position to enhance the escaping capability from local optima trap. θ is defined as deflection angle and represents the direction of movement (Eq. (10)).

$$\theta \sim U(-\gamma, \gamma) \quad (10)$$

where γ is a parameter that adjusts the deviation from the original direction. In most of implementation a value of about $\pi/4$ (rad) for γ results in good convergence of countries to the global optimum. 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.

The internal revolution operation is performed for each person with revolution probability P_i and the external revolution operation is performed for each country with revolution probability P_e. Figs. 5 and 6 show internal and external forms of assimilation and revolution.

Internal assimilation in multinational communities is different; it is implemented in two steps. In one step each person in ith country moves ith position toward the ith position of its leader (Eq. (11)) and in the other step, each part of the point to be produced is spread out over different country.

$$x_i \sim U(0, \beta \times d_i) \quad (11)$$

β is a number greater than 1 and d_i is the distance between ith position of person and leader.

It defines N_c countries for each community because each country optimizes its own part of variable. The number of these communities is coefficient of variable size (Eq. (12)).

$$N_{Multinational} = k \times N_c \quad (12)$$

k is constant input parameter. Each country tries to improve their given part. After produce different part in multinational community they export their product to others. It means each person receive set of $[1, \ldots, i-1, i+1, \ldots, N_c]$ parts from other countries (Eq. (13)):

$$X_j = leader_j.x_j \quad (13)$$

$leader_j \cdot x_j$ is the jth position of leader of jth country.

4. Evaluation and experimental results

For evaluating performance of the proposed algorithm, the simulation results are compared with results of EA and ICA. Performance parameter calculated from Eq. (14) as follow for minimum optimization:

$$P_{Algorithm} = 100 \times \left(1 - \frac{M_{SBA}}{M_{Algorithm}}\right) \quad (14)$$

Table 7
Comparing different initilized SBA with EA, ICA, PSO and ABC.

F	*F*21			*F*22			*F*23	*F*24			*F*25			*F*26		
n	10	20	50	10	20	30	2	10	20	50	10	20	50	10	20	50
Community	15	20	20	10	25	50	5	15	20	20	45	60	75	15	20	30
Generation	1000	1200	2000	2000	3000	4000	300	1000	1200	2000	1000	1500	3000	1000	1000	2000
Population	150	300	500	250	500	600	20	150	200	400	450	600	750	150	200	300
EA	9.74E−03	2.53E−01	5.15E+00	6.50E+00	3.47E+01	7.42E+01	−18.3454	1.27E+00	9.55E+00	7.81E+01	2.41E−02	1.06E+00	1.06E−02	8.52E+00	1.69E+02	1.58E+03
ICA	6.04E−04	2.33E−03	8.18E−03	8.49E−01	2.70E+00	1.11E+01	–	1.23E−01	7.46E−01	2.46E+00	1.77E−03	7.41E−E−03	2.15E−02	9.57E−01	7.27E+00	9.59E+01
PSO	2.30E−35	3.61E−21	2.14E−12	1.33E+00	9.36E+00	3.32E+01	–	1.69E+00	1.18E+01	6.72E+01	5.68E−02	2.72E−02	1.00E−02	6.88E+02	2.36E+03	7.76E+03
ABC	7.30E−17	2.49E−16	1.14E−15	2.60E−03	1.95E−02	5.79E+00	−18.5547	0	0	5.79E−12	1.67E−17	0	**2.22E−17**	2.90E−02	**4.20E−02**	**3.19E+02**
SBA-1	1.36E−39	2.31E−32	8.88E−23	3.23E−03	**1.75E−03**	7.06E+00	−18.5547	0	0	0	0	5.55E−18	3.33E−16	**1.27E−04**	2.55E−04	6.36E−04
SBA-2	1.36E−37	6.63E−33	4.00E−26	**3.95E−04**	2.58E−03	8.08E+00	−18.5547	0	**0**	0	**0**	5.55E−18	2.72E−16	1.27E−04	2.55E−04	6.36E−04
SBA-3	4.82E−40	**2.20E−33**	**2.00E−23**	5.76E−04	5.78E−03	6.10E+00	−18.5547	0	0	**0**	0	**0**	2.05E−16	1.27E−04	2.55E−04	6.36E−04
SBA-4	**3.23E−42**	1.86E−25	3.68E−17	1.59E−01	8.61E−01	8.67E−01	−**18.5547**	0	0	0	0	3.89E−17	2.50E−13	1.27E−04	2.55E−04	6.36E−04
SBA-5	2.41E−40	3.14E−28	2.52E−17	2.43E−01	5.59E−01	**8.42E−01**	−18.5547	**0**	0	1.78E−16	0	5.55E−17	4.89E−10	1.27E−04	2.55E−04	6.36E−04

F	*F*27			*F*28	*F*29	*F*30			*F*31			*F*32		
n	10	20	50	2	1	10	20	50	10	20	50	10	20	50
Community	15	20	45	15	10	15	20	20	15	20	20	15	20	20
Generation	1000	1200	2500	1000	500	1000	1200	1500	1000	1200	2000	1000	1200	2000
Population	150	200	450	150	100	150	200	200	150	200	500	150	200	200
EA	3.63E+05	1.31E+16	1.55E+45	2.09E−02	6.08E−04	1.09E+01	5.36E+02	3.73E+04	1.01E−02	5.75E−01	1.12E+02	1.52E+00	3.63E+00	6.03E+00
ICA	1.78E+01	1.18E+07	1.20E+28	1.34E−08	2.64E−06	1.00E+00	7.70E+00	1.19E+02	4.79E−04	5.71E−03	9.53E−02	2.74E−01	4.76E−01	7.15E−01
PSO	8.18E−16	2.91E−06	2.16E+11	3.45E−04	4.52E−16	–	–	–	2.61E−35	3.71E−20	7.73E−11	4.26E−15	2.80E−10	1.12E−03
ABC	6.32E−17	1.27E−12	3.19E+02	6.88E−06	6.02E−19	–	–	–	7.26E−17	2.68E−16	1.57E−15	6.93E−15	1.34E−12	2.02E−07
SBA-1	**5.80E−37**	**6.53E−15**	1.11E−01	**0**	7.95E−76	2.79E−39	6.29E−28	9.06E−12	**5.76E−39**	4.12E−27	8.77E−23	4.00E−15	2.85E−12	6.51E−11
SBA-2	1.00E−34	6.28E−11	1.17E+01	1.48E−03	**1.13E−118**	1.59E−35	1.84E−27	3.01E−12	2.18E−36	**4.90E−30**	**5.94E−24**	4.00E−15	1.03E−11	1.58E−10
SBA-3	6.51E−35	2.90E−14	**1.89E−02**	4.86E−04	2.16E−95	4.92E−40	**3.52E−28**	4.00E−11	9.52E−38	7.61E−28	1.51E−23	**4.00E−15**	2.12E−12	**1.01E−10**
SBA-4	7.42E−15	2.41E−05	1.28E+15	6.19E−07	3.73E−32	**7.37E−40**	1.61E−24	**1.94E−13**	3.57E−30	5.09E−25	6.10E−17	2.29E−13	**4.61E−14**	1.13E−09
SBA-5	7.35E−15	2.95E−05	7.37E+14	4.87E−04	2.42E−28	4.68E−38	2.26E−23	6.03E−13	1.76E−36	2.74E−24	2.39E−17	4.00E−15	7.49E−14	8.70E−10

Table 8
Statistical comparison of benchmark functions ($n = 10$).

F	$F21$		$F22$		$F24$		$F25$		$F26$		$F27$		$F30$		$F31$		$F32$	
Algorithm	EA	SBA	EA	SBA	EA	SBA	EA	SBA	EA	SBA	EA	SBA	EA	SBA	EA	SBA	EA	SBA
Best	2.32E−03	2.33E−59	4.94E−01	1.05E−04	2.92E−01	0	8.43E−01	0	4.32E+00	1.27E−04	3.34E+03	1.45E−49	2.31E+00	1.66E−53	1.23E−03	2.47E−53	3.25E−01	4.00E−15
Median	7.92E−03	1.87E−46	6.83E+00	8.32E−04	1.21E+00	0	1.06E+00	0	8.16E+00	1.27E−04	1.35E+05	1.12E−40	1.08E+01	9.68E−45	8.20E−03	3.09E−46	1.62E+00	4.00E−15
Worst	3.02E−02	2.71E−38	8.46E+00	2.67E−02	2.36E+00	0	1.20E+00	0	1.79E+01	1.27E−04	1.85E+06	7.64E−36	2.61E+01	4.34E−38	3.90E−02	8.84E−38	2.47E+00	4.00E−15
Mean	9.74E−03	1.36E−39	6.50E+00	3.23E−03	1.27E+00	0	1.06E+00	0	8.52E+00	1.27E−04	3.63E+05	5.80E−37	1.09E+01	2.79E−39	1.01E−02	5.76E−39	1.52E+00	4.00E−15
Std.	7.20E−03	6.06E−39	1.90E+00	6.56E−03	5.34E−01	0	9.35E−02	0	3.48E+00	3.33E−13	5.49E+05	1.70E−36	6.38E+00	9.74E−39	7.99E−03	2.03E−38	6.21E−01	1.62E−30

4.1. Large scale global optimization

Benchmark functions of CEC2010 are listed in Table 1. In Table 2, the proposed SBA algorithm was tested on 20 benchmark functions provided by CEC2010 Special Session on Large Scale Global Optimization [32]. SBA performs better on 1.2E+04[1] and 6.0E+05 function evaluations (FEs). jDElsgo [33] perform better on 3.0E+06 FEs. In the jDElsgo parameters were set as follows: F was self-adaptive, initially set to 0.5, CR was self-adaptive, initially set to 0.9, number of population was adaptive, initial value is set to 100, $p_{\max}$ was fixed during the optimization and set to 4 [33]. The parameter settings of SDENS are described as follows: the population size was set to 50. The control parameters F and CR were set to 0.5 and 0.9. The $p_{\rm ns}$ was set to 0.05 [34].

4.2. Common benchmark functions

Used common benchmark functions are listed in Table 3.[2] The parameter settings of SBA are described as follows: selection method is Roulette Wheel Selection with real representation. Crossover method is set to uniform with the rate of 0.75. Mutation method is set to gene flip with the rate of 0.0505. Replacement is one-elitism generation. External and internal assimilation coefficients are set to 2.0. External and internal revolutionary rates are set to 0.1. The algorithm is conducted 20 runs for each test function (see Table 3).

In Table 6, the performance of SBA algorithm is compared with PSO and ABC for high dimensional problem. As shown, performance of SBA is better than PSO and ABC (parameters values of PSO, ABC and different algorithms are shown in Tables 4 and 5). In Table 7, performance parameter shows SBA perform better optimization, bold, than EA, ICA, PSO and ABC in all test functions. Table 8 illustrates statistical comparison of benchmark functions for dimension of 10. In Table 9, different type of community is investigated. *Monarchy* community perform better.

4.3. Effects of tuning parameters on the convergence of the SBA algorithm

The sensitivity of parameters is analyzed to select the best possible parameters for optimization. Effects of tuning parameters on the convergence of the SBA algorithm presented in different challenging functions. It is known that the global optimum of the considered functions is 0.

In Fig. 7, the $F21$ and $F32$ functions ($n = 10$) are considered as a benchmark for studies on different external and internal assimilation coefficients. In this figure, external and internal assimilation coefficients equal to 3 and 2.5 perform better. Comparing different internal and external assimilation coefficients are discussed in Table 13.

$F22$ fitness profile with respect to number of initial population, imperialist communities, monarchy countries and problem size are illustrated in Fig. 8. Emperor and monarchy count equal to 5 and 30 perform better. Comparing different emperor and monarchy count are discussed in Table 11.

A logarithmic scale is a scale of measurement using the logarithm of physical quantity instead of the quantity itself. Since, in this study, the values cover a wide range. The convergence curves are plotted with logarithmic scale for the y-axis in order to be able to capture the convergence trend over a wide range of values.

[1] $aE \pm b$ refers to $a \times 10^{\pm b}$.

[2] The performance of the proposed algorithm is specified in the average value of the solution obtained in all trails.

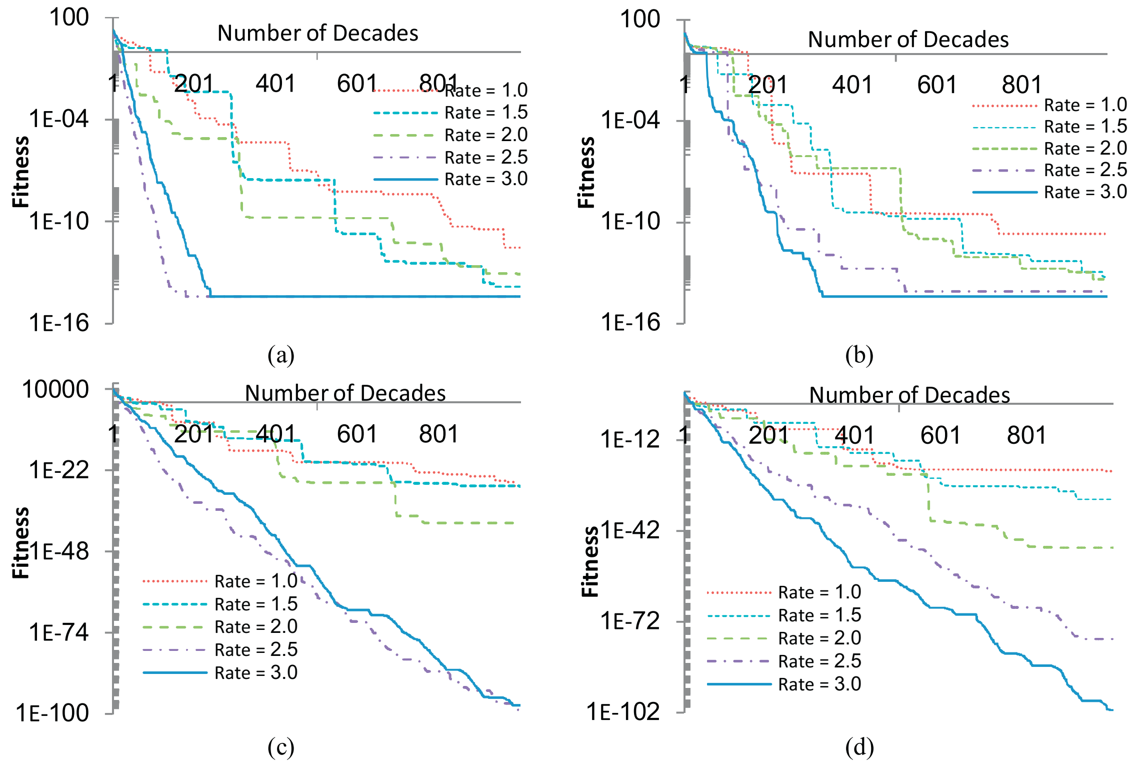

Fig. 7. External and internal assimilation coefficients: (a), (b) *F*32 ($n=10$), (c), (d) *F*21 ($n=10$).

Fig. 9 illustrates *F*30 convergence diagram for different dimensions including $n=10$, $n=20$, $n=50$ and $n=1000$. SBA reached near-optimal but in high dimensional it used more decades. Fig. 10 shows different convergence diagrams for EA, SBA without using EA and SBA algorithms. 2-Diminsional generated points, in decades 10, 50 and 100 are shown in Fig. 11 for *F*21 and *F*31. SBA converges to near-optimal points, rapidly.

In Table 10 the SBA are compared to different algorithms. Tables 11–13 show different parameters value: monarchy and emperor count, external and internal revolutionary rates and external and internal assimilation coefficients ($n=5$). Population size is 450 and size of generation is 100 for Table 11. Population size is 60 and size of decades is 500 for Tables 12 and 13. Revolutionary rates less than 0.5 perform better. External assimilation coefficient in [2,3.4] range and internal assimilation coefficient in [2,2.8] range perform better result. Count of emperor in range [3,7] percent of count of monarchy performs better. Count of monarchy in range [2,7] percent of population size performs better.

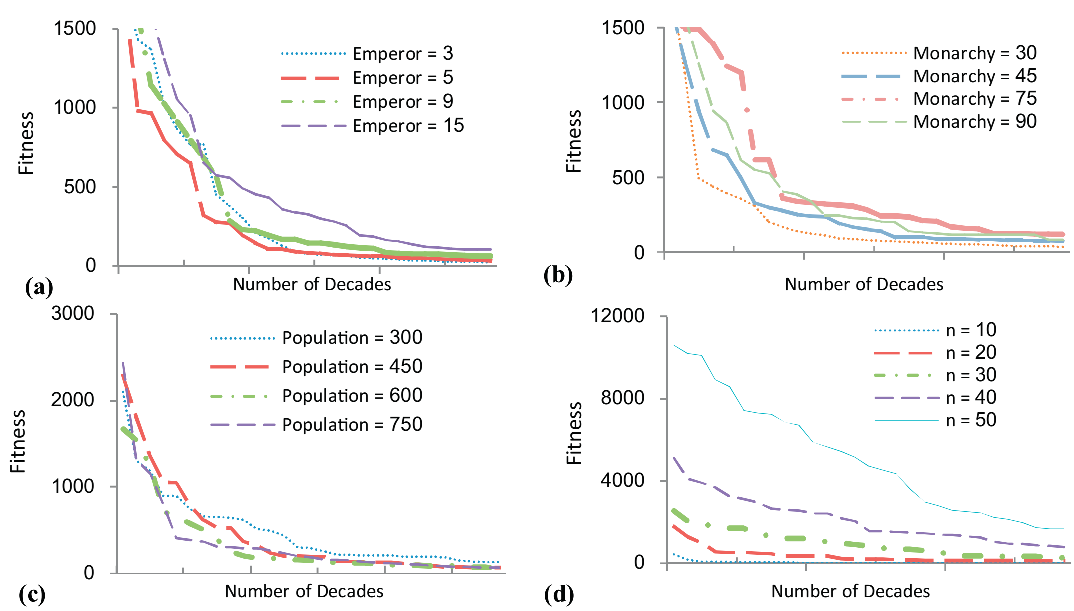

Fig. 8. *F*22's Parameter sensitivity diagrams: (a) emperor count, (b) monarchy count, (c) population size, and (d) dimension of problem.

Table 9
Comparing different communities (monarchy communities perform better).

F	*F*21			*F*22			*F*23	*F*24			*F*25			*F*26		
Dimension	10	20	50	10	20	30	2	10	20	50	10	20	50	10	20	50
Generation	1000	1200	2000	2000	3000	4000	300	1000	1200	2000	450	1500	3000	1000	1000	2000
Population	150	300	500	250	500	600	20	150	200	400	450	600	750	150	200	300
SBA																
Multinational	1.72E−05	2.58E−04	4.92E−04	9.12E−01	4.36E+00	1.13E+01	−18.54168	8.12E−03	3.30E−02	5.51E−02	1.75E−02	2.06E−02	2.38E+01	1.79E−01	2.60E+00	6.13E+02
Republic	2.80E−15	8.46E−13	1.10E−04	1.89E−02	4.93E−02	8.50E−02	−18.55138	1.47E−08	1.44E−06	2.43E−06	8.01E−03	6.16E−04	1.10E−04	1.27E−04	4.74E−04	4.10E−03
Autocracy	5.39E−13	3.29E−12	7.58E−10	8.81E−03	4.27E−02	8.36E−02	−18.53871	3.57E−08	1.05E−04	4.77E−07	7.89E−03	6.16E−04	4.85E−04	1.28E−04	3.04E−03	1.42E−02
Monarchy	1.36E−39	**2.31E−32**	**8.88E−23**	4.23E−01	3.41E+00	8.67E−01	−18.55472	**0**	**0**	**0**	**0**	**5.55E−18**	**3.33E−16**	1.27E−04	2.55E−04	6.36E−04
SBA without EA																
Multinational	3.70E−28	2.37E−13	3.16E+00	1.12E+00	4.70E+00	1.42E+01	−18.37897	0	5.27E−06	6.56E−05	0.018453	1.67E−02	4.51E−02	1.78E+01	7.35E+01	2.88E+03
Republic	1.66E−11	1.95E−10	3.95E−09	**6.66E−03**	**4.52E−03**	8.79E−02	−17.99086	1.04E−08	4.76E−05	7.27E−05	6.48E−03	1.72E−03	1.56E−04	1.65E−04	6.54E+00	3.22E−02
Autocracy	2.57E−17	6.52E−16	8.34E−10	8.80E−03	1.56E−02	**4.58E−02**	−17.92798	4.81E−13	2.05E−05	1.86E−06	2.96E−03	3.70E−04	1.60E−06	1.27E−04	6.08E+00	1.58E−03
Monarchy	**2.65E−48**	2.69E−27	1.34E−14	2.63E−01	4.80E+00	1.11E+01	**−18.55472**	0	0	2.69E−08	2.62E−02	6.92E−03	1.46E−02	**1.27E−04**	**2.55E−04**	**6.36E−04**

F	*F*27			*F*28	*F*29	*F*30			*F*31			*F*32		
Dimension	10	20	50	2	1	10	20	50	10	20	50	10	20	50
Generation	1000	1200	2500	1000	500	1000	1200	1500	1000	1200	2000	1000	1200	2000
Population	150	200	450	150	100	150	200	200	150	200	500	150	200	200
SBA														
Multinational	1.88E+04	1.06E+14	1.24E+43	4.86E−03	6.00E−53	4.14E−01	3.73E+00	1.30E+02	1.99E−04	3.64E−03	8.48E−03	1.42E−01	4.88E−01	2.86E−01
Republic	1.43E−04	1.39E+04	1.02E+32	1.47E−03	4.60E−31	8.98E−10	1.44E−07	3.39E+00	1.29E−11	6.58E−09	1.57E−09	1.87E−05	3.94E−04	3.70E−03
Autocracy	5.44E−04	2.46E+06	1.30E+31	**4.86E−04**	3.33E−42	1.64E−08	3.44E−07	3.02E+00	4.73E−11	1.37E−07	9.30E−10	4.05E−05	1.42E−03	2.38E−02
Monarchy	5.80E−37	6.53E−15	**1.11E−01**	9.72E−04	7.95E−76	2.79E−39	**6.29E−28**	**9.06E−12**	5.76E−39	4.12E−27	**8.77E−23**	3.10E−14	2.85E−12	**6.51E−11**
SBA without EA														
Multinational	4.48E−12	1.82E+05	9.86E+30	7.29E−03	6.70E−58	3.87E−15	8.24E−02	7.49E+04	2.18E−12	5.32E−11	5.03E+01	3.16E−05	3.94E+00	1.18E+01
Republic	9.12E−04	7.45E+06	4.06E+27	9.72E−04	2.76E−24	1.04E−07	7.44E−06	2.77E−01	2.03E−05	2.97E−08	4.89E−08	5.41E−03	7.61E−04	3.11E−03
Autocracy	2.13E+02	3.43E+02	1.21E+21	9.72E−04	9.42E−65	1.41E−13	1.18E−08	1.25E−03	2.15E−15	3.24E−10	3.88E−09	8.92E−07	3.88E−04	4.01E−02
Monarchy	**7.54E−53**	**1.03E−22**	4.82E+12	0	**1.64E−81**	**1.26E−52**	3.56E−21	6.27E−08	**1.52E−61**	**1.75E−33**	1.66E−12	**3.82E−15**	**7.02E−15**	6.90E−07

Table 10
Comparing with some well-known heuristic search methods.

F	*F*21			*F*22			*F*24		*F*25	*F*26	*F*30	*F*32		*F*33
Dem.	3 [35]	30 [37]	30 [36]	2 [35]	30 [37]	30 [36]	30 [37]	30 [36]	30 [36]	30 [36]	30 [36]	30 [37]	30 [36]	5[35]
GA	8.03E−15	1.88E+01	–	1.09E+00	1.71E+04	–	1.93E+01	–	–	–	–	2.67E+00	–	−29.4
GA–BF	1.43E−15	–	–	1.00E+00	–	–	–	–	–	–	–	–	–	−29.95
ICA	–	2.88E+01	–	–	3.64E+04	–	9.94E−02	–	–	–	–	1.97E+00	–	–
PSO	–	9.04E+00	5.00E−02	–	3.26E+01	3.70E+04	4.64E+00	7.28E+01	5.50E−02	−9.80E+03	2.90E+03	1.40E−01	0.02	–
CICA	–	2.50E−07	–	–	**2.07E+01**	–	**6.83E−04**	–	–	–	–	6.79E−06	–	–
RGA	–	–	2.35E+01	–	–	1.10E+03	–	5.92E+00	1.16E+00	−1.20E+04	5.60E+03	–	2.15E+00	–
GSA	–	–	2.10E−10	–	–	2.52E+01	–	1.53E+01	2.90E−01	−1.10E+03	1.60E+02	–	1.10E−05	–
SBA	**4.90E−37**	**5.87E−15**	**4.30E−19**	**5.55E−06**	3.48E+01	**2.37E+01**	9.95E−02	**3.55E+00**	**2.61E−02**	**−1.26E+04**	**2.65E−02**	**9.22E−08**	**1.32E−06**	**−30**

Table 11
Comparing different monarchy and emperor counts (count of emperor in range [3,7] percent of count of monarchy performs better. Count of monarchy in range [2,7] percent of population size performs better).

Count	*F*21	*F*22	*F*23	*F*24	*F*25	*F*26	*F*27	*F*28	*F*29	*F*30	*F*31	*F*32
Emperor												
3	**6.07E−13**	9.15E−02	−1.86E+01	**2.77E−08**	5.38E−02	**6.36E−05**	**1.34E−11**	**1.99E−03**	2.91E−21	1.51E−10	**5.36E−14**	6.96E−06
	(1.09E−12)	(1.37E−01)	(1.06E−10)	**(3.62E−08)**	(4.72E−02)	**(1.21E−08)**	**(2.20E−11)**	**(4.32E−03)**	(6.45E−21)	(2.89E−10)	**(8.35E−14)**	(8.74E−06)
6	1.07E−11	6.10E−01	−1.86E+01	2.04E−07	**2.55E−02**	6.36E−05	5.25E−09	2.11E−03	**3.16E−22**	**1.11E−10**	3.38E−12	2.11E−05
	(2.26E−11)	(4.79E−01)	(4.37E−12)	(4.52E−07)	**(1.80E−02)**	(2.56E−08)	(1.16E−08)	(4.27E−03)	**(2.86E−22)**	**(1.13E−10)**	(3.80E−12)	(2.65E−05)
9	2.96E−11	**1.71E−01**	−1.86E+01	5.66E−06	5.61E−02	6.37E−05	3.39E−09	4.91E−03	5.93E−17	4.84E−08	6.43E−13	1.90E−05
	(5.32E−11)	**(2.45E−01)**	(3.56E−10)	(4.69E−06)	(2.06E−02)	(4.92E−08)	(7.29E−09)	(4.86E−03)	(1.33E−16)	(8.91E−08)	(1.21E−12)	(1.37E−05)
15	2.38E−11	4.91E−02	−1.86E+01	8.10E−06	3.33E−02	6.38E−05	1.25E−08	3.89E−03	3.61E−19	2.11E−08	5.30E−11	1.67E−04
	(5.30E−11)	(5.98E−02)	(0)	(1.03E−05)	(1.54E−02)	(4.04E−07)	(1.95E−08)	(5.32E−03)	(6.04E−19)	(3.96E−08)	(1.08E−10)	(2.19E−04)
30	6.73E−11	3.33E−02	−1.86E+01	4.79E−04	3.73E−02	6.41E−05	1.00E−06	5.61E−03	5.67E−19	1.23E−07	1.28E−10	7.40E−05
	(8.01E−11)	(5.22E−02)	(1.03E−07)	(6.15E−04)	(1.39E−02)	(7.59E−07)	(2.10E−06)	(4.14E−03)	(9.00E−19)	(2.01E−07)	(2.60E−10)	(4.07E−05)
Monarchy												
9	**7.17E−19**	2.03E−01	−1.86E+01	**5.92E−13**	1.83E−02	**6.36E−05**	8.39E−14	4.10E−03	4.67E−30	**2.98E−14**	4.40E−15	3.02E−08
	(9.82E−19)	(1.26E−01)	(0)	**(7.36E−13)**	(1.02E−02)	**(3.22E−13)**	(1.11E−13)	(5.15E−03)	(9.54E−30)	**(2.16E−14)**	(9.81E−15)	(3.47E−08)
15	7.59E−17	**6.70E−02**	−1.86E+01	1.06E−11	1.98E−02	**6.36E−05**	1.35E−12	3.83E−03	6.89E−28	5.70E−12	**4.73E−16**	9.60E−08
	(7.93E−17)	**(4.11E−02)**	(0)	(2.03E−11)	(8.44E−03)	**(3.22E−13)**	(2.13E−12)	(5.24E−03)	(1.43E−27)	(8.88E−12)	**(6.30E−16)**	(1.31E−07)
30	2.51E−15	8.34E−02	−1.86E+01	5.37E−10	**1.58E−02**	6.36E−05	**5.95E−14**	3.81E−03	**1.36E−30**	1.54E−11	2.27E−14	**1.87E−08**
	(3.22E−15)	(1.24E−01)	(0)	(7.76E−10)	**(5.39E−03)**	(4,55E−13)	**(9.55E−14)**	(4.22E−03)	**(3.04E−30)**	(2.89E−11)	(4.86E−14)	**(1.67E−08)**
45	2.45E−14	2.79E−01	−1.86E+01	1.46E−08	3.31E−02	6.36E−05	5.66E−13	**2.14E−03**	1.22E−24	2.08E−11	1.53E−15	3.99E−08
	(4.61E−14)	(2.08E−01)	(0)	(3.11E−08)	(1.53E−02)	(1.18E−12)	(1.13E−12)	**(3.00E−03)**	(2.73E−24)	(3.23E−11)	(3.43E−15)	(5.84E−08)
90	7.97E−12	4.50E−01	−1.86E+01	4.51E−05	4.64E−02	6.36E−05	1.86E−10	2.40E−03	1.59E−21	1.52E−10	3.76E−13	4.51E−06
	(1.49E−11)	(4.05E−01)	(3.16E−11)	(7.01E−05)	(2.51E−02)	(2.66E−07)	(3.56E−10)	(4.17E−03)	(2.56E−21)	(2.49E−10)	(4.94E−13)	(6.97E−06)

Monarchy count is set to 90.

Table 12
Comparing different internal and external revolutionary rates (revolutionary rates less than 0.5 perform better).

Rate	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32
Revolutionary												
Internal												
0.1	7.00E−26	5.55E−01	**−1.86E+01**	0	2.17E−02	**6.36E−05**	**6.42E−28**	7.77E−03	**6.43E−105**	1.83E−19	1.54E−29	6.85E−10
	(1.02E−25)	(7.86E−01)	**(0)**	(0)	(1.19E−02)	**(2.03E−13)**	**(1.36E−27)**	(4.35E−03)	**(1.2E−104)**	(4.1E−19)	(3.44E−29)	(1.51E−09)
0.3	**1.78E−26**	7.30E−01	−1.86E+01	**0**	1.33E−02	6.36E−05	2.40E−26	3.89E−03	7.75E−41	**2.12E−26**	**2.05E−30**	**1.76E−13**
	(3.99E−26)	(4.68E−01)	(0)	**(0)**	(1.05E−02)	(2.49E−13)	(5.32E−26)	(5.32E−03)	(1.69E−40)	**(2.74E−26)**	**(2.93E−30)**	**(1.81E−13)**
0.5	1.34E−20	2.73E−01	−1.86E+01	7.11E−16	**7.39E−03**	6.36E−05	3.46E−17	**1.18E−15**	3.19E−20	1.64E−17	7.43E−22	9.32E−10
	(1.83E−20)	(2.22E−01)	(4.45E−14)	(1.59E−15)	**(6.97E−03)**	(2.03E−13)	(7.5E−17)	**(2.01E−15)**	(5.4E−20)	(2.79E−17)	(1.48E−21)	(1.36E−09)
0.7	2.91E−15	**1.58E−01**	−1.86E+01	1.42E−11	1.03E−02	6.36E−05	1.73E−11	2.11E−03	8.87E−14	1.39E−10	1.01E−14	6.82E−07
	(3.86E−15)	**(2.39E−01)**	(4.87E−12)	(2.29E−11)	(6.13E−03)	(2.16E−11)	(2.49E−11)	(4.27E−03)	(1.44E−13)	(3.01E−10)	(2.06E−14)	(3.96E−07)
0.9	6.63E−10	3.46E−01	−1.86E+01	5.06E−08	1.89E−02	6.36E−05	1.35E−07	1.94E−03	4.01E−10	7.46E−07	8.02E−11	1.27E−04
	(1E−09)	(2.53E−01)	(4.33E−10)	(6.02E−08)	(1.18E−02)	(3.38E−07)	(7.88E−08)	(4.35E−03)	(4.47E−10)	(6.68E−07)	(4.25E−11)	(5.68E−05)
External												
0.1	**1.57E−27**	1.08E+0	**−1.86E+01**	3.64E−11	2.27E−02	**6.36E−05**	2.41E−25	5.83E−03	**1.59E−104**	**2.65E−26**	**9.03E−27**	2.36E−07
	(2.16E−27)	(1.05E+0)	**(0)**	(8.13E−11)	(1.70E−02)	**(2.03E−13)**	(5.39E−25)	(5.32E−03)	**(3.4E−104)**	**(5.94E−26)**	**(2.02E−26)**	(5.1E−07)
0.3	1.37E−20	6.92E−01	−1.86E+01	**0**	2.32E−02	6.36E−05	**1.05E−26**	3.89E−03	1.20E−91	3.57E−25	3.98E−23	1.99E−08
	(3.07E−20)	(6.78E−01)	(0)	**(0)**	(8.28E−03)	(2.03E−13)	**(2.36E−26)**	(5.32E−03)	(2.68E−91)	(7.33E−25)	(8.89E−23)	(4.45E−08)
0.5	2.96E−13	8.46E−01	−1.86E+01	3.55E−16	**1.53E−02**	6.36E−05	3.32E−21	5.83E−03	9.98E−60	2.28E−18	1.48E−19	**2.91E−10**
	(6.61E−13)	(7.46E−01)	(0)	(7.94E−16)	**(6.38E−03)**	(2.03E−13)	(6.97E−21)	(5.32E−03)	(2.23E−59)	(5.1E−18)	(3.3E−19)	**(3.58E−10)**
0.7	6.90E−14	**5.96E−01**	−1.86E+01	1.62E−13	1.87E−02	6.36E−05	1.50E−15	5.83E−03	3.82E−42	7.02E−12	4.13E−18	4.87E−09
	(1.54E−13)	**(5.65E−01)**	(4.59E−11)	(3.59E−13)	(4.81E−03)	(8.45E−10)	(3.36E−15)	(5.32E−03)	(8.54E−42)	(1.57E−11)	(8.53E−18)	(6.93E−09)
0.9	4.83E−11	1.11E+00	−1.86E+01	1.27E−07	1.98E−02	6.36E−05	7.58E−10	**2.04E−03**	9.95E−23	1.12E−07	9.81E−11	2.09E−05
	(1.03E−10)	(6.05E−01)	(2.85E−11)	(2.65E−07)	(1.68E−02)	(9.9E−09)	(1.32E−09)	**(4.30E−03)**	(1.37E−22)	(2.5E−07)	(2.08E−10)	(3.22E−05)

4.4. One-sample t-test

The one-sample t-test compares the mean score of a sample to a known value; usually the population means (the average for the outcome of some population of interest). The basic idea of the test is a comparison of the average of the sample (observed average) and the population (expected average), with an adjustment for the number of cases in the sample and the standard deviation of the average.

In this case, the null hypothesis is that the difference between the observed mean of SBA algorithm results and the expected mean of optimum value is zero and alternative hypothesis is the difference between the observed mean of SBA algorithm and the expected mean of optimum is not zero. The p-value of 0.05 or less indicates that the condition of reject the null hypothesis is fulfilled. The requirement of the one-sample t-test or assumption is:

> A sample measured on a continuous scale from a population with normal distribution.

Histogram, Q–Q plot, Kolmogorov–Smirnov, D'Agostino–Pearson, Shapiro–Wilk and so on can be used to check for normal distribution.

Different example of graphical representations of histogram and Q–Q graphics are shown in Figs. 12 and 13. A histogram represents a statistical variable using bars, so that the area of each bar is proportional to the frequency of the represented values. A Q–Q graphic represents a confrontation between the qualities from data observed and those from normal distributions. The property of abnormality is clearly presented in Fig. 12. On the contrary, Fig. 13 is the illustration of a sample whose distribution follows a normal shape.

Kolmogorov–Smirnov normality test results of SBA algorithm are shown in Table 14. Table 14 shows the result where the symbol "*" indicates that the normality is not satisfied and the p-value. The p-value represents the dissimilarity of sample of results with respect to the normal shape. Hence, a low p-value points out a non-normal distribution. In this paper, a level of significance α is equal to 0.05. So p-value greater than α indicates that the condition of normality is fulfilled.

In most common benchmark functions, the normality condition is not satisfied, therefore Wilcoxon signed-rank test used in Section 4.5.

Outputs of a one-sample t-test for normal functions are shown in Table 15. In all outputs, there is not a significant difference between the two groups (the significance is less than 0.05). Therefore, sample mean is not significantly greater than the optimum mean of zero.

4.5. Wilcoxon signed-rank test

Frank Wilcoxon proposed both Wilcoxon signed-rank and the rank-sum test for two independent samples [38]. The test was popularized by Siegel [39] in his influential text book on non-parametric statistics. The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used when comparing two related samples, matched samples, or repeated measurements on a single sample to assess whether their population mean ranks differ (i.e. it is a paired difference test).

Wilcoxon signed-rank test is a most useful test to see whether the members of a pair differ in size. In fact, for large numbers it is almost as sensitive as the Student t-test. For small numbers with unknown distributions this test is even more sensitive than the Student t-test. This test is to be preferred over the Student t-test on rare occasions that we do know that values are Normal distributed. It can be used as an alternative to the paired Student's t-test when

Table 13
Comparing different internal and external assimilation coefficients (external assimilation coefficient in [2,3.4] range and internal assimilation coefficient in [2,2.8] range perform better result).

Rate	*F*21	*F*22	*F*23	*F*24	*F*25	*F*26	*F*27	*F*28	*F*29	*F*30	*F*31	*F*32
Assimilation												
Internal												
2	1.38E−23	1.28E+00	**−1.86E+01**	0	**1.18E−02**	6.36E−05	7.17E−26	9.72E−03	5.51E−106	7.19E−18	5.53E−31	1.46E−11
	(3.09E−23)	(9.22E−01)	**(0)**	(0)	**(9.12E−03)**	(2.31E−10)	(1.6E−25)	(0)	(1.2E−105)	(1.25E−17)	(1.22E−30)	(2.34E−11)
2.2	6.04E−35	1.21E+00	−1.86E+01	0	1.82E−02	6.36E−05	3.01E−18	7.77E−03	**1.13E−116**	2.83E−28	1.62E−35	8.33E−10
	(1.35E−34)	(6.98E−01)	(0)	(0)	(1.83E−02)	(2.03E−13)	(6.74E−18)	(4.35E−03)	**(2.5E−116)**	(6.33E−28)	(3.23E−35)	(1.86E−09)
2.4	5.92E−37	1.03E+00	−1.86E+01	0	1.67E−02	6.36E−05	5.43E−43	3.89E−03	2.93E−113	7.96E−32	8.68E−47	6.32E−12
	(1.32E−36)	(7.80E−01)	(0)	(0)	(1.13E−02)	(2.49E−13)	(1.21E−42)	(5.32E−03)	(6.5E−113)	(1.78E−31)	(1.94E−46)	(1.03E−11)
2.6	2.03E−32	4.78E−01	−1.86E+01	0	1.77E−02	6.36E−05	1.14E−56	5.83E−03	6.42E−107	1.62E−37	**6.20E−80**	8.50E−14
	(4.53E−32)	(5.48E−01)	(0)	(0)	(1.39E−02)	(2.03E−13)	(2.56E−56)	(5.32E−03)	(1.4E−106)	(3.63E−37)	**(1.35E−79)**	(1.73E−13)
2.8	8.30E−60	6.64E−01	−1.86E+01	0	1.72E−02	6.36E−05	1.72E−62	3.89E−03	4.58E−108	5.58E−62	7.81E−63	1.32E−14
	(1.86E−59)	(4.23E−01)	(0)	(0)	(6.97E−03)	(2.03E−13)	(3.85E−62)	(5.32E−03)	(1E−107)	(1.25E−61)	(1.75E−62)	(1.32E−14)
3	1.70E−67	5.11E−01	−1.86E+01	0	2.07E−02	6.36E−05	7.06E−53	**1.94E−03**	3.48E−77	2.60E−74	2.05E−63	4.00E−15
	(3.81E−67)	(2.36E−01)	(0)	(0)	(1.33E−02)	(0)	(1.58E−52)	**(4.34E−03)**	(7.79E−77)	(5.19E−74)	(4.59E−63)	(0)
3.2	**2.46E−75**	2.35E−01	−1.86E+01	0	1.72E−02	6.36E−05	2.92E−59	5.83E−03	7.38E−111	4.52E−71	1.39E−75	4.00E−15
	(4.11E−75)	(2.77E−01)	(0)	(0)	(1.07E−02)	(2.03E−13)	(6.52E−59)	(5.32E−03)	(1E−110)	(1.01E−70)	(3.11E−75)	(0)
3.4	3.84E−70	1.18E−01	−1.86E+01	0	1.67E−02	6.36E−05	**1.37E−92**	5.30E−03	6.54E−116	**4.72E−78**	1.49E−65	4.00E−15
	(8.59E−70)	(7.06E−02)	(0)	(0)	(1.06E−02)	(2.03E−13)	**(3.07E−92)**	(4.96E−03)	(1.5E−115)	**(1.06E−77)**	(3.34E−65)	(0)
3.6	1.71E−71	1.24E−01	−1.86E+01	0	1.53E−02	6.36E−05	5.64E−61	3.89E−03	1.49E−76	2.22E−53	2.75E−70	3.29E−15
	(3.83E−71)	(6.40E−02)	(0)	(0)	(1.39E−02)	(2.03E−13)	(1.26E−60)	(5.32E−03)	(3.34E−76)	(4.95E−53)	(6.15E−70)	(1.59E−15)
3.8	5.83E−72	1.77E−01	−1.86E+01	0	1.97E−02	6.36E−05	1.97E−71	3.91E−03	2.52E−97	2.95E−63	3.09E−68	3.29E−15
	(1.22E−71)	(1.54E−01)	(0)	(0)	(7.59E−03)	(0)	(4.41E−71)	(5.30E−03)	(5.64E−97)	(6.59E−63)	(6.92E−68)	(1.59E−15)
4	1.11E−61	**1.10E−01**	−1.86E+01	0	2.61E−02	**6.36E−05**	2.54E−68	5.83E−03	7.61E−98	1.07E−61	2.23E−71	**3.29E−15**
	(2.43E−61)	**(6.04E−02)**	(0)	(0)	(9.60E−03)	**(0)**	(5.67E−68)	(5.32E−03)	(1.7E−97)	(2.4E−61)	(4.98E−71)	**(1.59E−15)**
External												
2	5.58E−69	6.07E−01	−1.86E+01	0	2.66E−02	6.36E−05	**3.40E−72**	7.77E−03	5.34E−73	**2.98E−69**	4.74E−68	4.71E−15
	(1.25E−68)	(4.80E−01)	(0)	(0)	(6.84E−03)	(2.03E−13	**(7.6E−72**	(4.35E−03)	(1.19E−72)	**(4.83E−69)**	(9.96E−68)	(1.59E−15)
2.2	**3.44E−80**	3.03E−01	−1.86E+01	0	1.72E−02	6.36E−05	1.40E−61	6.76E−03	1.57E−78	2.70E−65	1.40E−64	4.00E−15
	(7.69E−80)	(2.96E−01)	(0)	(0)	(8.17E−03)	(0)	(3.14E−61	(4.37E−03)	(3.52E−78)	(6.03E−65)	(3.12E−64)	(0)
2.4	1.49E−76	**1.08E−01**	−1.86E+01	**0**	2.51E−02	6.36E−05	8.43E−68	3.89E−03	**5.59E−92**	1.49E−64	**1.80E−68**	3.29E−15
	(3.33E−76)	**(7.62E−02)**	(0)	**(0)**	(3.65E−03)	(0)	(1.88E−67	(5.32E−03)	**(1.25E−91)**	(3.33E−64)	**(4.02E−68)**	(1.59E−15)
2.6	3.64E−56	1.17E−01	−1.86E+01	3.55E−16	**1.18E−02**	6.36E−05	6.75E−61	5.83E−03	2.27E−84	3.47E−67	2.47E−53	3.29E−15
	(8.13E−56)	(3.61E−02)	(0)	(7.94E−16)	**(2.69E−03)**	(2.03E−13)	(1.51E−60)	(5.32E−)	(5.08E−84)	(7.76E−67)	(5.53E−53)	(1.59E−15)
2.8	3.32E−60	4.84E−02	**−1.86E+01**	0	2.32E−02	**6.36E−05**	4.47E−51	3.89E−03	1.09E−73	8.86E−60	1.02E−63	**1.87E−15**
	(7.11E−60)	(5.69E−02)	**(0)**	(0)	(6.43E−03)	**(0)**	(1.00E−50)	(5.32E−03)	(2.44E−73)	(1.98E−59)	(2.27E−63)	**(1.95E−15)**
3	7.62E−48	1.18E−01	−1.86E+01	0	1.24E−02	6.36E−05	6.23E−37	5.83E−03	1.14E−88	8.36E−57	3.30E−62	1.87E−15
	(1.7E−47)	(9.86E−02)	(0)	(0)	(9.70E−03)	(0)	(1.39E−36)	(5.32E−03)	(2.54E−88)	(1.63E−56)	(7.36E−62)	(1.95E−15)
3.2	6.85E−48	2.01E−01	−1.86E+01	0	2.32E−02	6.36E−05	3.63E−40	3.89E−03	6.92E−79	4.30E−38	3.35E−45	4.00E−15
	(1.53E−47)	(2.24E−01)	(0)	(0)	(7.10E−03)	(0)	(8.13E−40)	(5.32E−03)	(1.55E−78)	(9.61E−38)	(7.5E−45)	(0)
3.4	1.16E−45	2.36E−01	−1.86E+01	0	1.87E−02	6.36E−05	1.50E−32	4.77E−03	1.09E−84	4.42E−40	1.30E−36	3.29E−15
	(2.5E−45)	(1.40E−01)	(0)	(0)	(5.67E−03)	(0)	(3.36E−32)	(4.86E−03)	(2.42E−84)	(9.37E−40)	(2.44E−36)	(1.59E−15)
3.6	2.21E−35	2.10E−01	−1.86E+01	0	3.61E−02	6.36E−05	2.82E−35	4.55E−03	1.69E−57	7.27E−30	4.29E−36	2.58E−15
	(4.85E−35)	(1.50E−01)	(0)	(0)	(2.891E−02)	(0)	(6.3E−35)	(4.91E−03)	(3.78E−57)	(1.63E−29)	(8.58E−36)	(1.95E−15)
3.8	5.29E−38	2.31E−01	−1.86E+01	0	3.79E−02	6.36E−05	1.89E−27	**1.94E−03**	1.01E−73	2.97E−30	2.16E−35	4.00E−15
	(1.13E−37)	(1.96E−01)	(0)	(0)	(3.24E−02)	(0)	(4.21E−27)	**(4.35E−03)**	(2.25E−73)	(6.08E−30)	(4.82E−35)	(0)
4	9.94E−27	3.22E−01	−1.86E+01	4.62E−15	2.86E−02	6.36E−05	3.10E−22	5.83E−03	2.54E−81	6.68E−26	2.92E−29	3.29E−15
	(2.22E−26)	(3.33E−01)	(0)	(1.03E−14)	(1.55E−02)	(2.49E−13)	(6.81E−22)	(5.32E−03)	(5.68E−81)	(1.44E−25)	(6.46E−29)	(1.59E−15)

Table 14
Test of normality of Kolmogorov–Smirnov.

	F1	F2	F3	F5	F8	F9	F11	F12	F13	F15	F16	F17
Large scale benchmark function (FES = 1.2E+05)												
SBA	0.022	>0.1	>0.1	>0.1	<0.0001*	>0.1	>0.1	>0.1	>0.1	0.07	>0.1	>0.1

	F21	F22	F23	F25	F27	F29	F30	F31	F32
Common benchmark function									
SBA	<0.0001*	>0.1	<0.0001*	>0.1	<0.0001*	<0.0001*	<0.0001*	<0.0001*	0.096

Table 15
One-sample *t*-test results.

One-sample statistics

	N	Mean	Std. deviation	Std. error mean
F22	20	4.23E−01	2.94E−01	6.57E−02
F25	20	4.27E−02	1.80E−02	4.03E−03
F32	20	3.10E−14	2.75E−14	6.14E−15
F1	25	9.76E+08	1.78E+08	3.56E+08
F2	20	6.29E+03	1.84E+02	4.12E+01
F3	20	2.04E+01	4.12E−02	9.22E−03
F5	24	3.58E+08	6.22E+07	1.27E+07
F9	24	3.54E+09	3.05E+08	6.22E+07
F11	25	2.26E+02	8.18E−01	1.64E−01
F12	24	2.02E+06	1.78E+05	3.62E+04
F13	24	1.89E+07	7.05E+06	1.44E+06
F15	25	1.55E+04	3.11E+02	6.23E+01
F17	25	3.99E+06	2.27E+05	4.54E+04
F18	25	4.02E+10	4.40E+09	8.80E+08

One-sample test

	Test value = 0				
	t	df	2-Tailed p	95% confidence interval of the difference Lower	Upper
F22	6.44	19	<0.0001	2.86E−01	5.61E−01
F25	10.60	19	<0.0001	3.42E−02	5.11E−02
F32	5.05	19	<0.0001	1.82E−14	4.39E−14
F1	27.47	24	<0.0001	9.05E+08	–
F2	152.78	19	<0.0001	6.20E+03	–
F3	2215.99	19	<0.0001	20.42	–
F5	28.24	23	<0.0001	3.32E+08	3.85E+08
F9	56.87	23	<0.0001	3.41E+09	–
F11	1377.64	24	<0.0001	2.25E+02	–
F12	55.63	23	<0.0001	1.94E+06	
F13	13.09	23	<0.0001	1.59E+07	–
F15	248.15	24	<0.0001	1.53E+04	–
F17	87.97	24	<0.0001	3.90E+06	–
F18	45.69	24	<0.0001	3.38E+10	–

Table 16
Wilcoxon signed-rank test.

F	N	W_+	W_-	H_0
F1	12	50	28	Accepted
F2	10	40	15	Accepted
F3	10	31	24	Accepted
F5	12	53	25	Accepted
F8	12	41	37	Accepted
F9	12	33	45	Accepted
F11	12	46	32	Accepted
F12	12	27	51	Accepted
F13	12	48	30	Accepted
F15	12	43	35	Accepted
F17	12	48	30	Accepted
F18	12	47	31	Accepted
F21	10	27	28	Accepted
F22	10	22	33	Accepted
F25	10	30	25	Accepted
F27	10	18	37	Accepted
F29	10	21	34	Accepted
F30	10	28	27	Accepted
F31	10	38	17	Accepted
F32	10	22	33	Accepted

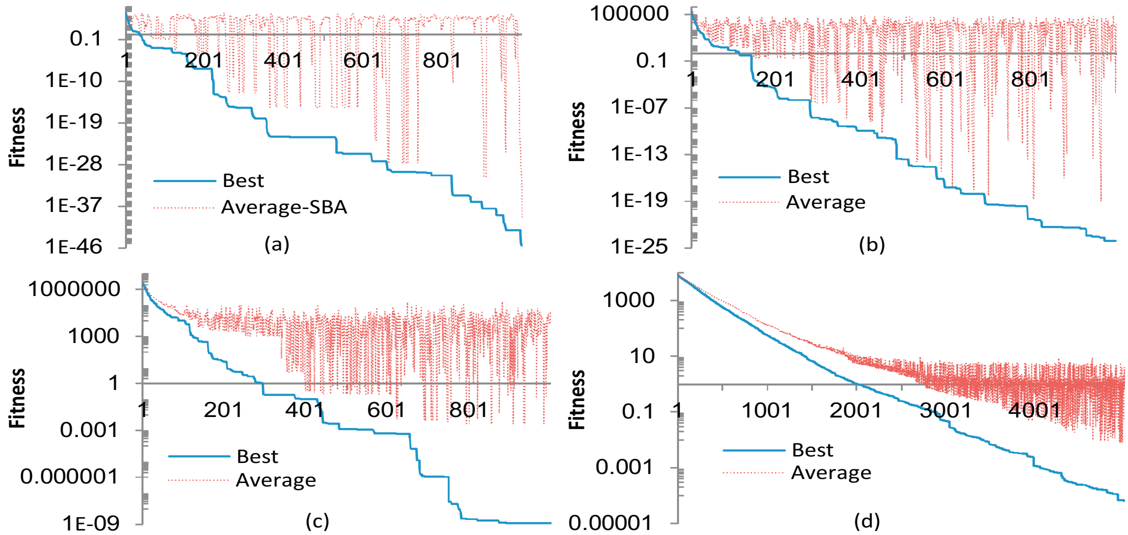

Fig. 9. *F*30's convergence diagram: (a) $n = 10$, (b) $n = 20$, (c) $n = 50$, and (d) $n = 1000$.

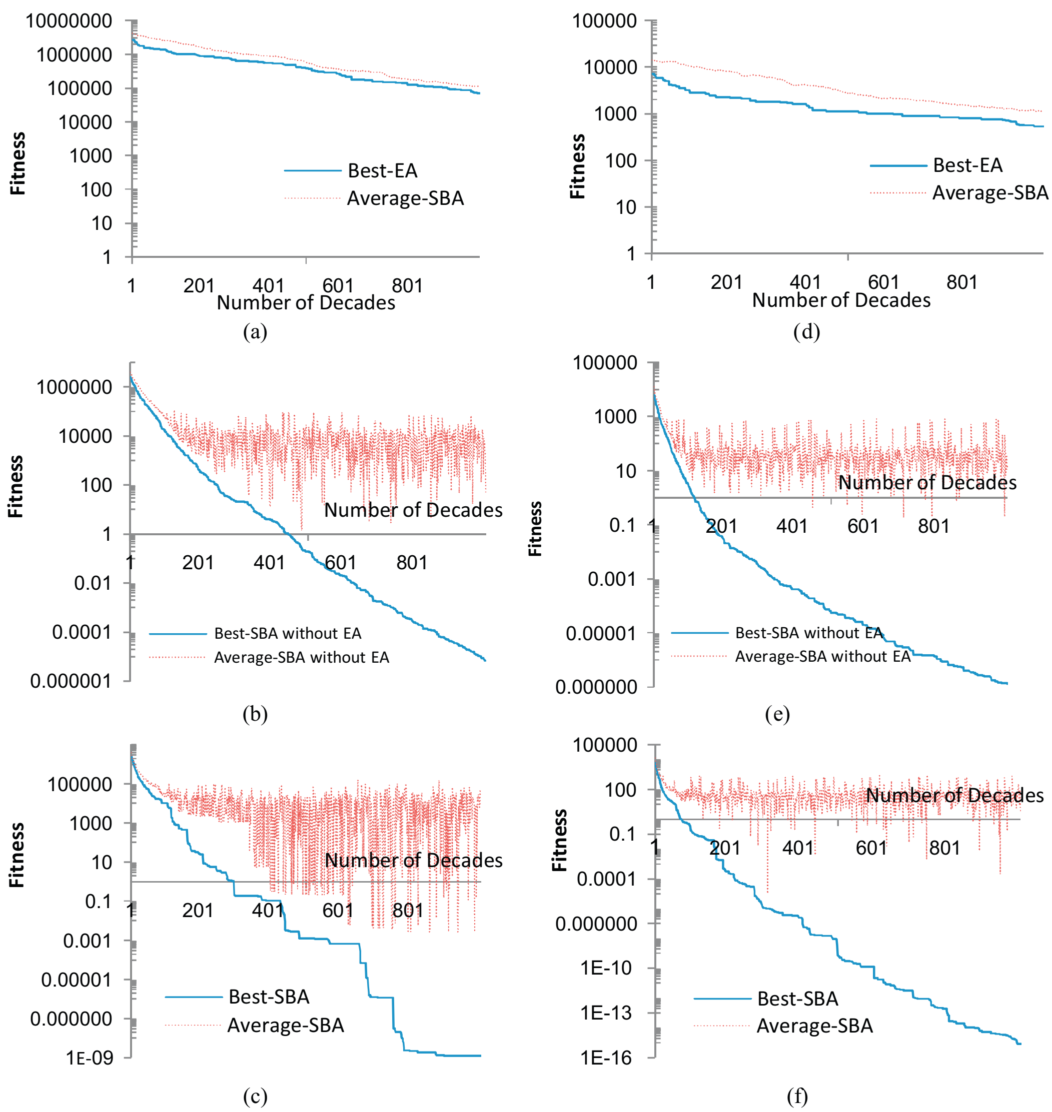

Fig. 10. Fitness profile using algorithms for *F*30 and *F*31 ($n = 50$). *F*30 fitness profile using: (a) EA, (b) SBA without EA, and (c) SBA, *F*31 fitness profile using: (d) EA, (e) SBA without EA, and (f) SBA.

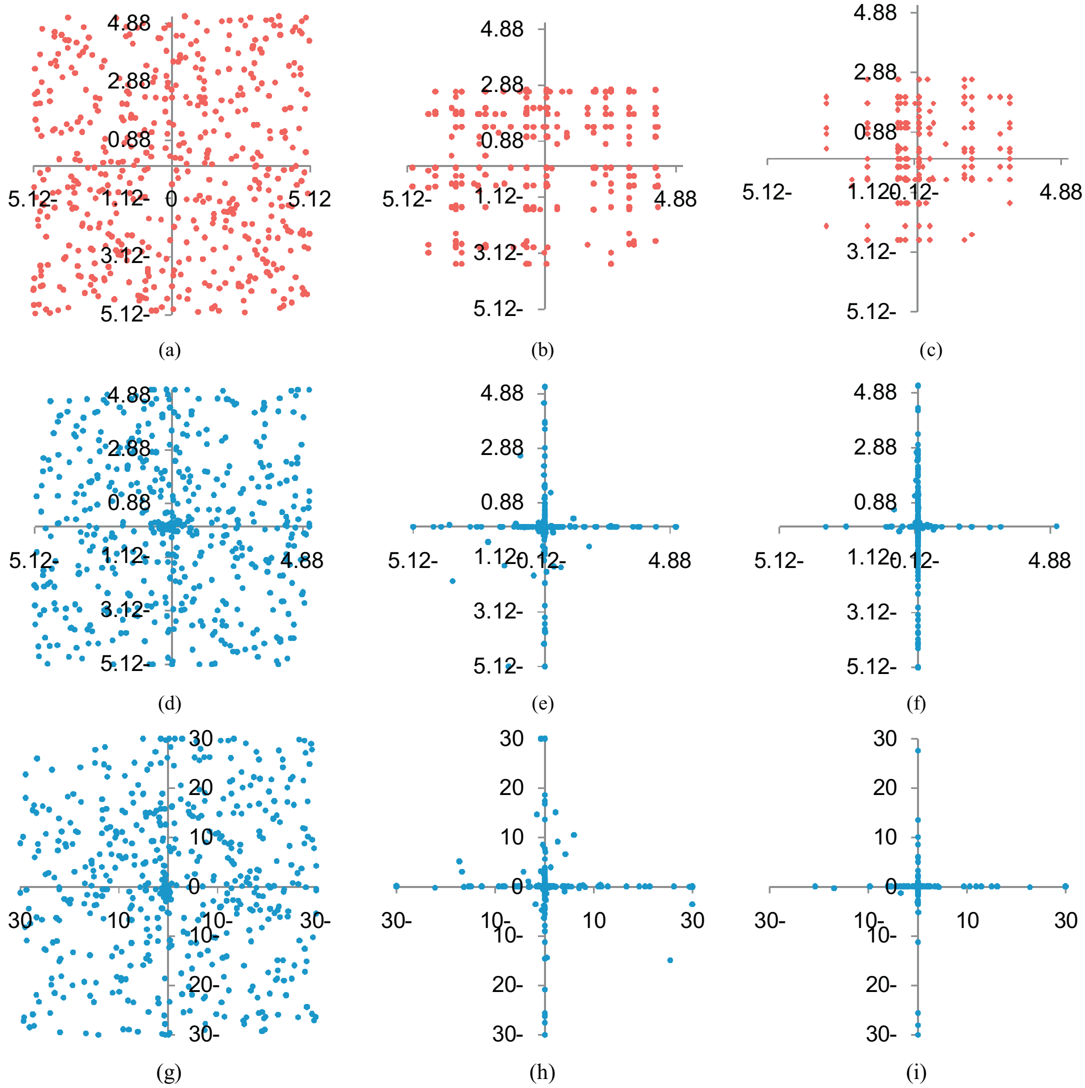

Fig. 11. *F*21 ($n = 2$) generated point of EA in generations (a) 10, (b) 50, (c) 100 and SBA in generations (d) 10, (e) 50, (f) 100 and SBA in generations (g) 10, (h) 50, (i) 100.

the population cannot be assumed to be normally distributed or the data is on the ordinal scale.

The null hypothesis tested is the difference ($z = x - y$) between the members of each pair (x, y) has median value zero (H_0: $\theta = 0$). To be complete, x and y have identical distributions and the alternate hypothesis tested $H1$: θ does not equal to 0.

Suppose collection of $2n$ observations, two observations of each of the n subjects. Let i denote the particular subject that is being referred to and the first observation measured on subject i be denoted by x_i and second observation be y_i. For each i in the observations, x_i and y_i should be paired together.

Obtained test statistic results of Wilcoxon signed-rank test for SBA algorithm are shown in Table 16. The test statistic, W_+, is given by the sum of all of the positive values in the Signed Rank column. The test statistic, W_-, is given by the sum of all of the negative values in the Signed Rank column. Lastly, this test statistic is analyzed using a table of critical values. If the test statistic is less than or equal to the critical value based on the number of observations n, then the null hypothesis is rejected for the alternative hypothesis. Otherwise, the null hypothesis is not rejected.

4.6. Discussion

A high percentage of meta-heuristics hybridizing population-based meta-heuristics has been proposed for various optimization problems. This paper presented a new hybrid approach. As mentioned, related work on hybridization EA and ICA are sequential (or cascaded). Furthermore, it should be pointed out and emphasized that the new approach proposes way to efficiently implement algorithms based on human community. A new hybrid approach by combining EA and ICA is called Social-Based Algorithm (SBA). SBA is a numerical stochastic search algorithm mimicking natural behavior of human colonizing in opportunity spaces for function optimization. It benefits from combination of EA and ICA. The idea was based on a socio-politically and personal promotion.

Convergence of SBA is studied for finding global minima of different benchmark functions; “Sphere”, “Griewank”, “Rastrigin”, “Ackley”, etc. Experimental results show that it can be used for different problems. Table 2 shows result for large scale global optimization and SBA performs well.

The feasibility and efficiency of SBA for optimization of different examples are compared to seven different algorithms. Table 10 depicts the final results, using Genetic algorithm (GA) [35], the

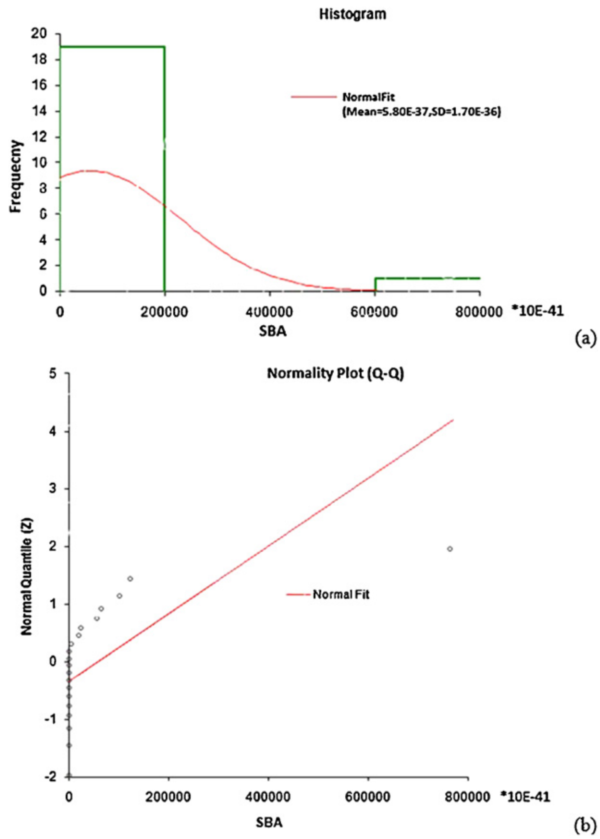

Fig. 12. Example of non-normal distribution: *F*27 and SBA algorithm: histogram and Q–Q graphic.

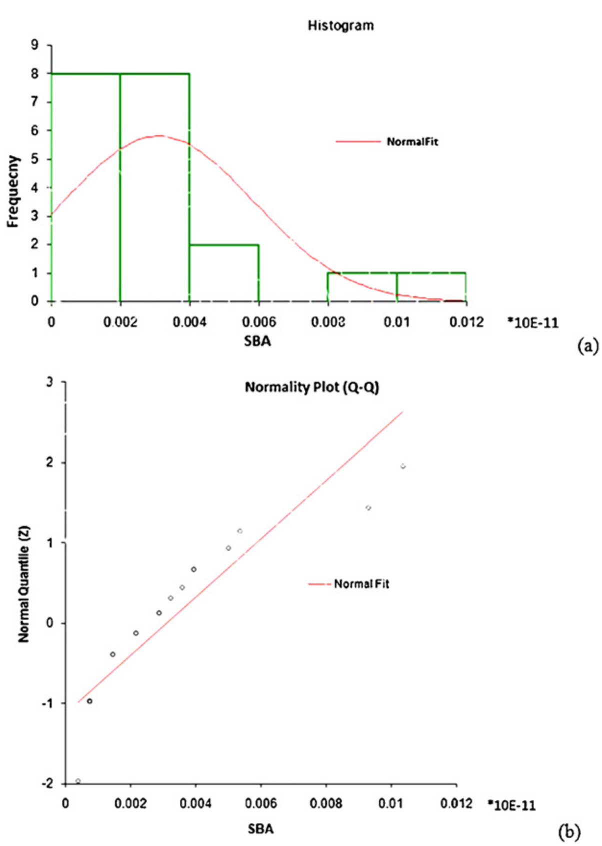

Fig. 13. Example of normal distribution: *F*32 and SBA algorithm: histogram and Q–Q graphic.

Hybrid Genetic Algorithm–Bacterial Foraging (GA–BF) [35], A Gravitational Search Algorithm (GSA) [36], Imperialist Competitive Algorithm (ICA) [37], Particle Swarm Optimization (PSO) [37], Chaotic Imperialist Competitive Algorithm (CICA) [37] and Real Genetic Algorithm (RGA) [36]. SBA provides better than GA, GA–BF, ICA, PSO, RGA and GSA. In most result SBA perform better than CICA. SBA uses different parameters therefore the effect of tuning parameters on performance of the proposed algorithm is studied.

Results show SBA is capable of finding globally optimal solutions in a relatively small number of generations. Numerous numerical simulations are performed to demonstrate effectiveness of SBA algorithm. The proposed algorithm can outperform EA, ICA, ABC and PSO algorithms.

Although SBA can be applied to many optimization problems their performance is still subject of the No Free Lunch (NFL) theorem. According to this theorem any two algorithms are equivalent, when their performance is compared across all possible problems.

“An Almost No Free Lunch (ANFL)” theorem shows that for each function which can be optimized efficiently by a search heuristic there can be constructed many related functions where the same heuristic is bad. As a consequence, search heuristics use some idea how to look for good points and can be successful only for functions “giving the right hints” [40].

Exploitation and exploration are two important issues in the evolution process of the genetic search [41]. Exploration is the creation of population diversity by exploring the search space; exploitation is the reduction of the diversity by focusing on the individuals of higher fitness, or exploiting the fitness information represented within the population. In EA mutation is often seen as an exploration operator because it introduces new material in an unbiased manner [42], revolution policy plays this role in ICA. Similarly, crossover can be seen as explorative operator because it recombines the old material of the parents into new configurations [42], assimilation policy plays this role in ICA.

5. Conclusion and future works

This paper proposes a novel hybrid approach consisting EA and ICA and its performance is evaluated using various test functions. It illustrates the performance of the proposed hybrid approach using 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 through reasonable time limit.

The results show the efficiency and capabilities of the new hybrid algorithm in finding the optimum. Amazingly, its performance is about 85% better than other algorithms such as EA, ICA, ABC and PSO. The performance achieved is quite satisfactory and promising for all test functions.

In this paper, we have tried to shed some light on the effectiveness and efficiency of hybridizing Evolutionary Algorithms with ICA. These approaches show that hybridizing is one possible way to build a competent algorithm that solves hard problems quickly, reliably and accurately. Our future research would include hybridization SBA with PSO algorithm. It focuses on the advantage of PSO into SBA. The authors are working to apply proposed algorithm for solving Artificial Neural Network (ANN) weights optimization on river flood problem.

Acknowledgments

The first author gratefully acknowledges support from Mr. E. Atashpaz Gargari, who has invented ICA algorithm and presented the related useful documents on his website, his support and help is truly appreciated.

References

[1] H. Hajimirsadeghi, C. Lucas, A hybrid IWO/PSO algorithm for fast and global optimization, in: IEEE EUROCON 2009, 2009, pp. 1964–1971.
[2] M.K. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Systems (2002) 52–67.
[3] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm intelligence: from natural to artificial systems, The Journal of Artificial Societies and Social Simulation (JASSS) (2001).
[4] A. Okubo, Dynamical aspects of animal grouping: swarms, schools, flocks, and herds, Advances in Biophysics (1986) 1–94.
[5] S. Gueron, S.A. Levin, The dynamics of group formation, Math Bioscience 128 (1995) 243–264.
[6] E. Ben-Jacob, Bacterial self-organization: co-enhancement of complexification and adaptability in a dynamic environment, Philosophical Transactions of the Royal Society Series A: Mathematical, Physical and Engineering Sciences 361 (2003) 1283–1312.
[7] E. Ben-Jacob, H. Levine, Self-engineering capabilities of bacteria, Journal of The Royal Society Interface (2006) 197–214.
[8] E. Ben-Jacob, I. Becker, Y. Shapira, Bacterial linguistic communication, Trends in Microbiology 12 (2004) 366–372.
[9] A. Trewavas, Aspects of plant intelligence, Annals of Botany 92 (2003) 1–20.
[10] K.M. Passino, T.D. Seeley, Modeling and analysis of nest-site selection by honey bee swarms: the speed and accuracy trade-off, Behavioral Ecology and Sociobiology 59 (2006) 427–442.
[11] N.F. Britton, Deciding on a new home: how do honey bees agree? Proceedings of the Royal Society of London Series B 269 (2002) 1383–1388.
[12] J.H. Holland, Adoption in Natural and Artificial Systems, University of Michigan, 1975.
[13] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[14] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS '95), 1995, pp. 39–43.
[15] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man and Cybernetics (1996) 1–13.
[16] C. Grosan, A. Abraham, Hybrid Evolutionary Algorithms: Methodologies, Architectures and Reviews: Studies in Computational Intelligence, Springer-Verlag, 2007, pp. 1–17.
[17] K.Y. Chan, C.K. Kwong, H. Jiang, M.E. Aydin, T.C. Fogarty, A new orthogonal array based crossover, with analysis of gene interactions, for evolutionary algorithms and its application to car door design, Expert Systems with Applications 37 (5) (2010) 3853–3862.
[18] K.Y. Chan, M.E. Aydin, T.C. Fogarty, Main effect fine-tuning of the mutation operator and the neighborhood function for uncapacitated facility location problems, Soft Computing: A Fusion of Foundations, Methodologies and Applications 10 (11) (2006).
[19] K.Y. Chan, T.S. Dillon, C.K. Kwong, Modeling of a liquid epoxy molding process using a particle swarm optimization-based fuzzy regression approach, IEEE Transactions on Industrial Informatics (2011) 148–158.
[20] T. Niknam, A new hybrid imperialist competitive algorithm on data clustering, Indian Academy of Sciences 36 (2010) 293–315.
[21] X. Shi, Y. Liang, H. Lee, C. Lu, L. Wang, An improved GA and a novel PSO-GA-based hybrid algorithm, Information Processing Letters 93 (2005) 255–261.
[22] L.Y. Tseng, S.C. Liang, A hybrid metaheuristic for the quadratic assignment, Computational Optimization and Applications (34) (2005) 85–113.
[23] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: IEEE Congress on Evolutionary Computation (CEC), 2007, pp. 4661–4667.
[24] M. Abdechiri, M.R. Meybod, Hybrid hopfield network-imperialist competitive algorithm for solving the SAT problem, in: 3rd International Conference on Signal Acquisition and Processing (ICSAP 2011), 2011.
[25] V. Khorani, F. Razavi, E. Ghoncheh, A new hybrid evolutionary algorithm based on ICA and GA: Recursive-ICA-GA, in: The 2010 International Conference on Artificial Intelligence (ICAI'10), 2010.
[26] T. Jain, M.J. Nigam, Synergy of evolutionary algorithm and socio-political process for global optimization, Expert Systems with Applications 37 (2010) 3706–3713.
[27] F. Razavi, Using Evolutionary Imperialist Competitive Algorithm (ICA) to Coordinate Overcurrent Relays: GEM'11, 2010.
[28] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers and Operations Research 13 (1986) 533–549.
[29] T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, USA, 1996.
[30] C. Pitelis, R. Sugden, The Nature of the Transnational Firm, Routledge, 2000, pp. 27–30.
[31] M.M. Eusuff, K.E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, Journal of Water Resources Planning and Management 129 (2003) 210–225.
[32] K. Tang, Benchmark Functions for the CEC'2010 Special Session and Competition on Large-scale Global Optimization, Nature Inspired Computation and Applications Laboratory, Technical Report, http://nical.ustc.edu.cn/cec10ss.php, 2010.
[33] J. Brest, Large scale global optimization using self-adaptive differential evolution algorithm, in: WCCI 2010 IEEE World Congress on Computational Intelligence, 2010, pp. 3097–3104.
[34] H. Wang, Sequential DE enhanced by neighborhood search for large scale global optimization, in: WCCI 2010 IEEE World Congress on Computational Intelligence, 2010, pp. 4056–4062.
[35] A. Dong Hwa Kim, B. Ajith Abraham, J.H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Sciences (2007) 3918–3937.
[36] E. Rashedi, H. Nezamabadi-pour, S.G.S.A. Saryazdi, A gravitational search algorithm, Information Sciences (2009) 2232–2248.
[37] H. Bahrami, K. Faez, M. Abdechiri, Imperialist competitive algorithm using chaos theory for optimization (CICA), in: 12th International Conference on Computer Modelling and Simulation (UKSim), 2010, pp. 98–103.
[38] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) 80–83.
[39] S. Siegel, Non-parametric Statistics for the Behavioral Sciences, McGraw-Hill, New York, 1956, pp. 75–83.
[40] S. Droste, T. Jansen, I. Wegener, Optimization with randomized search heuristics – the (A)NFL theorem, realistic scenarios, and difficult functions, Theoretical Computer Science 287 (2002) 131–144.
[41] L. Hansheng, K. Lishan, Balance between exploration and exploitation in genetic search, Wuhan University Journal of Natural Sciences 4 (1999) 28–32.
[42] A.E. Eiben, C.A. Schippers, On evolutionary exploration and exploitation, Fundamental Information (1998) 1–6.