Best Guided Backtracking Search Algorithm for Numerical Optimization Problems

Wenting Zhao[1], Lijin Wang[1,2], Bingqing Wang[1], and Yilong Yin[1,3](✉)

[1] School of Computer Science and Technology,
Shandong University, Jinan 250101, China
ylyin@sdu.edu.cn
[2] College of Computer and Information Science,
Fujian Agriculture and Forestry University, Fuzhou 350002, China
[3] School of Computer Science and Technology,
Shandong University of Finance and Economics, Jinan 250014, China

Abstract. Backtracking search algorithm is a promising stochastic search technique by using its historical information to guide the population evolution. Using historical population information improves the exploration capability, but slows the convergence, especially on the later stage of iteration. In this paper, a best guided backtracking search algorithm, termed as BGBSA, is proposed to enhance the convergence performance. BGBSA employs the historical information on the beginning stage of iteration, while using the best individual obtained so far on the later stage of iteration. Experiments are carried on the 28 benchmark functions to test BGBSA, and the results show the improvement in efficiency and effectiveness of BGBSA.

Keywords: Backtracking search algorithm · Best guided · Historical information · Numerical optimization problems

1 Introduction

Optimization plays a vital role in various fields, including decision science and physical system. Generally speaking, the first step of solving the optimization problem is to specify the objective function which describes the relationship between the variables and constraints. Then, we select the appropriate optimization method to reach the global optimum according to the characteristics of the objective function. When the objective function turns out to be complex, non-linear or non-differential, evolutionary algorithm (EA) is chosen to find the global optimum. EA is expected to reach a problems global minimum value quickly with a small number of control parameters and low computational cost. Experts have come up with numerous EAs, for example, genetic algorithm(GA) [1], differential evolutionary algorithm (DE) [2], ant colony optimization algorithm (ACO) [3], particle swarm optimization algorithm (PSO) [4], artificial bee colony algorithm (ABC) [5], biogeography-based optimization algorithm (BBO) [6] and backtracking search optimization algorithm (BSA) [7].

F. Lehner and N. Fteimi (Eds.): KSEM 2016, LNAI 9983, pp. 414–425, 2016.
DOI: 10.1007/978-3-319-47650-6_33

BSA is a novel population-based nature inspired optimization algorithm with a simple structure which only needs one input parameter and easy to be implemented. The experiments has been carried out on benchmark CEC-2005 and CEC-2011 and BSA shows the promising performance [7]. In addition, BSA has been applied to different kinds of engineering optimization problems recently due to its strong solving ability [8–10].

Similar to common evolutionary computation method, BSA employs mutation, crossover and selection three basic genetic operators. BSA is a double-population algorithm that employs both the historical and current populations. In the process of trial vectors generation, BSA maintains a block of memory to store the information derived from history population. During selection stage, BSA selects better individuals based on a greedy strategy guiding population towards the global optimum. BSA has a powerful exploration capability and achieves good results in solving multi-model problems. However, influenced by historical experience, the convergence speed of BSA slows down and prejudice

2 Backtracking Search Optimization Algorithm

The backtracking search optimization algorithm was first proposed by Civicioglu [7]. BSA is a novel stochastic population-based algorithm for real-valued numerical optimization problems. The structure of BSA is described as follows.

Initialization. At the first step, BSA initializes the current population P and the historical population $OldP$ containing N individuals according to the following form:

$$p_{ij} = p_j^{\min} + (p_j^{\max} - p_j^{\min}) \times rand(0,1) \tag{1}$$

$i = 1, 2, ..., N$, $j = 1, 2, ..., D$ where N is the population size and D is the problem dimension; p_i is the i-th individual in the current population P, $p_j^{\max}$ and $p_j^{\min}$ are the upper bound and lower bound of dimension j, and $rand(0, 1)$ generates a random number distributed uniformly from 0 to 1.

Selection-I. At this step, the selection strategy is used to select the historical population which will guide the population evolution in mutation step. Before the historical population is selected, BSA has a option of updating the historical population according to Eq. (2) where a and b are randomly generated numbers distributed uniformly over the range $(0, 1)$. Then, the order of the individuals in $OldP$ is changed randomly according to Eq. (3). The mechanism ensures that BSA is able to utilize randomly selected previous generation as the historical population.

$$OldP := \begin{cases} P, & a < b \\ OldP, & \text{otherwise} \end{cases} \tag{2}$$

$$OldP := permuting(OldP) \tag{3}$$

Mutation. BSA's mutation process generates the initial form of trial vectors by Eq. (4). The search-direction matrix $(OldP - P)$ is calculated and the amplitude is controlled by F which is generated randomly over the standard normal distribution range (0,3). Due to the utilization of $OldP$, BSA takes partial advantage of experiences from previous generations.

$$Mutant = P + F \times (OldP - P) \tag{4}$$

Crossover. A nonuniform and more complex crossover strategy is designed in BSA. The crossover process generates the final form of the trial population V. Firstly, a binary integer-valued matrix map of size $N * D$ is obtained. Secondly, it depends on the value of map for BSA to update the relevant dimensions of mutant individual by using the relevant individual in P. This crossover process can be presented in Eq. (5).

$$v_{ij} = \begin{cases} p_{ij}, & map_{ij} = 1 \\ mutant_{ij}, & \text{otherwise} \end{cases} \tag{5}$$

Algorithm 1. Best Individual Guided Backtracking Search Algorithm

Input N- population size, *dimRate*- crossover parameter,
$MaxFes$- total number of function evaluation, α- stage control parameter
Output X_{best}- the best individual
Step 1: Initialization
initialize $P = \{p_1, ...p_N\}$, $OldP = \{oldp_1, ...oldp_N\}$,
and current number of function evaluation $FES = 0$
Step 2: Selection-I
redefine *OldP* using Eqs. (2) and (3)
Step 3: Mutation
Step 3.1 generate *mutant* according to Eq. (4)
if $FES < \alpha * MaxFes$
Step 3.2 generate *mutant* according to Eq. (7)
if $FES \geq \alpha * MaxFes$
Step 4: Crossover
generate trial vector V using Eq. (5)
Step 5: Selection-II
update current population P according to Eq. (6)
Step 6: Variable Update
select the best individual X_{best} and update FES
Step 7: Stopping Criteria
If stopping criteria is fulfilled, then output X_{best}; otherwise go to **Step 2**

Selection-II. At this step, the individuals with better fitness value are selected and evolved in the next iteration until the stop condition is satisfied. The greedy selection mechanism is shown in Eq. (7) where $f(v_i)$ and $f(p_i)$ represents the fitness value of v_i and p_i.

$$p_i = \begin{cases} v_i, & if\ f(v_i) \leq f(p_i) \\ p_i, & \text{otherwise} \end{cases} \tag{6}$$

3 Best Guided Backtracking Search Algorithm

BSA has a powerful exploration capability but a relatively slow convergence speed, since the algorithm makes full use of historical experiences to guide the evolution. The enhanced backtracking search algorithm, combining the historical experience and the experience from best individual, is proposed. The character of the best individual obtained at later stage of iteration is able to provide efficient information contributing to convergence acceleration. The pseudo-code of BGBSA is summarized in Algorithm 1.

The whole iteration period is divided into two stages: early stage and later stage. To do this, a control parameter α over the range $(0, 1)$ is used. If the sum of function evaluations till the current iteration are less than the α scale of the maximum function evaluations, then BGBSA is regarded on the early stage of iteration, else, on the later stage of iteration. On the early stage, the population is still guided by the historical information. In this case, the exploration capacity is kept. On the later stage, to improve the convergence, the population is guided by the information derived from the best individual obtained so far, resulting in the improvement on the exploitation capacity. As a result, the balance of exploration and exploitation is kept. The best guided operator is presented in Eq. (7), where P_{best} is the best individual obtained so far, and F is the same as Eq. (4).

$$BGMutant = P + F \times (P_{best} - P) \tag{7}$$

4 Experiments

4.1 Experimental Settings

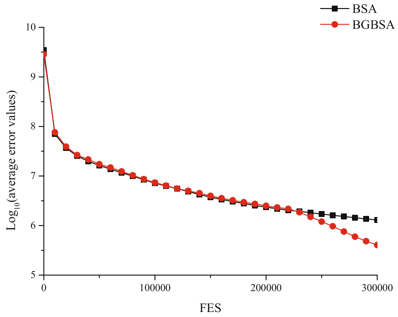
In this section, BGBSA is verified on CEC-2013 benchmark test suite including unimodal functions $F_1 - F_5$, basic multimodal functions $F_6 - F_{20}$ and composition functions $F_{21} - F_{28}$ [16]. The parameter of crossover rate *dimRate* is set to 1.0, stage control parameter α is equal to 0.75, the population size N is 30 and the dimension is set to 30. For each problem, 25 independent runs are performed. The algorithm is stopped if a maximum of 300000 iterations for each run is met. In this paper, the amplitude control parameter F mentioned in Eq. (4) and in Eq. (7) which is subject to lévy distribution is implemented using matlab statistics and machine learning toolbox. To evaluate the performance of BGBSA, the average and standard deviation of the best error values, presented as "$AVG_{ER} \pm STD_{ER}$", are used in the following result tables. The results with high quality are marked in bold. In addition, we examine the significant differences between two algorithms utilizing Wilcoxon signed-rank test at the 5 % significance level. The statistical result is listed at the bottom of tables.

4.2 The Effect of BGBSA

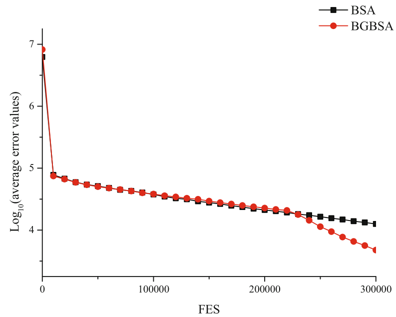
Table 1 summarizes the results obtained by BSA and BGBSA. For uni-modal functions $F_1 - F_5$, BGBSA gains global optimum on F_1, F_5 and brings superior solutions on F_2 and F_4. For F_3, BGBSA exhibits a little inferior to BSA. However, it is not significant according to the Wilcoxon test results. For basic multimodal functions $F_6 - F_{20}$, BGBSA overall outperforms BSA. BGBSA obtains solutions with high quality on F_7, $F_9 - F_{18}$ and F_{20} with the help of average error values. BGBSA behaves poor solving ability on F_{19}, but these two methods are not significant. For composition functions $F_{21} - F_{28}$, BGBSA gains superior solutions on F_{22}, F_{23}, F_{25}, F_{27}, equal ones on F_{26}, F_{28}, and inferior ones on F_{21}, F_{24}. However, based on the Wilcoxon test results, BGBSA shows similar performance on F_{21}, F_{24} compared with BSA. Summarily, BGBSA wins and ties BSA on 12 and 16 out of 28 benchmark functions according to "$+/=/-$", respectively. Particularly, the proposed method shows promising performance on basic multimodal functions.

Table 1. Error values obtained by BSA and BGBSA for 30-dimensional CEC-2013 benchmark functions

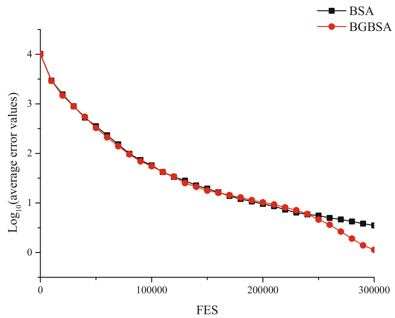
	BSA	BGBSA		P value
	$AVG_{Er} \pm STD_{Er}$	$AVG_{Er} \pm STD_{Er}$		
F_1	1.01e−30 ± 3.49e−30	1.01e−30 ± 3.49e−30	=	1.000000
F_2	1.37e+06 ± 5.35e+05	**4.26e+05 ± 2.13e+05**	+	0.000020
F_3	**4.54e+06 ± 4.60e+06**	6.44e+06 ± 9.98e+06	=	0.861162
F_4	1.27e+04 ± 3.58e+03	**4.95e+03 ± 1.93e+03**	+	0.000018
F_5	0.00e+00 ± 0.00e+00	5.05e−31 ± 2.52e−30	=	1.000000
F_6	2.74e+01 ± 2.47e+01	2.74e+01 ± 2.64e+01	=	0.492633
F_7	6.82e+01 ± 1.35e+01	**5.94e+01 ± 1.36e+01**	=	0.082653
F_8	2.09e+01 ± 6.72e−02	2.09e+01 ± 3.98e−02	=	0.618641
F_9	2.73e+01 ± 2.75e+00	**2.58e+01 ± 2.86e+00**	=	0.078001
F_{10}	1.90e−01 ± 1.42e−01	**1.49e−01 ± 1.59e−01**	=	0.287862
F_{11}	7.96e−02 ± 2.75e−01	**3.98e−02 ± 1.99e−01**	=	1.000000
F_{12}	8.71e+01 ± 2.14e+01	**8.36e+01 ± 1.74e+01**	=	0.492633
F_{13}	1.49e+02 ± 2.53e+01	**1.42e+02 ± 2.19e+01**	=	0.287862
F_{14}	3.56e+00 ± 1.73e+00	**1.52e+00 ± 1.28e+00**	+	0.000157
F_{15}	3.81e+03 ± 4.16e+02	**3.48e+03 ± 4.60e+02**	+	0.002259
F_{16}	1.26e+00 ± 1.66e−01	**1.10e+00 ± 3.01e−01**	+	0.021418
F_{17}	3.09e+01 ± 1.75e−01	**3.06e+01 ± 1.06e−01**	+	0.000029
F_{18}	1.16e+02 ± 1.99e+01	**9.78e+01 ± 1.90e+01**	+	0.002947
F_{19}	**1.07e+00 ± 2.11e−01**	1.13e+00 ± 2.38e−01	=	0.312970
F_{20}	1.14e+01 ± 4.91e−01	**1.10e+01 ± 6.39e−01**	+	0.017253
F_{21}	**2.67e+02 ± 8.00e+01**	2.90e+02 ± 4.91e+01	=	0.142970
F_{22}	4.33e+01 ± 1.72e+01	**2.59e+01 ± 1.12e+01**	+	0.000602
F_{23}	4.36e+03 ± 5.00e+02	**4.09e+03 ± 3.81e+02**	+	0.042207
F_{24}	**2.33e+02 ± 1.03e+01**	2.35e+02 ± 1.16e+01	=	0.396679
F_{25}	2.89e+02 ± 8.80e+00	**2.81e+02 ± 1.44e+01**	+	0.028314
F_{26}	2.00e+02 ± 1.32e−02	2.00e+02 ± 7.07e−03	+	0.000029
F_{27}	8.89e+02 ± 1.45e+02	**8.85e+02 ± 1.10e+02**	=	0.798248
F_{28}	3.00e+02 ± 1.95e−13	3.00e+02 ± 1.62e−13	=	0.637352
+/=/−				12/16/0



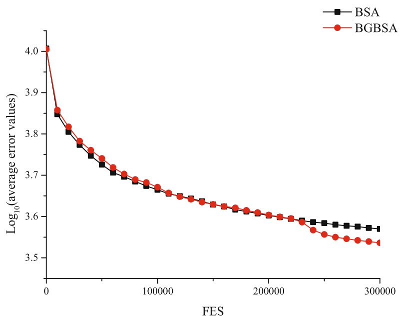
(a) F_2



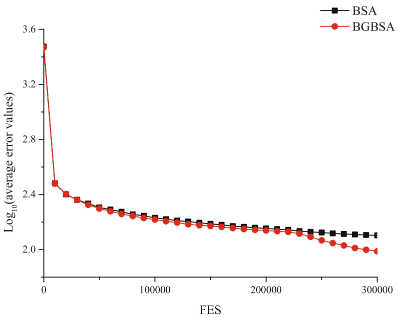
(b) F_4



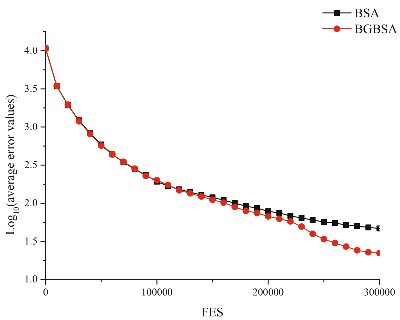
(c) F_{14}



(d) F_{15}



(e) F_{18}



(f) F_{22}

Fig. 1. The convergence curves of BSA and BGBSA for selected benchmark functions.

Table 2. Average ranking obtained by BGBSA and three variants of BSA for 10, 30 and 50-dimensional CEC-2013 benchmark functions

D	BGBSA	HBD	IBSA	COOBSA
10	1.93	1.93	2.14	4
30	1.84	1.89	2.27	4
50	1.75	1.93	2.32	4

Furthermore, convergence curves for six selected functions F_2, F_4, F_{14}, F_{15}, F_{18}, and F_{22} are plotted in Fig. 1 to investigate the convergence speed. It can be observed from Fig. 1 that BGBSA has a better convergence performance than BSA on the later stage of iteration, and shows the exploitation capacity.

In terms of the accuracy of solutions and convergence, BGBSA is overall superior to BSA. This is because that the information dervied from the best solution improves the exploitation capacity of BGBSA, and keeps the balance of exploration and exploitation.

4.3 Compared with Other Variants of BSA

BGBSA is compared with three BSA approaches, called COOBSA [11], HBD [14] and IBSA [15]. For each algorithm, parameter settings are kept the same. The population size N is equal to the dimension D at 30 or 50, while it is 30 in the case of $D = 10$. The stopping criterion is that a maximum of function evaluation times equal to $10000 * D$ is reached. The average rankings of the four algorithms by the Friedman test for CEC-2013 test suite at $D = 10, 30, 50$ are presented in Table 2.

In Table 2, each row shows the results obtained by the Friedman test at different dimension $D = 10, 30, 50$ respectively. In the top row of the table, it can be seen that BGBSA and HBD offer the best performance, followed by IBSA and COOBSA at $D = 10$. At $D = 30$, BGBSA is slightly better than the second best performance algorithm HBD, followed by IBSA and COOBSA. When the dimension is increased to 50, BGBSA is the best and superior to the other three algorithms. It can be concluded that the best guided BSA shows better performance than HBD, IBSA and COOBSA, and exhibits higher stability.

4.4 Compared with Other Algorithms

BGBSA is compared with five state-of-art algorithms which do not combine with BSA named NBIPOP-aCMA [17], fk-PSO [18], SPSO2011 [19], SPSOABC [20], and PVADE [21] which were proposed during CEC-2013 Special Session &Competition on Real-Parameter Single Objective Optimization. To compare fair and conveniently, the maximum of fitness evaluations is set to 300000 for each algorithm. Comparison results are listed in Table 3. In addition, the results of Friedman test similarly done in [22] for six problems are presented in Table 4.

Table 3. Error values obtained by BGBSA and 5 compared algorithms for CEC-2013 benchmark functions at $D = 30$

	NBIPOP-aCMA	fk-PSO	SPSO2011	SPSOABC	PVADE	BGBSA
	$AVG_{Er} \pm STD_{Er}$	$AVG_{Er} \pm STD_{Er}$	$AVG_{Er} \pm STD_{Er}$	$AVG_{Er} \pm STD_{Er}$	$AVG_{Er} \pm STD_{Er}$	$AVG_{Er} \pm STD_{Er}$
F_1	**0.00E+00** ± **0.00E+00**	**0.00E+00** ± **0.00E+00**	**0.00E+00** ± **0.00E+00**	**0.00E+00** ± **0.00E+00**	**0.00E+00** ± **0.00E+00**	**1.01e−30** ± **3.49e−30**
F_2	**0.00E+00** ± **0.00E+00**	1.59E+06 ± 8.03E+05	3.38E+05 ± 1.67E+05	8.78E+05 ± 1.69E+06	2.12E+06 ± 1.56E+06	4.26e+05 ± 2.13e+05
F_3	**0.00E+00** ± **0.00E+00**	2.40E+08 ± 3.71E+08	2.88E+08 ± 5.24E+08	5.16E+07 ± 8.00E+07	1.65E+03 ± 2.83E+03	6.44e+06 ± 9.98e+06
F_4	**0.00E+00** ± **0.00E+00**	4.78E+02 ± 1.96E+02	3.86E+04 ± 6.70E+03	6.02E+03 ± 2.30E+03	1.70E+04 ± 2.85E+03	4.95e+03 ± 1.93e+03
F_5	**0.00E+00** ± **0.00E+00**	**0.00E+00** ± **0.00E+00**	5.42E−04 ± 4.91E−05	**0.00E+00** ± **0.00E+00**	1.40E−07 ± 1.86E−07	**5.05e−31** ± **2.52e−30**
F_6	**0.00E+00** ± **0.00E+00**	2.99E+01 ± 1.76E+01	3.79E+01 ± 2.83E+01	1.09E+01 ± 1.09E+01	8.29E+00 ± 5.82E+00	2.74e+01 ± 2.64e+01
F_7	2.31E+00 ± 6.05E+00	6.39E+01 ± 3.09E+01	8.79E+01 ± 2.11E+01	5.12E+01 ± 2.04E+01	**1.29E+00** ± **1.22E+00**	5.94e+01 ± 1.36e+01
F_8	**2.09E+01** ± **4.80E−02**	**2.09E+01** ± **6.28E−02**	**2.09E+01** ± **5.89E−02**	**2.09E+01** ± **4.92E−02**	**2.09E+01** ± **4.82E−02**	**2.09e+01** ± **3.98e−02**
F_9	**3.30E+00** ± **1.38E+00**	1.85E+01 ± 2.69E+00	2.88E+01 ± 4.43E+00	2.95E+01 ± 2.62E+00	6.30E+00 ± 3.27E+00	2.58e+01 ± 2.86e+00
F_{10}	**0.00E+00** ± **0.00E+00**	2.29E−01 ± 1.32E−01	3.40E−01 ± 1.48E−01	1.32E−01 ± 6.23E−02	2.16E−02 ± 1.36E−02	1.49e−01 ± 1.59e−01
F_{11}	3.04E+00 ± 1.41E+00	2.36E+01 ± 8.76E+00	1.05E+02 ± 2.74E+01	**0.00E+00** ± **0.00E+00**	5.84E+01 ± 1.11E+01	3.98e−02 ± 1.99e−01
F_{12}	**2.91E+00** ± **1.38E+00**	5.64E+01 ± 1.51E+01	1.04E+02 ± 3.54E+01	6.44E+01 ± 1.48E+01	1.15E+02 ± 1.14E+01	8.36e+01 ± 1.74e+01
F_{13}	**2.78E+00** ± **1.45E+00**	1.23E+02 ± 2.19E+01	1.94E+02 ± 3.86E+01	1.15E+02 ± 2.24E+01	1.31E+02 ± 1.24E+01	1.42e+02 ± 2.19e+01
F_{14}	8.10E+02 ± 3.60E+02	7.04E+02 ± 2.38E+02	3.99E+03 ± 6.19E+02	1.55E+01 ± 6.13E+00	3.20E+03 ± 4.38E+02	**1.52e+00** ± **1.28e+00**
F_{15}	**7.65E+02** ± **2.95E+02**	3.42E+03 ± 5.16E+02	3.81E+03 ± 6.94E+02	3.55E+03 ± 3.04E+02	5.16E+03 ± 3.19E+02	3.48e+03 ± 4.60e+02
F_{16}	**4.40E−01** ± **9.26E−01**	8.48E−01 ± 2.20E−01	1.31E+00 ± 3.59E−01	1.03E+00 ± 2.01E−01	2.39E+00 ± 2.66E−01	1.10e+00 ± 3.01e−01
F_{17}	3.44E+01 ± 1.87E+00	5.26E+01 ± 7.11E+00	1.16E+02 ± 2.02E+01	3.09E+01 ± 1.23E−01	1.02E+02 ± 1.17E+01	**3.06e+01** ± **1.06e−01**
F_{18}	**6.23E+01** ± **4.56E+01**	6.81E+01 ± 9.68E+00	1.21E+02 ± 2.46E+01	9.01E+01 ± 8.95E+00	1.82E+02 ± 1.20E+01	9.78e+01 ± 1.90e+01
F_{19}	2.23E+00 ± 3.41E−01	3.12E+00 ± 9.83E−01	9.51E+00 ± 4.42E+00	1.71E+00 ± 4.68E−01	5.40E+00 ± 8.10E−01	**1.13e+00** ± **2.38e−01**
F_{20}	1.29E+01 ± 5.98E−01	1.20E+01 ± 9.26E−01	1.35E+01 ± 1.11E+00	1.11E+01 ± 7.60E−01	1.13E+01 ± 3.28E−01	**1.10e+01** ± **6.39e−01**
F_{21}	**1.92E+02** ± **2.72E+01**	3.11E+02 ± 7.92E+01	3.09E+02 ± 6.80E+01	3.18E+02 ± 7.53E+01	3.19E+02 ± 6.26E+01	2.90e+02 ± 4.91e+01
F_{22}	8.38E+02 ± 4.60E+02	8.59E+02 ± 3.10E+02	4.30E+03 ± 7.67E+02	8.41E+01 ± 3.90E+01	2.50E+03 ± 3.86E+02	**2.59e+01** ± **1.12e+01**
F_{23}	**6.67E+02** ± **2.90E+02**	3.57E+03 ± 5.90E+02	4.83E+03 ± 8.23E+02	4.18E+03 ± 5.62E+02	5.81E+03 ± 5.04E+02	4.09e+03 ± 3.81e+02
F_{24}	**1.62E+02** ± **3.00E+01**	2.48E+02 ± 8.11E+00	2.67E+02 ± 1.25E+01	2.51E+02 ± 1.43E+01	2.02E+02 ± 1.40E+00	2.35e+02 ± 1.16e+01
F_{25}	**2.20E+02** ± **1.11E+01**	2.49E+02 ± 7.82E+00	2.99E+02 ± 1.05E+01	2.75E+02 ± 9.76E+00	2.30E+02 ± 2.08E+01	2.81e+02 ± 1.44e+01
F_{26}	**1.58E+02** ± **3.00E+01**	2.95E+02 ± 7.06E+01	2.86E+02 ± 8.24E+01	2.60E+02 ± 7.62E+01	2.18E+02 ± 4.01E+01	2.00e+02 ± 7.07e−03
F_{27}	**4.69E+02** ± **7.38E+01**	7.76E+02 ± 7.11E+01	1.00E+03 ± 1.12E+02	9.10E+02 ± 1.62E+02	3.26E+02 ± 1.14E+01	8.85e+02 ± 1.10e+02
F_{28}	**2.69E+02** ± **7.35E+01**	4.01E+02 ± 3.48E+02	4.01E+02 ± 4.76E+02	3.33E+02 ± 2.32E+02	3.00E+02 ± 2.24E−02	3.00e+02 ± 1.62e−13

Table 4. Average ranking of six algorithms by the Friedman test for CEC-2013 functions at $D = 30$

Methods	NBIPOP-aCMA	BGBSA	SPSOABC	fk-PSO	PVADE	SPSO2011
Ranking	1.8	3.11	3.3	3.57	3.93	5.29

Table 5. Average ranking on 28 benchmark functions with varying stage control parameter

α	0.55	0.65	0.75	0.85	0.95
Ranking	3.09	2.71	2.61	3.05	3.54

As depicted in Table 3, NBIPOP-aCMA achieves the best performance since it is one of top three algorithm during CEC-2013. NBIPOP-aCMA, fk-PSO, SPSO2011, SPSOABC, PVADE and BGBSA perform better in 21, 3, 2, 4, 3, and 8 out of 28 functions respectively. According to the average ranking value of different algorithms by Friedman test, it can be concluded that NBIPOP-aCMA shows the best performance, and BGBSA offers the second overall performance, followed by SPSOABC, fk-PSO, PVADE and SPSO2011.

4.5 The Effect of the Parameter α

In BGBSA, the stage control parameter α is set to 0.75 which means the function evaluation times in early stage accounts for 75 percent of total evolutionary iteration times. It's necessary to choose a suitable value for α which determines the longitude of early evolutionary stage. The convergence speed of the proposed algorithm declines with bigger value of α. And BGBSA suffers from premature convergence if α is set to a smaller value.

To verify the effectiveness of α, five different values are tested and the Friedman test results are shown in Table 5. It can be seen clearly from Table 5, the algorithm with α equal to 0.75 offers the best performance, followed by 0.65, 0.55, 0.85 and 0.95. We infer that global search ability is gradually enhanced and local search ability can be ensured to bring high quality solutions as α is increased from 0.55 to 0.75. However, the performance of BGBSA suffers rapid decline when α is increased from 0.75 to 0.95. It can be analyzed that there is not enough time for best guided operator to exhibit exploitation ability. Therefore, $\alpha = 0.75$ is a reasonable choice for BGBSA.

5 Conclusions

In this work, we suggested a revised version of BSA for numerical optimization problems. The proposed algorithm combined the historical experience and the experience from the best individual obtained so far to enhance the convergence speed on the later stage of iteration. Our experimental results indicate that

BGBSA is effective and shows promising performance, especially when solving basic multimodal functions. For composition problems, BGBSA still leaves a large promotion space. In the future work, we plan to design BSA with adaptive experience guided operator.

Acknowledgments. This work was supported by the NSFC Joint Fund with Guangdong of China under Key Project U1201258, the National Natural Science Foundation of China under Grant No. 61573219, the Shandong Natural Science Funds for Distinguished Young Scholar under Grant No. JQ201316, the Natural Science Foundation of Fujian Province of China under Grant No. 2016J01280 and the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions.

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems. MIT press, Cambridge (1992)
2. Storn, R., Price, K.: Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. ICSI, Berkeley (1995)
3. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B Cybern. **26**, 29–41 (1996)
4. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: 6th International Symposium on Micro Machine and Human Science, pp. 39–43. IEEE Press, New York (1995)
5. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report, Erciyes University (2005)
6. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**, 702–713 (2008)
7. Civicioglu, P.: Backtracking search optimization algorithm for numerical optimization problems. Appl. Math. Comput. **219**, 8121–8144 (2013)
8. Agarwal, S.K., Shah, S., Kumar, R.: Classification of mental tasks from eeg data using backtracking search optimization based neural classifier. Neurocomputing **166**, 397–403 (2015)
9. Yang, D.D., Ma, H.G., Xu, D.H., Zhang, B.H.: Fault measurement for siso system using the chaotic excitation. J. Franklin Inst. **352**, 3267–3284 (2015)
10. Mallick, S., Kar, R., Mandal, D., Ghoshal, S.: CMOS analogue amplifier circuits optimisation using hybrid backtracking search algorithm with differential evolution. J. Exp. Theor. Artif. Intell. **28**(4), 719–749 (2016)
11. Xu, Q., Guo, L., Wang, N., Li, X.: Opposition-based backtracking search algorithm for numerical optimization problems. In: He, X., Gao, X., Zhang, Y., Zhou, Z.-H., Liu, Z.-Y., Fu, B., Hu, F., Zhang, Z. (eds.) 5th International Conference on Intelligence Science and Big Data Engineering. LNCS, vol. 9243, pp. 223–234. Springer, Switzerland (2015)
12. Duan, H., Luo, Q.: Adaptive backtracking search algorithm for induction magnetometer optimization. IEEE Trans. Mag. **50**(12), 1–6 (2014)
13. Askarzadeh, A., Coelho, L.D.S.: A backtracking search algorithm combined with Burger's chaotic map for parameter estimation of PEMFC electrochemical model. Int. J. Hydrogen Energy **39**, 11165–11174 (2014)

14. Wang, L., Zhong, Y., Yin, Y., et al.: A hybrid backtracking search optimization algorithm with differential evolution. Math. Probl. Eng. **2015**, 1 (2015)
15. Zhao, W., Wang, L., Yin, Y., Wang, B., Wei, Y., Yin, Y.: An improved backtracking search algorithm for constrained optimization problems. In: Buchmann, R., Kifor, C.V., Yu, J. (eds.) KSEM 2014. LNCS (LNAI), vol. 8793, pp. 222–233. Springer, Heidelberg (2014). doi:10.1007/978-3-319-12096-6_20
16. Liang, J., Qu, B., Suganthan, P., Hernndez-Daz, A.G.: Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Technical report, Singapore (2013)
17. Loshchilov, I.: CMA-ES with restarts for solving CEC 2013 benchmark problems. In: 2013 IEEE Congress on Evolutionary Computation, pp. 369–376. IEEE Press (2013)
18. Nepomuceno, F.V., Engelbrecht, A.P.: A self-adaptive heterogeneous PSO for real-parameter optimization. In: 2013 IEEE Congress on Evolutionary Computation, pp. 361–368. IEEE Press (2013)
19. Zambrano-Bigiarini, M., Clerc, M., Rojas, R.: Standard particle swarm optimisation 2011 at CEC-2013: a baseline for future PSO improvements. In: 2013 IEEE Congress on Evolutionary Computation, pp. 2337–2344. IEEE Press (2013)
20. El-Abd, M.: Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks. In: 2013 IEEE Congress on Evolutionary Computation, pp. 2215–2220. IEEE Press (2013)
21. Coelho, L.D.S., Ayala, V.H., Freire, R.Z.: Population's variance-based adaptive differential evolution for real parameter optimization. In: 2013 IEEE Congress on Evolutionary Computation, pp. 1672–1677. IEEE Press (2013)
22. Gong, W.Y., Cai, Z.H.: Differential evolution with rankingbased mutation operators. IEEE Trans. Cybern. **43**, 2066–2081 (2013)