Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 1954 – 1969

doi:10.3906/elk-1808-194

Hybrid parliamentary optimization and big bang-big crunch algorithm for global optimization

Soner KIZILOLUK[1,*], **Ahmet Bedri ÖZER**[2]
[1]Department of Computer Engineering, Faculty of Engineering, Munzur University, Tunceli, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Fırat University, Elazığ, Turkey

Received: 28.08.2018 • **Accepted/Published Online:** 25.02.2019 • **Final Version:** 15.05.2019

Abstract: Researchers have developed different metaheuristic algorithms to solve various optimization problems. The efficiency of a metaheuristic algorithm depends on the balance between exploration and exploitation. This paper presents the hybrid parliamentary optimization and big bang-big crunch (HPO-BBBC) algorithm, which is a combination of the parliamentary optimization algorithm (POA) and the big bang-big crunch (BB-BC) optimization algorithm. The intragroup competition phase of the POA is a process that searches for potential points in the search space, thereby providing an exploration mechanism. By contrast, the BB-BC algorithm has an effective exploitation mechanism. In the proposed method, steps of the BB-BC algorithm are added to the intragroup competition phase of the POA in order to improve the exploitation capabilities of the POA. Thus, the proposed method achieves a good balance between exploration and exploitation. The performance of the HPO-BBBC algorithm was tested using well-known mathematical test functions and compared with that of the POA, the BB-BC algorithm, and some other metaheuristics, namely the genetic algorithm, multiverse optimizer, crow search algorithm, dragonfly algorithm, and moth-flame optimization algorithm. The HPO-BBBC algorithm was found to achieve better optimization performance and a higher convergence speed than the above-mentioned algorithms on most benchmark problems.

Key words: Parliamentary optimization algorithm, big bang-big crunch algorithm, global optimization, hybridization

1. Introduction

Optimization refers to the selection of the best solution from among multiple solutions to a problem. Traditional optimization techniques (such as Newton's method, steepest descent, and linear programming) usually fail to solve global optimization problems that have many local optima and nonlinear objective functions. By contrast, metaheuristic algorithms are more efficient in overcoming these challenges. Many metaheuristic algorithms are inspired by biological phenomena as well as by physical, social, and chemical processes [1]. For example, the genetic algorithm (GA) [2] and artificial immune systems (AISs) [3] are based on biology, the gravitational search algorithm (GSA) [4] is based on physics, the imperialist competitive algorithm (ICA) [5] is based on social concepts, and the artificial chemical reaction optimization algorithm (ACROA) [6] is based on chemistry. Although various metaheuristic algorithms can successfully solve some specific problems, they do not show similar performances in solving all problems. Therefore, new algorithms have been proposed to improve the existing algorithms. Hybridization, which aims to combine the properties of two or more algorithms into a single hybrid algorithm, is one such technique. The unique benefit of hybridization is that the new algorithm provides

*Correspondence: sonerkiziloluk@munzur.edu.tr

better performance compared to its individual components [7]. Recently, many hybrid versions of well-known optimization methods have been developed by researchers, such as hybrid GA-particle swarm optimization (PSO)-symbiotic organisms search (SOS) by Farnad et al. [8], hybrid genetic deflated Newton (HGDN) method by Noack and Funke [9], hybrid firefly algorithm (FA)-PSO by Aydilek [10], hybrid biogeography-based optimization (BBO)-gray wolf optimizer (GWO) by Zhang et al. [11], hybrid hierarchical backtracking search optimization (HHBSA) based on backtracking search optimization (BSA), differential evolution (DE), and teaching-learning-based optimization (TLBO) by Zou et al. [12], hybrid harmony search (HS)-simulated annealing (SA) by Assad and Deep [13], hybrid artificial bee colony (ABC)-DE by Jadon et al. [14], memory-based hybrid dragonfly algorithm (MHDA) by Ranjini and Murugan [15], and hybrid flower pollination algorithm (FPA)-clonal selection algorithm (CSA) by Nabil [16].

The parliamentary optimization algorithm (POA) was proposed by Borji [17] for global optimization. It is inspired by the competitive and cooperative behaviors of parliamentary parties. The POA consists of two phases: intragroup competition and intergroup cooperation. In the first phase, the regular members are biased toward the candidate members in the ratio of their fitness values, which allows the algorithm to search for potential points in the search space. There are two different scenarios in the second phase. In the first scenario, the most powerful groups can be merged into a single group in order to increase their power. In the second scenario, the weakest groups can be removed in order to preserve the computation power and decrease function evaluations. Only a few studies have investigated the POA. In these studies, the POA was used for different problems, such as global optimization [18], permutation constraint satisfaction problems [19], overlapping community detection in social networks [20], finding numerical classification rules [1], and classification of Web pages [21]. Furthermore, a hybrid version of the POA, i.e. a combination of the POA and artificial neural networks, was proposed for passenger flow prediction [22].

The big bang-big crunch (BB-BC) algorithm, inspired by one of the evolutionary theories of the universe, was initially proposed by Erol and Eksin [23]. The algorithm consists of two phases. In the big bang phase, the particles are randomly created in a search space. In the big crunch phase, the randomly distributed particles are drawn into an order. Various applications of the BB-BC algorithm have been reported in the literature, such as data clustering [24], optimal placement and sizing of voltage-controlled distributed generators [25], and optimal design of structures [26]. Furthermore, some hybrid variations of the BB-BC have been proposed [27], including hybrid PSO-BB-BC for optimal reactive power dispatch [28]; hybrid BB-BC-PSO for optimal sizing of a stand-alone hybrid power system including a photovoltaic panel, wind turbine, and battery bank [29]; hybrid BB-BC-conjugate gradient (CG) algorithm for operational reliability modeling of hydrogenerator groups [30]; and hybrid BB-BC-PSO for parameter identification of a proton-exchange membrane fuel cell [31].

In this study, the hybrid parliamentary optimization and big bang-big crunch (HPO-BBBC) algorithm, which is a combination of the POA and the BB-BC algorithm, is proposed to solve global numerical optimization problems. The proposed method achieves a balance between exploration and exploitation by using the exploration ability of the POA and the exploitation ability of the BB-BC algorithm. The performance of the HPO-BBBC algorithm is tested using nine standard mathematical test functions and eight composition, rotated, shifted, and expanded functions selected from CEC 2005. It is compared with that of the POA, the BB-BC algorithm, and five other metaheuristics, namely the GA [2], multiverse optimizer (MVO) [32], crow search algorithm (CSA) [33], dragonfly algorithm (DA) [34], and moth-flame optimization algorithm (MFO) [35]. The results show that the HPO-BBBC algorithm can effectively solve most benchmark problems and has a higher convergence speed than the above-mentioned algorithms. The remainder of this paper is organized as follows. The POA and the BB-BC algorithm are described in Sections 2 and 3, respectively. Section 4 provides a detailed explanation of the HPO-BBBC algorithm. The experimental results are discussed and compared in Section 5. Finally, our conclusions are stated in Section 6.

2. Parliamentary optimization algorithm

The POA is inspired by the competitive and cooperative behaviors of parliamentary parties. The flowchart of the POA is shown in Figure 1.

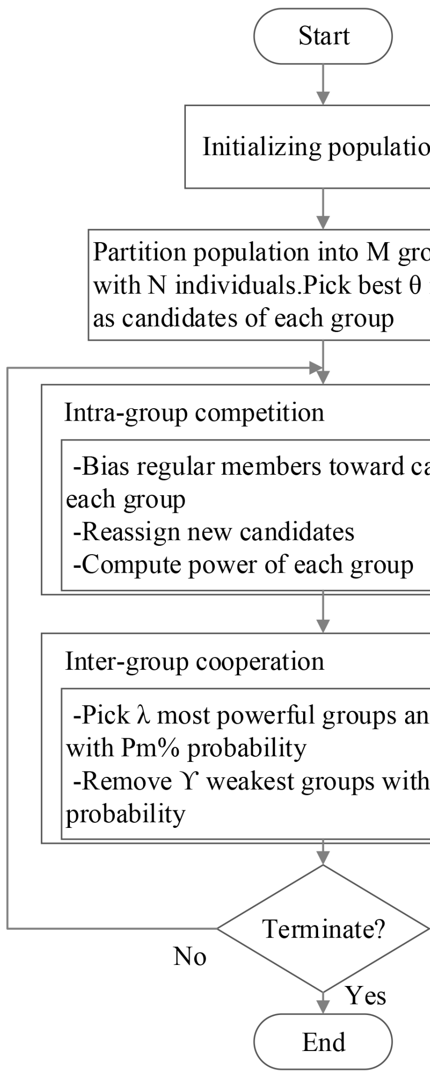

Figure 1. Flowchart of the POA.

The POA begins with an initialization process. The individuals are created with random positions throughout the search space. Then the initialized individuals are evenly partitioned into M groups, where each group contains N individuals. A few individuals with the highest fitness in each group are considered as candidate members. The remaining individuals are referred to as regular members [17, 18].

Next, the intragroup competition phase begins. In this phase, the regular members are biased toward the candidate members in the ratio of their fitness values. The new position of a regular member is calculated as

$$p^{'} = p_0 + \pi \left(\frac{\sum_{i=1}^{\theta} (p_i - p_0).f(p_i)}{\sum_{i=1}^{\theta} f(p_i)} \right), \tag{1}$$

where π is a random value between 0.5 and 2, $p^{'}$ is the new position and p_0 is the current position of the regular member, p_i is the position of a candidate member, and f is the fitness function. The biasing operation is shown in Figure 2. After biasing, the regular members might have higher fitness values than the candidate members. In this case, the candidate members are reassigned. After the reassignment, the power of the groups is calculated as

$$power^i = \frac{m.avg(Q^i) + n.avg(R^i)}{m+n}; m > n, \tag{2}$$

where Qi and Ri are the fitness values of candidate members and regular members of group i, respectively, while m and n represent weight constants.

In intergroup cooperation, a random number is generated, and if it is smaller than Pm, the λ most powerful groups can be merged into one group in order to increase their power. Like merging, a random number is generated, and if it is smaller than Pd, the γ weakest groups can be removed in order to preserve the computation power and decrease function evaluations. When the stopping conditions are satisfied, the algorithm terminates and the best member of the best group is considered as the solution [17, 18].

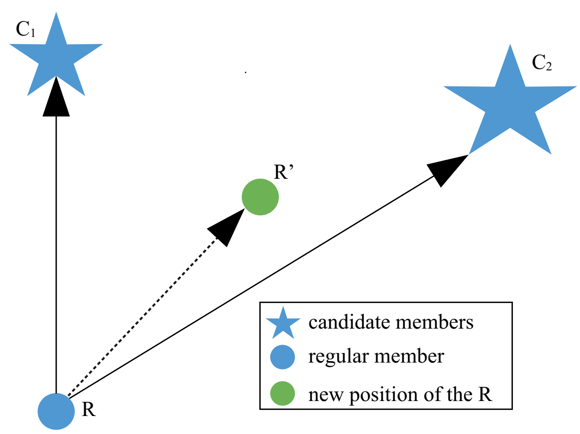

Figure 2. Biasing operation.

3. Big bang-big crunch algorithm

The BB-BC algorithm has two main phases: big bang and big crunch. In the first big bang phase, an initial population is created with random particles within the search space boundaries. Then the fitness values of all the particles are computed. Next, a contraction procedure is applied during the big crunch phase. In this phase, the center of mass (x^c) is calculated by accounting for the position and fitness value of each particle as follows:

$$x^c = \frac{\sum_{i=1}^{N} \frac{x^i}{f^i}}{\sum_{i=1}^{N} \frac{1}{f^i}}, \tag{3}$$

where x^i and f^i denote the position and fitness values of particle i, respectively, and N denotes the population size. Alternatively, the particle with the best fitness value can also be chosen as x^c. After the big crunch phase, the second big bang phase begins. In this phase, new particles are created around x^c by adding or subtracting random values, which decrease with each iteration, as follows [23]:

$$x^{new} = x^c + \frac{l.r}{k}, \tag{4}$$

where l denotes the upper limit of the search space, r is a random value between 0 and 1, k is the iteration step, and x^{new} is the location of the newly formed particle. The flowchart of the BB-BC algorithm is shown in Figure 3 [23].

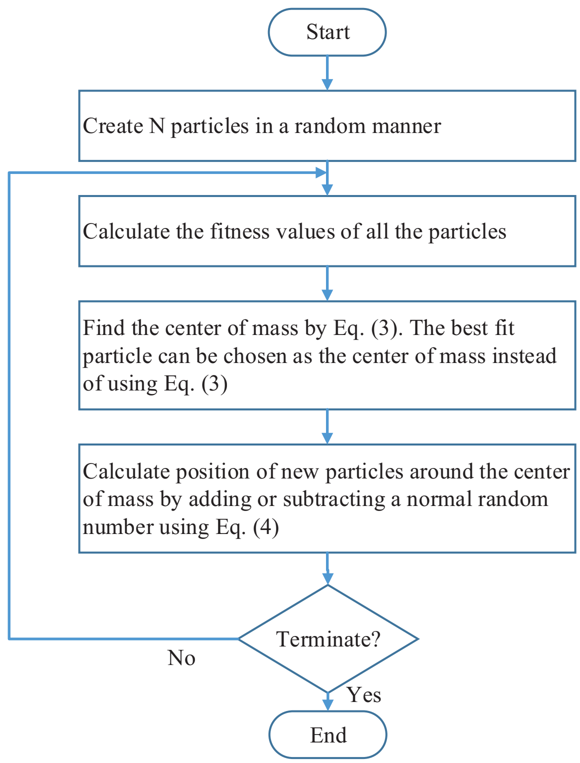

Figure 3. Flowchart of the BB-BC algorithm.

4. Hybrid parliamentary optimization and big bang-big crunch algorithm

In the intragroup competition phase of the POA, the regular members are biased toward the candidate members in the ratio of their fitness values, which allows the algorithm to explore the search space, thereby providing an exploration mechanism [17, 18]. By contrast, the BB-BC algorithm has an effective exploitation mechanism [31]. In the proposed method, steps of the BB-BC algorithm are added to the intragroup competition phase of the POA in order to improve the exploitation performance of the POA. Thus, the proposed method achieves a balance between exploration and exploitation.

After biasing and reassigning new candidates, the proposed method selects each regular member as x^c. Then the big-bang approach is adopted to search for better individuals around the regular members. P new individuals are created around x^c by using Eq. (4). After that, the fitness values of the individuals are calculated

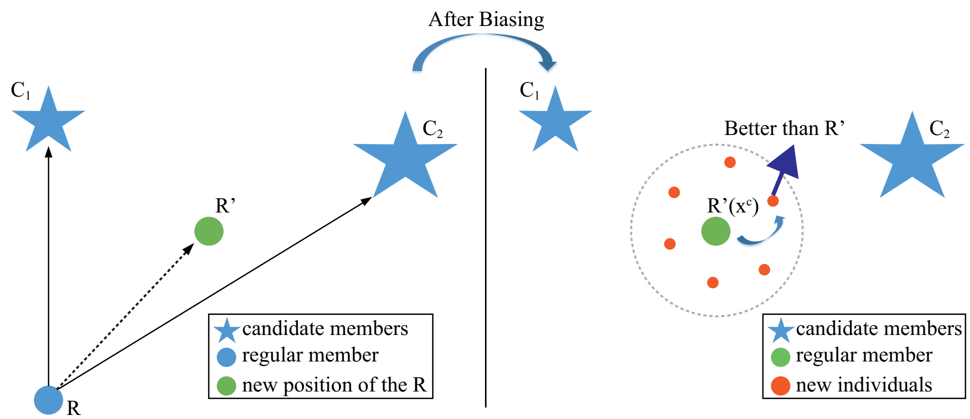

Figure 4. Search mechanism for finding better individuals around the regular members.

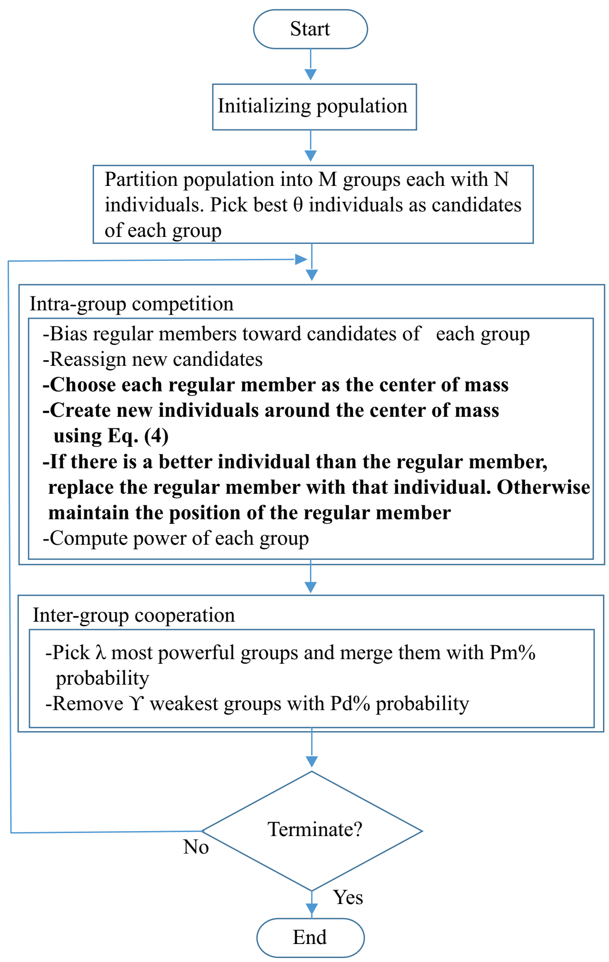

Figure 5. Flowchart of the HPO-BBBC algorithm.

and compared with the regular member. If there is a better individual than the regular member, the regular member is replaced with that individual; otherwise, the regular member maintains its position. This search mechanism provides an exploitation ability to the POA. This search mechanism is shown in Figure 4. The flowchart of the HPO-BBBC algorithm is shown in Figure 5.

5. Experiments and results

Mathematical test functions can be used to evaluate the performance of an optimization algorithm. Most of these functions have the same complexity as engineering problems. The difficulty ratings of the functions can be adjusted by changing the parameters [36]. Nine standard mathematical test functions and eight composition, rotated, shifted, and expanded functions selected from CEC 2005 were used to test the efficiency of the HPO-BBBC algorithm. The details of the standard test functions are summarized in Table 1 [37, 38]. Among these functions, the Sphere and Rosenbrock functions are unimodal functions (containing only one optimum), whereas the remaining functions are multimodal functions (containing many local optima but only one global optimum). Eight functions selected from CEC 2005 are summarized in Table 2 and more information about these test problems can be found in [39].

Table 1. Details of the standard mathematical test functions.

Name	Formulation	Property	Range	Optimum
Rastrigin	$F1(x) = 10d + \sum_{i=1}^{d}(x_i^2 - 10cos(2\pi x_i))$	Multimodal	±5.12	0
Rosenbrock	$F2(x) = \sum_{i=1}^{d-1}(100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2)$	Unimodal	±2.048	0
Sphere	$F3(x) = \sum_{i=1}^{d} x_i^2$	Unimodal	±5.12	0
Griewank	$F4(x) = 1 + \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos(\frac{x_i}{\sqrt{i}})$	Multimodal	±10	0
Ackley	$F5(x) = -20exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) -$ $exp(\frac{1}{d}\sum_{i=1}^{d} cos(2\pi x_i)) + 20 + e$	Multimodal	±32	0
Levy	$F6(x) = sin^2(\pi y_1) + \sum_{i=1}^{d-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] +$ $(y_d - 1)^2, y_i = 1 + (x_{i-1}/4)$ *for all i=1,...,d*	Multimodal	±10	0
Alpine	$F7(x) = \sum_{i=1}^{d} \lvert x_i sin(x_i) + 0.1x_i \rvert$	Multimodal	±10	0
Quintic	$F8(x) = \sum_{i=1}^{d} \lvert x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4 \rvert$	Multimodal	±10	0
Trigonometric	$F9(x) = 1 + \sum_{i=1}^{d} 8sin^2[7(x_i - 0.9)^2]$ $+6sin^2[14(x_i - 0.9)^2] + (x_i - 0.9)^2$	Multimodal	±500	1

The initial population was set as 30, and the maximum number of iterations was set as 1000 for all the algorithms for fair comparison. The performances of the algorithms for each test function were evaluated on the basis of the results obtained in 30 independent runs. The initial parameters used in the tests for algorithms are listed in Table 3. The comparative test results obtained from standard test functions (F1–F9) and the CEC 2005 functions (F10–F17) are summarized in Tables 4, 5, 6, and 7, respectively. They list the mean, best, worst, and standard deviation values for the 30 independent runs. The mean, best, and worst values represent the global convergence of the algorithms, and the standard deviation represents the stability of the algorithms [40].

Our results revealed that the HPO-BBBC outperformed the its component algorithms in all the test functions. In most benchmark problems, HPO-BBBC finds better values than GA, MVO, DA, CSA, and MFO, except the benchmarks shown in Table 4 (F1, F3, F5, F7, F8), Table 5 (F1, F3, F7, F8), Table 6 (F1, F3, F6, F8), and Table 7 (F12, F13). Moreover, the standard deviation values, which reflect the stability of the proposed method, were smaller for most test functions than those of the above-mentioned algorithms. To illustrate the convergence speeds of the algorithms, the convergence plots for the F1, F2, F4, F6, F8, F9, F10, F15, and F17 functions are shown in Figure 6, and it indicates that the HPO-BBBC algorithm converges faster than the mentioned algorithms in most of the benchmark functions.

Table 2. Details of the functions selected from CEC 2005.

	Function name	Property	Range	Dim	Optimum
F10	CEC-05-F1: Shifted Sphere Function	Unimodal	±100	10	–450
F11	CEC-05-F2: Shifted Schwefel's Problem 1.2	Unimodal	±100	10	–450
F12	CEC-05-F6: Shifted Rosenbrock's Function	Multimodal-basic	±100	10	390
F13	CEC-05-F9: Shifted Rastrigin's Function	Multimodal-basic	±5	10	–330
F14	CEC-05-F12: Schwefel's Problem 2.13	Multimodal-basic	$\pm\pi$	10	–460
F15	CEC-05-F13: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)	Multimodal-expanded	±5	10	–130
F16	CEC-05-F15: Hybrid Composition Function	Composition	±5	10	120
F17	CEC-05-F16: Rotated Hybrid Composition Function	Composition	±5	10	120

Table 3. Parameter settings of algorithms.

Algorithm	Parameter
POA	*M=6, N=5, Pd=0.001, Pm=0.007*
BB-BC	x^c= *Best particle*
HPO-BBBC	*M=6, N=5, Pd=0.001, Pm=0.007,* x^c=*regular members, p=5*
GA	*Crossover probability=0.7, mutation probability=0.01*
MVO	*min=0.2, max=1, p=6*
CSA	*AP=0.1, fl=2*
DA	β *= 3/2*

The Wilcoxon signed-rank test has been applied to statistically analyze the results between HPO-BBBC vs. POA and HPO-BBBC vs. BBBC. This test allows assessing result differences among two related methods [41, 42]. As shown in Table 8, P-values are less than 0.05 for all test functions (except F8). This make obvious the significant differences between the proposed algorithm and its components.

Table 9 shows the average running times (s) of the HPO-BBBC, its components, and other tested algorithms for 30 independent runs with 1000 iteration. In this study, to improve the exploitation capabilities of the POA, steps of the BB-BC algorithm have been added to the intragroup competition phase of the POA for searching for better individuals around the regular members. These steps have been created as extra components for the HPO-BBBC so the running time of the HPO-BBBC has increased. Although it shows that the HPO-BBBC consumes more running time than its components and other algorithms, the experimental results show that the proposed algorithm can effectively solve numerical global optimization problems and has a higher convergence speed in most benchmark problems. In other words, the HPO-BBBC can find better results in a smaller number of iterations than its components and other tested algorithms. Thus, the running time problems can be reduced by less than the number of iterations.

Table 4. Results of standard test functions with 10 dimensions (best results in bold font).

		POA	BBBC	HPO-BBBC	GA	MVO	DA	CSA	MFO
F1	Mean	19.845	24.157	6.4820	**0.5794**	9.4536	18.592	59.607	22.692
	Best	8.9546	5.9709	2.9861	**0.0806**	1.9923	2.9854	42.750	5.9697
	Worst	35.884	45.770	10.788	1.4810	21.890	45.703	75.252	50.813
	Std.D.	7.7137	10.430	2.1764	0.3351	4.8588	12.387	9.6995	10.343
F2	Mean	14.482	2.4359	**1.1205**	7.5421	4.5860	10.707	6.2089	5.0068
	Best	4.5968	0.5003	**0.3813**	0.4254	2.3807	1.2469	3.7216	0.6505
	Worst	56.768	7.3541	2.7945	68.107	5.7579	99.228	8.6481	9.4622
	Std.D.	13.241	1.6528	0.5602	14.547	0.8669	16.519	1.2715	1.7900
F3	Mean	0.0240	1.70E-5	4.82E-7	0.0029	1.07E-5	6.4E-10	7.3E-11	**3.9E-33**
	Best	2.03E-6	9.89E-6	8.8E-11	0.0005	4.96E-6	**0**	6.1E-12	2.9E-35
	Worst	0.2340	2.20E-5	3.31E-6	0.0077	1.98E-5	7.66E-9	3.4E-10	4.2E-32
	Std.D.	0.0559	3.01E-6	8.00E-7	0.0017	5.22E-6	1.74E-9	7.8E-11	8.2E-33
F4	Mean	0.0736	0.1312	**0.0029**	0.0594	0.0949	0.1887	0.0056	0.2032
	Best	0.0123	0.0196	1.93E-8	0.0109	0.0344	0.0320	**1.3E-10**	0.0295
	Worst	0.5458	0.3374	0.0167	0.1362	0.2356	0.5933	0.0246	0.6199
	Std D.	0.0919	0.0811	0.0047	0.0244	0.0537	0.1244	0.0070	0.1411
F5	Mean	5.0859	1.5043	0.0045	0.7779	0.0293	0.8586	1.5003	**5.E-15**
	Best	2.3223	0.0265	0.0002	0.2694	0.0168	**4.4E-15**	2.26E-5	**4.4E-15**
	Worst	9.7322	19.951	0.0180	1.5249	0.0440	3.0551	3.4041	8.0E-15
	Std.D.	1.7589	4.9495	0.0052	0.3028	0.0076	1.0035	1.1161	1.3E-15
F6	Mean	0.5112	2.9754	**7.25E-7**	0.0049	0.0089	0.508	0.0927	1.7989
	Best	0.0003	2.27E-5	1.42E-9	0.0015	9.87E-6	1.73E-8	1.52E-9	**1.5E-32**
	Worst	2.2773	9.2174	5.19E-6	0.0196	0.0896	4.0265	0.5438	10.125
	Std.D.	0.5705	2.5087	1.15E-6	0.0034	0.0268	0.8935	0.1105	2.8253
F7	Mean	0.5544	0.0546	0.0027	0.0138	0.2487	0.4053	0.0251	**7.2E-15**
	Best	0.0006	0.0014	0.0004	0.0048	0.0271	0.0023	1.54E-5	**1.7E-18**
	Worst	4.1337	0.2380	0.0112	0.0407	0.9082	2.0570	0.5028	4.4E-14
	Std.D.	1.0687	0.0636	0.0025	0.0092	0.2384	0.5072	0.0901	7.9E-15
F8	Mean	8.8271	0.2895	0.1259	1.0558	0.8444	5.8475	4.8827	**3.5E-15**
	Best	0.0911	0.2044	0.0135	0.4267	0.1735	0.0184	0.2973	**0**
	Worst	35.957	0.5130	0.2147	1.7485	1.7788	30.216	18.525	2.8E-14
	Std.D.	9.1292	0.0664	0.0530	0.3844	0.6086	7.8390	4.2949	6.6E-15
F9	Mean	565.81	22.090	**4.3082**	69.948	39.812	43.663	14.817	10.293
	Best	20.669	8.1708	**1.0001**	19.521	17.790	9.8978	4.1578	1.4486
	Worst	4026.9	36.773	16.169	108.37	51.135	115.07	34.186	27.824
	Std.D.	928.21	7.5627	3.8056	19.614	10.073	26.910	6.8172	6.8711

Table 5. Results of standard test functions with 20 dimensions (best results in bold font).

		POA	BBBC	HPO-BBBC	GA	MVO	DA	CSA	MFO
F1	Mean	73.511	54.304	22.625	**3.5574**	61.917	59.126	153.96	87.413
	Best	19.843	25.883	11.956	**1.1947**	34.845	21.217	87.357	42.783
	Worst	163.63	161.82	28.732	7.1123	88.603	116.84	257.34	141.42
	Std.D.	31.484	29.046	3.6199	1.2275	14.658	21.906	27.831	27.431
F2	Mean	98.978	14.399	**12.739**	38.367	17.609	18.600	18.556	13.4401
	Best	17.581	12.745	11.314	6.5879	15.850	15.915	17.610	**5.8397**
	Worst	247.99	17.497	13.817	73.769	19.155	23.137	19.498	20.587
	Std.D.	47.819	0.9822	0.6665	26.686	1.0094	1.6232	0.5272	4.4058
F3	Mean	1.4828	8.50E-5	1.57E-5	0.0173	0.0001	0.0223	2.39E-6	**1.5E-13**
	Best	0.0891	5.38E-5	9.28E-7	0.0064	0.0001	0.0001	3.59E-7	**1.6E-15**
	Worst	9.4540	0.0001	5.16E-5	0.0455	0.0002	0.1052	6.54E-6	8.8E-13
	Std.D.	1.8683	1.34E-5	1.48E-5	0.0090	5.50E-5	0.0270	1.81E-6	2.2E-13
F4	Mean	0.2504	0.0146	**4.82E-6**	0.0258	0.0324	0.0745	0.0024	0.1197
	Best	0.0538	1.89E-5	**2.53E-7**	0.0025	5.74E-5	0.0006	4.96E-6	0.0014
	Worst	0.7296	0.0973	1.82E-5	0.1173	0.1194	0.5032	0.0222	0.4678
	Std.D.	0.1742	0.0236	3.90E-6	0.0215	0.0312	0.1274	0.0058	0.1268
F5	Mean	11.289	1.4465	**0.0604**	1.7657	0.4517	3.8391	2.9594	6.3947
	Best	6.8554	0.0393	0.0096	1.2299	0.0760	1.4413	1.1584	**1.1E-6**
	Worst	15.872	19.898	1.1565	2.4623	1.8744	5.9212	4.3620	19.924
	Std.D.	2.4129	4.9261	0.2037	0.3202	0.7053	1.0810	1.0451	8.2449
F6	Mean	6.6845	9.3292	**0.0119**	0.0312	8.0285	3.1385	1.0612	14.865
	Best	1.3690	0.5440	**3.0E-6**	0.0112	0.4547	0.2648	0.0904	6.0901
	Worst	19.348	16.544	0.0896	0.0703	17.351	9.9220	3.3646	25.793
	Std.D.	3.6340	4.3173	0.0304	0.0122	5.6916	2.6534	0.8726	6.6497
F7	Mean	5.2226	0.8966	0.0780	**0.0597**	1.6431	6.9029	0.1881	1.1840
	Best	0.2377	0.0701	0.0021	0.0250	0.5381	0.5132	0.0174	**4.3E-9**
	Worst	18.591	10.105	0.3094	0.1230	5.3546	17.308	0.5959	8.8804
	Std.D.	4.1637	1.8212	0.0803	0.0259	1.4232	5.0331	0.1775	2.5464
F8	Mean	176.23	3.3407	1.6119	4.3535	5.5233	27.159	16.955	**1.6E-6**
	Best	27.436	0.7706	0.4141	2.3754	1.0909	9.2072	3.8331	**4.8E-7**
	Worst	3119.9	13.636	9.2314	6.0788	14.247	45.694	28.302	6.93E-6
	Std.D.	549.05	3.8124	1.5536	0.8266	3.5896	11.446	6.9066	1.57E-6
F9	Mean	16660	83.712	**27.881**	347.04	144.45	2479	81.534	53.793
	Best	2634	45.259	**1.8416**	189.21	105.43	33.700	37.870	16.155
	Worst	59308	112.66	65.803	706.23	221.89	9390	109.38	119.97
	Std.D.	15180	15.422	18.868	122.47	34.434	2341	21.088	27.834

6. Conclusion

Hybridization is a well-known technique for enhancing the performance of an algorithm. The main idea of hybridization is to combine the properties of two or more algorithms into a single algorithm. In this study, the

Table 6. Results of standard test functions with 30 dimensions (best results in bold font).

		POA	BBBC	HPO-BBBC	GA	MVO	DA	CSA	MFO
F1	Mean	139.38	76.724	36.315	**11.258**	125.11	96.731	246.45	173.34
	Best	42.851	38.836	27.846	**6.2049**	71.764	48.432	196.06	108.45
	Worst	240.37	146.05	47.608	16.770	197.16	131.69	292.72	243.19
	Std.D.	43.879	27.891	5.2941	2.3331	34.870	24.678	27.154	36.246
F2	Mean	215.02	25.444	**23.519**	84.587	27.975	38.405	28.454	91.209
	Best	104.67	21.393	21.574	23.883	25.318	30.097	27.020	**5.9869**
	Worst	392.52	28.256	25.210	140.70	29.347	60.744	29.153	438.41
	Std.D.	69.048	1.1609	0.9046	31.565	1.1471	8.5616	0.5419	112.26
F3	Mean	7.1565	0.0002	5.99E-5	0.1059	0.0007	0.5842	0.0002	**6.8E-7**
	Best	1.9739	0.0001	1.25E-5	0.0507	0.0005	0.0039	7.08E-5	**3.72E-8**
	Worst	23.898	0.0002	0.0001	0.1960	0.0011	1.4400	0.0004	4.59E-6
	Std.D.	5.2014	3.32E-5	3.28E-5	0.0364	0.0001	0.4392	0.0001	1.15E-6
F4	Mean	0.6679	0.0063	**1.75E-5**	0.0371	0.0106	0.1929	0.0031	0.2172
	Best	0.2867	3.41E-5	4.71E-6	0.0103	0.0003	0.0542	0.0001	**3.8E-9**
	Worst	1.0185	0.0271	4.03E-5	0.0889	0.0251	0.4591	0.0111	0.7500
	Std.D.	0.2051	0.0083	8.47E-6	0.0141	0.0086	0.1233	0.0039	0.2745
F5	Mean	15.111	1.3702	**0.0453**	3.0500	1.2810	6.0044	3.8421	16.770
	Best	10.910	0.0646	**0.0189**	2.3346	0.1570	4.0958	2.5867	2.1189
	Worst	17.508	20.127	0.0737	3.6406	2.9421	8.1291	5.9284	19.963
	Std.D.	1.7652	3.5862	0.0135	0.2942	0.7530	1.1756	0.9395	5.7066
F6	Mean	21.042	14.694	0.2059	**0.1939**	12.902	6.9293	2.1502	30.948
	Best	10.771	3.2707	**4.68E-5**	0.0821	0.9997	0.7056	0.3013	18.088
	Worst	56.300	24.852	0.7162	0.3201	50.234	20.024	6.5336	51.613
	Std.D.	8.4023	5.4543	0.1893	0.0628	13.861	5.4272	1.4687	9.1261
F7	Mean	14.542	2.9715	**0.2344**	0.2861	4.8346	8.7549	0.8523	4.7069
	Best	3.8052	0.1639	0.0114	0.1321	2.2619	1.5562	0.1343	**2.14E-5**
	Worst	38.282	23.393	0.6631	0.4355	7.5051	23.657	3.2352	13.320
	Std.D.	8.7992	4.6124	0.1737	0.0821	1.5118	6.1361	0.8137	4.6077
F8	Mean	918.58	8.2363	9.5809	12.626	19.764	61.308	50.008	**0.0165**
	Best	72.935	1.4663	2.7813	9.2785	11.593	29.339	15.424	**0.0024**
	Worst	6358.3	25.660	20.741	17.795	42.780	85.297	77.537	0.0658
	Std.D.	1473.5	6.7426	4.1867	2.1352	8.9336	15.046	17.949	0.0176
F9	Mean	72518	176.52	**90.487**	1428.2	238.99	7266.19	189.35	96.539
	Best	11158	141.50	**22.522**	844.54	195.22	1694	121.94	53.064
	Worst	253017	259.10	233.68	2474.9	296.45	38836	303.43	164.73
	Std.D.	51795	24.446	42.406	369.887	31.598	9336.17	51.363	33.600

HPO-BBBC algorithm was proposed for solving global numerical optimization problems using a combination of the POA and the BB-BC algorithm. The intragroup competition phase of the POA provides an exploration

Table 7. Results of CEC 2005 functions (best results in bold font).

		POA	BBBC	HPO-BBBC	GA	MVO	DA	CSA	MFO
F10	Mean	-362.741	-449.989	**-450**	-444.828	-449.996	-373.628	**-450**	**-450**
	Best	-449.996	-449.995	**-450**	-448.407	-449.998	**-450**	**-450**	**-450**
	Worst	-19.4705	-449.968	-450	-435.924	-449.993	-331.573	-450	-450
	Std.D.	154.292	0.00704	0	3.49304	0.00141	49.8798	0	0
F11	Mean	1773.873	-443.822	**-450**	1104.858	-449.972	-274.663	-449.945	-267.496
	Best	35.68258	-448.189	**-450**	31.1204	-449.985	-441.217	-449.994	**-450**
	Worst	4068.804	-429.607	-450	3197.778	-449.949	-105.779	-449.791	-19.47
	Std.D.	1368.145	5.321796	0	1077.415	0.010405	87.28489	0.054372	114.3102
F12	Mean	36185.39	412.4278	397.7567	1868.764	462.6775	8373.619	**397.6995**	597.8785
	Best	457.6932	399.6124	**394.0019**	617.6063	395.242	397.5998	394.2551	395.2381
	Worst	114197.1	549.3627	399.5399	4080.398	593.1746	24075.39	400.8513	1044.006
	Std.D.	40388.16	41.28796	1.568115	1060.257	70.73772	8245.912	1.944389	195.1874
F13	Mean	-313.739	-291.219	-320.211	**-328.055**	-313.262	-284.334	-301.743	-304.341
	Best	-326.02	-312.088	-324.053	**-329.24**	-323.034	-311.953	-312.091	-323.035
	Worst	-301.145	-239.706	-316.625	-326.997	-302.919	-257.361	-287.217	-285.216
	Std.D.	7.16026	19.46133	2.242528	0.613318	6.600104	18.47242	7.644954	11.87931
F14	Mean	3475.521	4565.467	**-451.012**	1656.562	1703.497	8127.568	-412.664	3916.584
	Best	12.02199	-448.529	**-459.76**	48.06925	-458.932	-426.998	-453.927	-455.066
	Worst	8389.517	18892.8	-444.362	5553.795	12904.89	20042.8	-203.248	12358.52
	Std.D.	3045.651	6295.205	4.572402	1805.15	4433.449	7376.842	71.8757	5202.762
F15	Mean	-117.222	-128.455	**-129.368**	-128.424	-128.649	-128.438	-128.783	-128.479
	Best	-128.211	-129.259	-129.56	-128.929	-129.374	**-129.594**	-129.393	-129.387
	Worst	-67.4785	-126.853	-129.17	-127.372	-127.298	-127.168	-127.97	-127.188
	Std.D.	16.98593	0.755526	0.104705	0.437258	0.686201	0.742833	0.471149	0.654516
F16	Mean	414.9083	630.7858	**250.4758**	376.7742	406.5959	653.5628	523.1934	544.1959
	Best	251.9286	447.0495	168.7983	**124.5589**	225.7706	430.1585	300.6117	255.3404
	Worst	745.2497	920.0004	327.8062	571.041	598.886	815.428	730.3999	837.6443
	Std.D.	157.6085	123.9433	47.00732	187.0318	131.5393	105.6845	138.9255	163.2489
F17	Mean	316.0176	336.7429	**253.7221**	303.6232	261.2031	398.4749	296.6478	290.9328
	Best	293.9723	242.4552	**217.811**	267.7282	224.7097	260.0353	258.4047	244.2714
	Worst	350.6052	416.4739	273.3766	363.9921	304.8998	621.7829	349.6444	392.1373
	Std.D.	19.92804	59.70134	16.07612	25.24678	26.35556	95.93782	28.99286	44.52406

mechanism. By contrast, the BB-BC algorithm has an effective exploitation mechanism. In the proposed method, steps of the BB-BC algorithm are added to the intragroup competition phase of the POA; thus, the proposed method achieves a balance between exploration and exploitation.

The performance of the HPO-BBBC algorithm was tested using nine standard mathematical test functions and eight composition, rotated, shifted, and expanded functions selected from CEC 2005. The experimental results were compared with those of the POA, the BB-BC algorithm, and five other metaheuristics, namely GA, MVO, DA, CSA, and MFO. It shows that the HPO-BBBC algorithm has higher convergence speed and produced better results than the above-mentioned algorithms in most benchmark problems. In the future, we plan to test the performance of the HPO-BBBC algorithm in data-mining techniques such as association rules and

Table 8. P-values of the HPO-BBBC vs. POA and HPO-BBBC vs. BBBC.

Function	POA	BBBC	Function	POA	BBBC	Function	POA	BBBC
F1 (10-d)	0.0000	0.0000	F4 (10-d)	0.0000	0.0000	F7 (10-d)	0.0000	0.0000
F2 (10-d)	0.0000	0.0018	F5 (10-d)	0.0000	0.0000	F8 (10-d)	0.0000	0.0000
F3 (10-d)	0.0000	0.0000	F6 (10-d)	0.0000	0.0000	F9 (10-d)	0.0000	0.0000
Function	POA	BBBC	Function	POA	BBBC	Function	POA	BBBC
F1 (20-d)	0.0000	0.0000	F4 (20-d)	0.0000	0.0000	F7 (20-d)	0.0000	0.0000
F2 (20-d)	0.0000	0.0000	F5 (20-d)	0.0000	0.0000	F8 (20-d)	0.0000	0.0818
F3 (20-d)	0.0000	0.0000	F6 (20-d)	0.0000	0.0000	F9 (20-d)	0.0000	0.0000
Function	POA	BBBC	Function	POA	BBBC	Function	POA	BBBC
F1 (30-d)	0.0000	0.0000	F4 (30-d)	0.0000	0.0000	F7 (30-d)	0.0000	0.0000
F2 (30-d)	0.0000	0.0000	F5 (30-d)	0.0000	0.0000	F8 (30-d)	0.0000	0.2801
F3 (30-d)	0.0000	0.0000	F6 (30-d)	0.0000	0.0000	F9 (30-d)	0.0000	0.0000
Function	POA	BBBC	Function	POA	BBBC	Function	POA	BBBC
F10	0.0000	0.0000	F13	0.0168	0.0000	F16	0.0006	0.0000
F11	0.0000	0.0000	F14	0.0000	0.0000	F17	0.0000	0.0000
F12	0.0000	0.0000	F15	0.0000	0.0000			

Table 9. Average times (s) of the algorithms.

	POA	BBBC	HPO-BBBC	GA	MVO	DA	CSA	MFO
F1	2.95	0.46	5.80	1.02	0.81	19.47	0.33	0.43
F2	2.65	0.39	5.01	0.82	0.79	19.45	0.34	0.46
F3	2.49	0.38	5.03	0.75	0.74	19.03	0.34	0.41
F4	2.53	0.37	5.26	0.90	0.87	25.18	0.49	0.47
F5	2.61	0.33	5.16	0.81	0.83	19.22	0.43	0.42
F6	2.97	0.54	5.96	1.59	0.98	20.24	0.37	0.55
F7	2.56	0.32	5.01	0.78	0.75	20.91	0.35	0.32
F8	2.81	0.49	6.01	1.21	0.89	19.13	0.49	0.48
F9	2.63	0.37	4.79	1.01	0.74	18.39	0.36	0.34
F10	128	69	319	134	72	122	68	68
F11	135	70	331	135	73	126	69	70
F12	137	70	333	134	73	121	68	68
F13	138	71	332	135	74	114	68	68
F14	146	75	354	147	80	128	72	74
F15	138	70	340	136	72	116	68	69
F16	402	207	974	401	215	301	205	202
F17	416	219	1022	427	226	318	216	215

classification. The efficiency of the HPO-BBBC algorithm can be improved through some modifications, e.g., chaotic maps could be embedded to create the initial population instead of using random numbers. Moreover, generalization of the HPO-BBBC for multiobjective optimization problems may also be one of the further works.

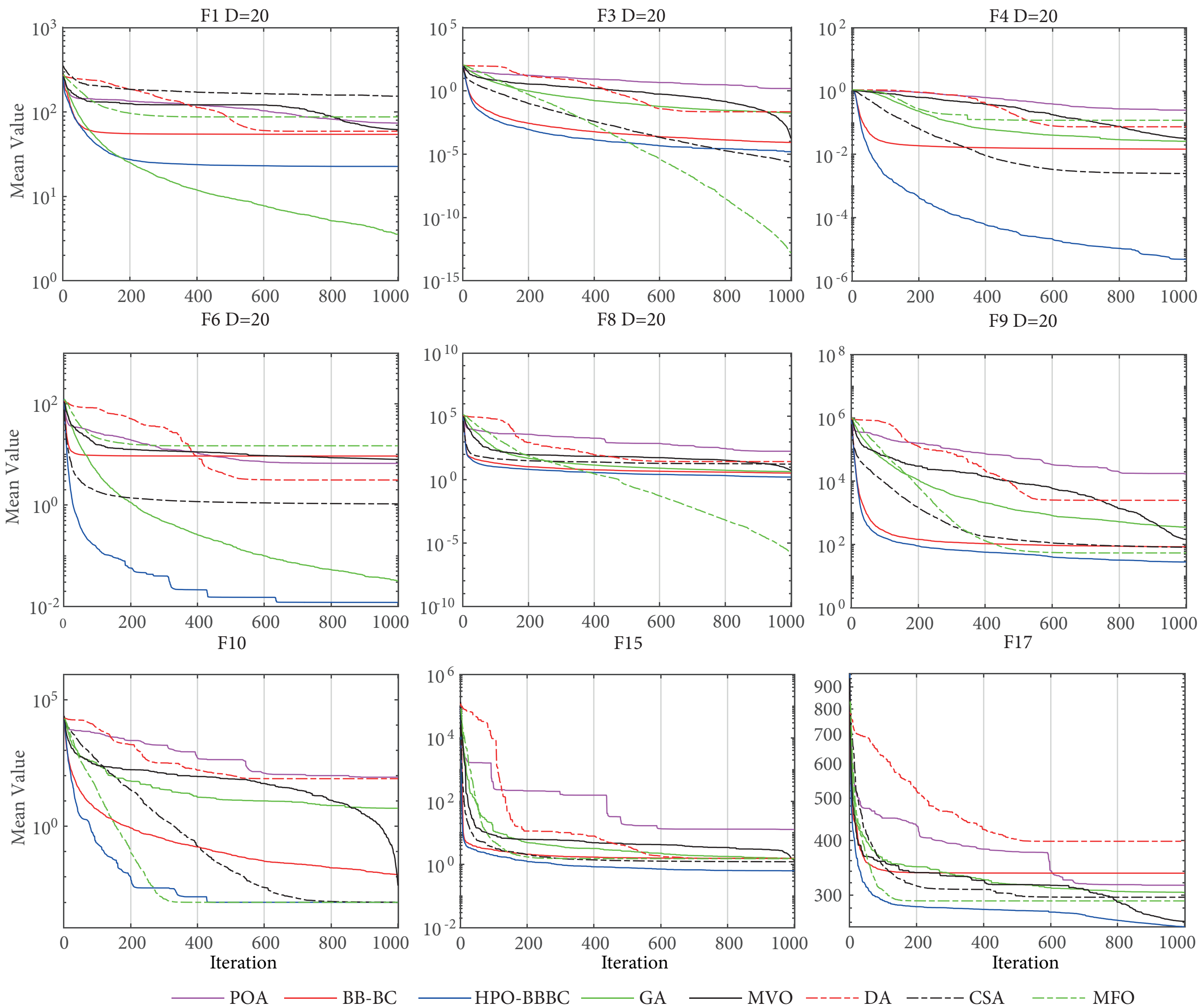

Figure 6. Convergence plots of F1, F3, F4, F6, F8, F9, F10, F15, and F17.

References

[1] Kızıloluk S, Alataş B. Automatic mining of numerical classification rules with parliamentary optimization algorithm. Adv Electr Comput Eng 2015; 15: 17-24.

[2] Holland JH. Adaptation in Natural and Artificial Systems. Ann Arbor, MI, USA: University of Michigan Press, 1975.

[3] De Castro L, Timmis J. Artificial Immune Systems: A New Computational Intelligence Approach. London, UK: Springer-Verlag, 2002.

[4] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009; 179: 2232-2248.

[5] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: IEEE Congress on Evolutionary Computation; 25–28 September 2007; Singapore. New York, NY, USA: IEEE. pp. 4661-4667.

[6] Alatas B. ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 2011; 38: 13170-13180.

[7] Malek M, Guruswamy M, Owens H, Pandya M. A Hybrid Algorithm Technique. Technical Report. Austin, TX, USA: University of Texas at Austin Department of Computer Sciences, 1989.

[8] Farnad B, Jafarian A, Baleanu D. A new hybrid algorithm for continuous optimization problem. Appl Math Modell 2018; 55: 652-673.

[9] Noack MM, Funke SW. Hybrid genetic deflated newton method for global optimisation. J Comput Appl Math 2017; 325: 97-112.

[10] Aydilek İB. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. Appl Soft Comput 2018; 66: 232-249.

[11] Zhang X, Kang Q, Cheng J, Wang X. A novel hybrid algorithm based on biogeography-based optimization and grey wolf optimizer. Appl Soft Comput 2018; 67: 197-214.

[12] Zou F, Chen D, Lu R. Hybrid hierarchical backtracking search optimization algorithm and its application. Arab J Sci Eng 2018; 43: 993-1014.

[13] Assad A, Deep K. A Hybrid harmony search and simulated annealing algorithm for continuous optimization. Inf Sci 2018; 450: 246-266.

[14] Jadon SS, Tiwari R, Sharma H, Bansal JC. Hybrid artificial bee colony algorithm with differential evolution. Appl Soft Comput 2017; 58: 11-24.

[15] Sree Ranjini KS, Murugan S. Memory based hybrid dragonfly algorithm for numerical optimization problems. Expert Syst Appl 2017; 83: 63-78.

[16] Nabil E. A modified flower pollination algorithm for global optimization. Expert Syst Appl 2016; 57: 192-203.

[17] Borji A. A new global optimization algorithm inspired by parliamentary political competitions. In: 6th Mexican International Conference on Artificial Intelligence; 4–10 November 2007; Aguascalientes, Mexico. pp. 61-71.

[18] Borji A, Hamidi M. A new approach to global optimization motivated by parliamentary political competitions. Int J Innov Comput 2009; 5: 1643-1653.

[19] De-Marcos L, García A, García E, Martínez JJ, Gutiérrez JA, Barchino R, Otón S. An adaptation of the parliamentary metaheuristic for permutation constraint satisfaction. In: IEEE Congress on Evolutionary Computation; 18–23 July 2010; Barcelona, Spain. New York, NY, USA: IEEE. pp. 1-8.

[20] Altunbey F, Alatas B. Overlapping community detection in social networks using parliamentary optimization algorithm. Int J Comput Netw Appl 2015; 2: 12-19.

[21] Kiziloluk S, Ozer AB. Web pages classification with parliamentary optimization algorithm. Int J Software Eng Knowl Eng 2017; 27: 499-513.

[22] Pekel E, Soner Kara S. Passenger flow prediction based on newly adopted algorithms. Appl Artif Intell 2017; 31: 64-79.

[23] Erol OK, Eksin I. A new optimization method: big bang–big crunch. Adv Eng Software 2006; 37: 106-111.

[24] Hatamlou A, Abdullah S, Hatamlou M. Data clustering using big bang–big crunch algorithm. In: Innovative Computing Technology; 13–15 December 2011; Tehran, Iran. pp. 383-388.

[25] Othman MM, El-Khattam W, Hegazy YG, Abdelaziz AY. Optimal placement and sizing of distributed generators in unbalanced distribution systems using supervised big bang-big crunch method. IEEE T Power Syst 2015; 30: 911-919.

[26] Kaveh A, Talatahari S. A discrete big bang-big crunch algorithm for optimal design of skeletal structures. Asian J Civ Eng 2010; 11: 103-122.

[27] Xing B, Gao WJ. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Cham, Switzerland: Springer International Publishing, 2014.

[28] Zandi Z, Afjei E, Sedighizadeh M. Hybrid big bang-big crunch optimization based optimal reactive power dispatch for voltage stability enhancement. J Theor Appl Inf Technol 2013; 47: 537-546.

[29] Ahmadi S, Abdi S. Application of the hybrid big bang–big crunch algorithm for optimal sizing of a stand-alone hybrid PV/wind/battery system. Sol Energy 2016; 134: 366-374.

[30] Hora C, Secui DC, Bendea G, Dzitac S. BB-BC-CG algorithm for operational reliability modeling of hydro generator groups. Procedia Comput Sci 2016; 91: 1088-1095.

[31] Sedighizadeh M, Mahmoodi MM, Soltanian M. Parameter identification of proton exchange membrane fuel cell using a hybrid big bang-big crunch optimization. In: 5th Conference on Thermal Power Plants; 10–11 June 2014; Tehran, Iran. New York, NY, USA: IEEE. pp. 35-39.

[32] Mirjalili S, Mirjalili SM, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Applic 2016; 27: 495-513.

[33] Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 2016; 169: 1-12.

[34] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Applic 2016; 27: 1053-1073.

[35] Mirjalili S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 2015; 89: 228–249.

[36] Ozer AB. CIDE: Chaotically initialized differential evolution. Expert Syst Appl 2010; 37: 4632-4641.

[37] Lux T. Convergence rate evaluation of derivative-free optimization techniques. In: International Workshop on Machine Learning, Optimization and Big Data; 26–29 August 2016; Volterra, Italy. pp. 246-256.

[38] Jamil M, Yang XS. A literature survey of benchmark functions for global optimisation problems. Int J Math Modell Numer Optim 2013; 4: 150-194.

[39] Suganthan P, Hansen N, Liang J, Deb K, Chen Y, Auger A, Tiwari S. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL report 2005.

[40] Wu B, Qian C, Ni W, Fan S. Hybrid harmony search and artificial bee colony algorithm for global optimization problems. Comput Math Appl 2012; 64: 2621-2634.

[41] Guvenc U, Katircioglu F. Escape velocity: a new operator for gravitational search algorithm. Neural Comput Appl 2019; 31: 27-42.

[42] Wilcoxon F. Individual comparisons by ranking methods. Biometrics Bulletin 1945; 6: 80–83.