Mathematical Biosciences and Engineering

MBE, 19(8): 8537–8553.
DOI: 10.3934/mbe.2022396
Received: 13 February 2022
Revised: 22 May 2022
Accepted: 29 May 2022
Published: 10 June 2022

http://www.aimspress.com/journal/MBE

Research article

An adaptive differential evolution algorithm with elite gaussian mutation and bare-bones strategy

Lingyu Wu, Zixu Li, Wanzhen Ge and Xinchao Zhao*

School of Science, Beijing University of Posts and Telecommunications, Beijing, China

* **Correspondence:** Email: xczhao@126.com.

Abstract: Both differential evolution algorithm (DE) and Bare-bones algorithm (BB) are simple and efficient, but their performance in dealing with complex multimodal problems still has room for improvement. DE algorithm has great advantages in global search and BB algorithm has great advantages in local search. Therefore, how to combine these two algorithms' advantages remains open for further research. An adaptive differential evolution algorithm based on elite Gaussian mutation strategy and bare-bones operations (EGBDE) is proposed in this paper. Some elite individuals are selected and then the mean and the variance of the bare-bones operation are adjusted with the information from the selected elite individuals. This new mutation strategy enhances the global search ability and search accuracy of differential evolution with parameters free. It also helps algorithm get a better search direction and effectively balance the exploration and exploitation. An adaptive adjustment factor is adopted to dynamically balance between differential mutation strategy and the elite Gaussian mutation. Twenty test functions are chosen to verify the performance of EGBDE algorithm. The results show that EGBDE has excellent performance when comparing with other competitors.

Keywords: differential evolution (DE); bare-bones (BB); gaussian mutation; global optimization

1. Introduction

Evolutionary algorithms (EAs) are random search algorithms inspired by the evolution of natural organisms [1]. In the past decades, EAs have shown their excellent performance and have been widely used in various problems, such as continuous optimization, discrete optimization,

constrained optimization, and multi-objective optimization. In addition, swarm intelligence techniques which is inspired from the behavior of social insects or animals, nature inspired algorithms rise in recent years. In swarm intelligence, every individual has its own intelligence and behavior, but the integration of the individuals gives more power to solve complex problems [2]. The most common swarm intelligence algorithms include ant colony algorithm (AG), particle swarm optimization (PSO), and so on.

Differential evolution (DE) is one of evolutionary algorithms. It is a population-based algorithm and was proposed by Storn and Price [3] on the basis of evolutionary ideas such as genetic algorithm in 1997. Since DE was proposed, it has been widely used in various fields because of its effectiveness, simplicity, robustness, and a lower number of control parameters [4]. It has been used in image recognition [5], vehicle scheduling [6] and other practical problems [7–10]. However, DE still has some problems, such as premature convergence [11], slow convergence [12], and many improved versions of DE [13] are proposed. Some DE variants for adaptive adjustment of control parameters and mutation strategies have been proposed, such as JADE [14], SaDE [15], ODE [16,17]. The characteristic of this kind of algorithm is to bring the adjustment of control parameters and mutation strategies into the evolutionary process. It records the experience of generating high-quality solutions each time, so as to provide beneficial reference for the next iteration [18,19].

The Bare-bones idea was firstly applied in particle swarm optimization algorithm by Kennedy in 2003, and BBPSO was proposed [20]. Its high search efficiency and accuracy, and relatively simple parameter setting have been successfully applied to many fields, such as constrained optimization problems [21], data clustering [22] and PV systems [23]. In the improvement of differential evolution algorithm, in order to reduce the sensitivity of control parameters to algorithm performance, Omran et al. [24] combined BBPSO algorithm and DE algorithm to obtain a skeleton differential evolution (BBDE) algorithm in 2008. The algorithm uses the weighted average of individual with global optimal position and individual with historical optimal position, which reduces the influence of control parameters, such as scaling factor, on the performance of the algorithm. But it is easy to fall into local optimal when solving multimodal optimization problems. Therefore, Wang et al. [25] proposed a GBDE algorithm based on BBDE algorithm, which implements Gaussian mutation strategy. The algorithm uses individual information to maintain population diversity, but its mining ability is poor. To improve the performance of the algorithm, Peng et al. [26] proposed tBBDE algorithm respectively on the basis of GBDE algorithm. The tBBDE algorithm uses the Gaussian mutation strategy based on three random individuals to search the solution space more comprehensively. But its large randomness leads to the blind search of the algorithm, which affects the convergence speed. Besides tBBDE, Wang et al. [27] proposed MGBDE algorithm which uses DE/best/1 mutation strategy and executes two mutation strategies with fixed probability. However, the MGBDE algorithm does not make adaptive adjustment to the mutation strategy according to the different evolution stages, so it is easy to fall into local optimal.

The above modified algorithms are based on Gaussian sampling. Gaussian sampling is an adaptive sampling process, which can adjust the size of the sampling area with the progress of sampling. Some research think that the sampling process of Gaussian sampling is very similar to the iterative process of DE/best/1 or DE/current-to-best/1. However, it should note that the adaptive adjustment of Gaussian sampling can work in a larger area, but the exploration area is still too narrow. So, this paper's motivation is to improve the Gaussian mutation and propose an adaptive bare-bones differential algorithm to make differential evolution algorithm performance better.

This paper proposes a new elite Gaussian mutation strategy. The new strategy adds more elite individual information to the bare-bones operation which can make the population search area dynamically adjust in each period of population evolution and make the search area more reasonable. So, the strategy not only retains the advantages of the original Gaussian mutation strategy (less parameters), but also compensates for the loss of the search range of the mutation strategy. Experiments and analysis are presented for the parameter selection in the new strategy. Besides, this paper combines both mutation strategies and adjusts them dynamically. The algorithm realizes the dynamic balance between exploration and exploitation. And in the process of population evolution, premature convergence and search stagnation are effectively reduced.

The rest of this paper is organized as follows. The second section is the introduction of DE algorithm. The third section presents the core improving strategies of EGBDE algorithm. The fourth section is for the simulation comparison and experimental analysis. The last Section concludes this paper.

2. Differential evolution

DE algorithm is a population-based stochastic search algorithm [28]. It is distinguished from other evolutionary algorithms (EAs) by its unique trial vector generation strategy (mutation operator and crossover operator). The properties of its mutation operator and crossover operator allow DE to control the exploration magnitude and direction to find promising solutions [29].

Firstly, the algorithm needs to generate an initial population containing N D-dimensional vectors $X_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$, and then carries out three operations: differential mutation, crossover and selection until the pre-defined termination condition is met.

2.1. Differential mutation

Differential mutation is the core generation step of DE. In each iteration, the algorithm performs differential mutation operation in sequence according to the order of individuals in the population. Several individuals are randomly selected in the population to perform the differential mutation and generate a differential vector $V_{i,G}$ each time. The mutation operation has quite a significant impact on differential evolution algorithm performance. Therefore, a great number of researchers have conducted research on mutation operators. Since the differential evolution algorithm first was proposed, a significant number of mutation operators have been proposed, and the following are classic mutation operators.

① DE/rand/1

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \tag{1}$$

② DE/rand/2

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) + F \cdot (X_{r4,G} - X_{r5,G}) \tag{2}$$

③ DE/best/1

$$V_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G}) \tag{3}$$

④ DE/current/1

$$V_{i,G} = X_{i,G} + F \cdot (X_{r1,G} - X_{r2,G}) \tag{4}$$

⑤ DE/current-to-best/1

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r1,G} - X_{r2,G}) \tag{5}$$

In Eqs (1)–(5), G is the iteration generation, $X_{best,G}$ is the best individual of the current population. $X_{i,G}$ is the current individual. The indexes $r1, r2, r3, r4, r5$ are random integers in $[1, Np]$ and they are not equal. The parameter F is the scaling factor in $[0,2]$, usually 0.5. $i \in \{1,2,\cdots,Np\}$.

2.2. Crossover operation

Once the mutation vector has been obtained, the crossover operator is applied to the parent vector and mutation vector in order to obtain the trial vector. There are many common crossover operations in DE algorithm. Binomial crossover is widely used in DE algorithm. The formula for binomial crossover is as follows.

$$U_{i,j,G} = \begin{cases} V_{i,j,G}, rand(0,1) \le CR \ or \ j = j_{rand} \\ X_{i,j,G}, otherwise \end{cases} \tag{6}$$

where j_{rand} is a predefined random integer in $[1, D]$, the parameter CR is the crossover probability, $rand(0,1)$ is a random number on $U(0,1)$ and $U(0,1)$ represents the uniform distribution on the interval $[0,1]$.

2.3. Selection operation

In DE algorithm, greedy selection strategy is generally used to select the competitive individual according to the fitness value of $X_{i,G}$ and mutation crossover vector $U_{i,G}$. For the minimization problem, the formula of the selection operation is as follows.

$$X_{i,G+1} = \begin{cases} U_{i,G}, f(U_{i,G}) < f(X_{i,G}) \\ X_{i,G}, \text{otherwise} \end{cases} \tag{7}$$

3. Adaptive differential evolution algorithm based on elite gaussian mutation strategy

3.1. Research motivation

Traditional DE algorithms often use mutation strategy (1) and it has strong global search ability, but its randomness is so high that the convergence speed of the algorithm is slow [30,31]. Strategy (3) has fast convergence speed, but the small search range leads to the poor effect of the algorithm in dealing with complex optimization problems. Many improved DE variants hope to enhance the algorithm adaptability and realize the balance between exploration and exploitation [32]. Bare-bones algorithm is also a common optimization algorithm. But compared with DE algorithm, its poor ability to jump out of the local optimum makes the bare-bones algorithm's performance poor. To search for better balance between the global exploration and the local exploitation, this paper proposes an adaptive bare-bones differential evolution algorithm based on improved Gaussian mutation strategy. The new hybrid mutation strategy uses more elite individual information and avoids the problem that elite learning is easy to fall into local optimum. In addition, the new strategy

has fewer parameters and can be used in other algorithms.

3.2. Elite gaussian mutation strategy based on bare-bones

The improved Gaussian mutation strategy proposed in this paper is different from the traditional bare-bones Gaussian mutation strategy. The traditional bare-bones Gaussian strategy is as follows.

$$V_{i,G} = N\left(\frac{X_{best,G}+X_{i,G}}{2}, \left|X_{best,G} - X_{i,G}\right|\right) \tag{8}$$

where $X_{best,G}$ represents the individual with the best fitness, $X_{i,G}$ represents the current individual, and $N(\mu,\sigma)$ represents normal distribution with mean μ and variance σ.

In order to improve the general performance of DE algorithm, this paper integrates the idea of bare-bones strategy into DE algorithm, so that the advantages and disadvantages of differential mutation strategy and bare-bones generation strategy complement each other. The main differential mutation strategy used in this paper is DE/rand/1, and the improved Gaussian mutation strategy with the heuristic information of elite individuals is as follows.

$$V_{i,G}^{B} = N\left(X_{\partial}, \frac{\sum_{i=1}^{Ls}\|X_{ri,G}-X_{\partial}\|}{Ls}\right) \tag{9}$$

$$X_{\partial} = \frac{X_{r1,G}+X_{r2,G}+\cdots+X_{rLs,G}}{Ls} \tag{10}$$

where indexes $r1, r2, \cdots, rLs$ are integers from $\{1,2,\cdots,Np/k\}$, where k is a positive number, Ls is set to 3 in this paper, G is the current iteration number, and $N(\mu,\sigma)$ represents normal distribution with mean μ and variance σ.

3.3. Discussion on the elite population size

Table 1. Average ranking comparison among different parameters.

	$k=2$	$k=10/3$	$k=4$	$k=5$	$k=10$
Ranking	3.57	3.23	2.87	2.33	3

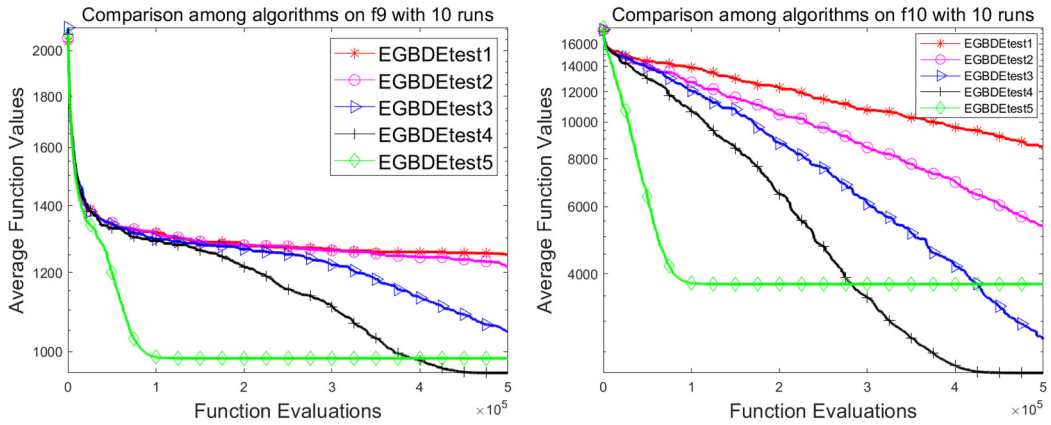

Figure 1. Experimental comparison results with $k=2$, $k=\frac{10}{3}$, $k=4$, $k=5$, $k=10$.

Parameter k is used to control the elite population size. Five simulation experiments are conducted on five parameters k = 2，k = 10/3，k = 4，k = 5 and k = 10 in which Np = 100. The test function is same as Section 4.1 and the algorithm will run 10 times in a simulation experiment. After 5 simulation experiments, each parameter will get a ranking on each test function. The results will be each parameter's average ranking on 20 test function and these results are shown in Table 1. Some of the results on one algorithm are shown in Fig. 1 where EGBDEtest1 shows the result of k = 2, EGBDEtest2 shows the result of k = 10/3, EGBDEtest3 shows the result of k = 4, EGBDEtest4 shows the result of k = 5, EGBDEtest5 shows the result of k = 10. Observed from the simulation results, it finds that the situation when k = 5, the optimization effect is the best. Based on the situation, when the selected sample range is too large, the randomness is too strong, which affects the convergence speed and convergence accuracy of the algorithm. When the selected sample range is too small, the search range is relatively limited and the possibility of being trapped in the local area is greatly improved. Therefore, the value of key parameter k will be set to 5 in the following experiment and analysis of this paper.

3.4. Comparison analysis between both gaussian mutations

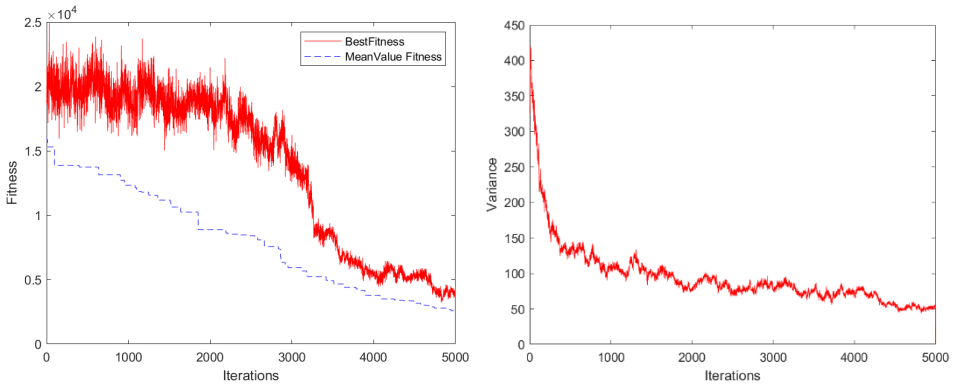

Figure 2. The changing of mean value fitness and variance of the improved Gaussian mutation.

Different from the improved Gaussian mutation proposed in this paper, the traditional Gaussian mutation takes the average between the present optimal vector and the current vector as the mean. The new individual will be obtained with the absolute difference between the optimal vector and the current vector as the variance. According to the new individuals' generation process, all the new individuals will locate around the present optimal solution although the neighbor size is different. The improved Gaussian mutation strategy takes three individuals from the first best quarter of the population randomly and computes their geometric center of gravity as mean. The mean distance from three individuals to the center of gravity will be used as the variance to generate new individuals. Compared with the traditional Gaussian mutation strategy, this strategy utilizes more elite individual information and expands the scope of local search. The new strategy also retains the characteristics of the original Gaussian algorithm. In the early stage of evolution, the distance among the selected three individuals is relatively larger. Most of the new individuals generated by the improved Gaussian mutation are concentrated near X_∂. The new strategy has even more uniformly distribution and has strong exploration ability in this stage. With the progress of evolution, the distance among the selected three individuals is decreasing. The strategy will gradually turn from

exploration to exploitation, and its local search ability is continuously enhanced.

Figure 2 selects the first test function's result and indicates the change of mean and variance in the improved Gaussian mutation strategy. It can track the change trend of the optimal value of the population to a certain extent. The variance is gradually shrinking and retaining the characteristics of the original Gaussian variation strategy.

3.5. Hybrid mutation strategy

The DE/rand/1 variation strategy has outstanding performance in global search and maintaining population diversity, but the convergence speed is slower than the improved elite Gaussian mutation. On the other hand, the improved elite Gaussian mutation utilizes much more heuristic information from the elite individuals, so it can use the information of the elite individuals to help population converge. In order to take full use of the beneficial features from both mutation strategies and to realize the complementarity of the pros and cons in different evolutionary stages, this paper combines both mutation strategies into a hybrid mutation strategy. The improved elite Gaussian mutation will be used with a high probability in the early stage of the algorithm, so that it can help the algorithm find believable exploration direction. An adaptive parameter is used to adjust the rate of both strategies with the increase of generation. The hybrid mutation is shown as follows.

$$V_{i,G} = \begin{cases} V_{i,G}^{A}, rand(0,1) < W_l \\ V_{i,G}^{B}, \text{otherwise} \end{cases} \tag{11}$$

where $V_{i,G}^{A}$ is DE/rand/1 variation strategy, $rand(0,1)$ is a random number on $U(0,1)$.

The equation for the adaptive parameter value W_l is as follows.

$$W_l = w_l + (1 - w_l) \cdot \frac{FES}{maxFES} \tag{12}$$

where w_l represents the initial predefined selection probability of the first mutation, FES represents the current function evaluation number, and $maxFES$ represents the maximum function evaluation number.

In this paper, the using proportion of both strategies is dynamically adjusted. At the beginning of the proposed strategy, the improved elite Gaussian mutation strategy has a larger probability to be chosen. Because it can provide better search direction, the strategy assists the algorithm to carry out effective exploration to a great extent. With the process of the algorithm, the improved elite Gaussian mutation strategy has a smaller and smaller probability [33,34]. Its local search ability cannot help the algorithm exploit locally, so the improved elite Gaussian mutation strategy has a low probability at the end of the algorithm. The algorithm procedure is shown in Algorithm 1.

Algorithm 1: Improved Gaussian mixture mutation strategy

Input: current individual $X_{i,G}$

Output: experimental individual $V_{i,G}$

$W_l = w_l + (1\text{-}w_l) \times \frac{FES}{maxFES}$;
if $rand(0,1) < W_l$ **then**
  $V_{i,G} = x_{r1,G} + F \cdot (x_{r2,G}\text{-}x_{r3,G})$;
else
  $V_{i,G} = N(X_\partial, \frac{\sum_1^3 \|X_{ri,G} - X_\partial\|}{3})$;
end
return experimental individual $V_{i,G}$

3.6. Procedure of EGBDE

Algorithm 2: EGBDE

Input: Objective function:f Maximal evaluation times:$maxFES$
Lower bounds:X_{min} Upper bounds:X_{max} Crossover rate:CR Scaling factor:F
Initial selection probability of the first mutation:w_l

Output: the best individual value

Initialize a population $P_0 = \{X_{1,0}, X_{2,0}, ..., X_{Np,0}\}$ within the lower bounds X_{min} and upper bounds X_{max};
$FES = Np$;
while $FES <= maxFES$ **do**
  $W_l = w_l + (1 - w_l) \times \frac{FES}{maxFES}$;
  for $i = 1$ *to* Np **do**
    if $rand(0,1) < W_l$ **then**
      $V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G}\text{-}X_{r3,G})$;
    else
      choose elite Gaussian mutation;
      $V_{i,G} \sim N(X_\partial, \frac{\sum_1^{Ls} \|X_{ri,G} - X_\partial\|}{Ls})$;
    Using crossover operation to generate a trial vector $U_{i,G}$;
    if $f(U_{i,G}) < f(X_{i,G})$ **then**
      $X_{i,G+1} = U_{i,G}$;
    else
      $X_{i,G+1} = X_{i,G}$;
  $FES = FES + Np$
end

EGBDE algorithm has similar framework with classic DE algorithm, but it presents more choices in mutation strategy. As presented in Algorithm 2, the proposed EGBDE starts with the random initialization population with size N. In each generation, every individual generates a random number in $U(0,1)$ and selects mutation strategy according to the random number. Then, every individual will experience crossover operation and elite selection operation. When the

termination is met, the algorithm will stop.

4. Experimental comparison and verification analysis

4.1. Test functions and parameter setting

In order to fully verify the performance and effectiveness of EGBDE algorithm, this paper uses 20 benchmark functions in CEC2014 to compare the performance with the well-known DE variants ODE, JADE, SaDE and the improved Gaussian bare-bones algorithm without differential mutation operation. In the test function, unimodal function includes f3, simple multimodal functions include f4, f5, f6, f7, f9, f11, f12, f15 and f16, hybrid functions include f18, f19, f20 and f21, and composition functions include f24, f25, f26, f27, f28 and f30. Each function is for minimization which are renamed as F1~F20.

In order to ensure the fairness of comparison, the common parameter settings of all algorithms are the same, as shown in Table 2. Other parameters take the recommended values of each algorithm, and each algorithm runs for 30 times independently. All experiments were run on a computer with Intel (R) core (TM) i7-9750h CPU @ 2.6 GHz, 16 GB memory and 64-bit Windows 10 (R) operating system.

Table 2. Parameter setting.

Population Size	*CR*	*F*	Max Iterations
100	0.9	0.5	500

4.2. Comparison analysis of convergence accuracy

Table 3 shows the average best fitness and standard deviation of the final solution at each independent run found by each algorithm in 30 runs. The best results of each test function are marked in bold. The evaluation criterion is that the smaller the average value is, the better for the corresponding function. When the average fitness is the same, the one with the smaller standard deviation is better. Wilcoxon signed rank test where the significance level of 0.05 is carried out. From the results in Table 4 of Wilcoxon signed rank test [35], there is no significant performance difference between EGBDE and JADE. Besides, EGBDE is significantly better than SaDE, ODE and BB.

Table 3. The average values and standard deviation of the best results.

Function	EGBDE		JADE		SaDE		ODE		BB	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	3.558E+02	1.164E+02	6.461E+03	2.839E+03	1.283E+03	8.155E+02	**3.002E+02**	2.270E-01	1.062E+05	1.681E+04
F2	4.941E+02	3.788E+00	**4.231E+02**	3.965E+01	5.032E+02	3.646E+01	9.869E+02	2.280E+02	1.608E+03	2.707E+02
F3	**5.201E+02**	2.016E-01	5.204E+02	3.728E-02	5.208E+02	3.921E-02	5.211E+02	1.128E-01	5.205E+02	1.510E-01
F4	**6.020E+02**	1.824E+00	6.178E+02	6.833E+00	6.111E+02	2.731E+00	6.272E+02	5.105E+00	6.539E+02	3.297E+00
F5	**7.000E+02**	2.160E-11	7.000E+02	5.083E-03	7.000E+02	1.135E-02	7.435E+02	2.724E+01	8.192E+02	2.220E+01
F6	**9.519E+02**	1.064E+01	9.684E+02	1.037E+01	9.886E+02	3.097E+01	1.057E+03	9.859E+01	1.303E+03	4.360E+01
F7	6.899E+03	2.467E+03	**5.808E+03**	3.055E+02	9.641E+03	3.100E+02	9.593E+03	2.624E+03	1.048E+04	8.431E+02
F8	**1.200E+03**	1.458E-01	1.200E+03	4.039E-02	1.201E+03	1.170E-01	1.202E+03	9.711E-01	1.201E+03	3.066E-01
F9	1.511E+03	7.756E+00	**1.510E+03**	9.975E-01	1.522E+03	4.664E+00	2.615E+03	2.628E+03	1.578E+04	1.287E+04
F10	1.620E+03	1.008E+00	**1.618E+03**	4.543E-01	1.620E+03	2.068E-01	1.621E+03	7.042E-01	1.622E+03	3.851E-01
F11	**1.939E+03**	4.033E+01	1.989E+03	4.337E+01	2.235E+03	3.034E+02	1.251E+07	2.625E+07	1.794E+07	1.565E+07
F12	**1.911E+03**	1.269E+00	1.915E+03	3.321E+00	1.924E+03	1.501E+01	1.974E+03	2.434E+01	2.007E+03	2.292E+01
F13	**2.085E+03**	3.016E+01	1.982E+04	7.694E+03	2.199E+03	1.336E+02	2.110E+03	1.013E+01	6.700E+04	2.448E+04
F14	5.637E+03	2.490E+03	**3.430E+03**	2.822E+02	1.412E+04	1.340E+04	1.022E+06	3.324E+06	7.656E+06	4.468E+06
F15	2.672E+03	1.738E+00	2.674E+03	1.894E+00	2.673E+03	2.488E+00	**2.600E+03**	2.146E-02	2.750E+03	8.343E+00
F16	2.706E+03	4.894E-01	2.715E+03	5.913E+00	**2.703E+03**	6.513E+00	2.709E+03	7.807E+00	2.768E+03	1.473E+01
F17	**2.700E+03**	5.651E-02	2.707E+03	2.485E+01	2.773E+03	4.410E+01	2.724E+03	4.235E+01	2.701E+03	3.182E-01
F18	**3.083E+03**	3.680E+01	3.121E+03	7.072E+01	3.312E+03	6.314E+01	3.721E+03	1.992E+02	4.328E+03	1.016E+02
F19	**3.893E+03**	2.912E+01	3.940E+03	5.664E+01	4.118E+03	9.251E+01	5.175E+03	5.576E+02	6.055E+03	6.253E+02
F20	**1.140E+04**	3.512E+02	1.266E+04	6.983E+02	1.389E+04	1.719E+03	5.754E+04	3.546E+04	2.274E+05	8.999E+04

Table 4. Statistical comparison results of Wilcoxon signed rank test.

	EGBDE	JADE	SaDE	ODE	BB
p-value	-	0.079	< 0.0001	0.002	< 0.001

Observed from the numerical results of Table 3, the proposed EGBDE algorithm achieves the best performance in general. For unimodal problems, the performance of EGBDE is only second to ODE. Because ODE adopts the contraposition search method for exploration and development, it has a larger search range on unimodal problems than other algorithms and can search a larger area within a limited number of iterations, so it achieves better performance in solving unimodal problems.

For the simple multimodal problem, the search area of EGBDE can be changed with the update of individuals and dynamically adjust the balance between exploration and exploitation. Therefore, very excellent results have been obtained in test functions F3, F4, F5, F6 and F8. It is secondly only to JADE in test functions F2, F7, F9 and F10. The possible reason is the ensemble strategy of JADE for the diverse search trajectories.

For hybrid functions, EGBDE achieves excellent results on test functions F11, F12 and F13, and is worse than JADE on F14. As shown in the Figure 3, at the initial search stage of algorithm, the function image can be approximated as a unimodal function. The requirement for local search ability is not high. JADE uses the improved DE/current-to-best/1 mutation strategy and adaptive parameters. So, JADE can find the optimal evolving direction in a short time and has better performance in function F14. Besides, the results show that EGBDE has good adaptability and obvious improvement in the treatment of hybrid problems.

For composite functions, EGBDE is obviously superior to other competitors in test functions F17, F18, f19 and F20. It ranks only second to ODE algorithm for test function F15 and SaDE algorithm for test function F16. The results show the competitive advantages of EGBDE in dealing with complex problems.

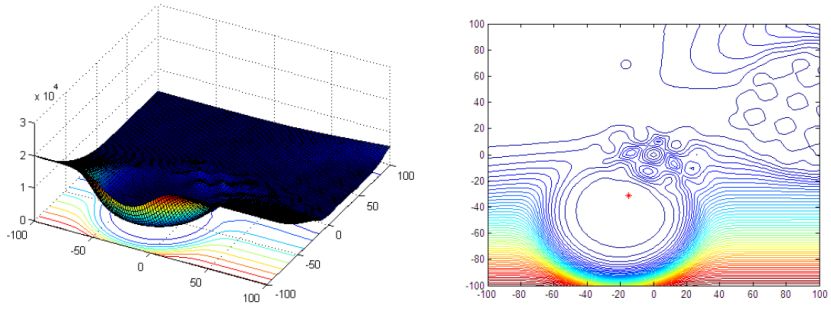

Figure 3. 3-D map and Contour map for 2-D function of F14 [36].

To further statistically obtain the pros and cons of each algorithm, Fried-man test is used to rank the comprehensive optimization performance of the algorithms. In the experiment, the smaller the rank mean, the better the performance of the algorithm. The statistical results are shown in Table 5.

Table 5. Statistical comparison of Friedman test.

	EGBDE	JADE	SaDE	ODE	BB
Rank	1.40	2.25	3.10	3.65	4.60

4.3. Comprehensive evolving behavior comparison

Besides the final convergence accuracy comparison, the comprehensive evolving behaviors comparison of the competing algorithms is also presented in this paper. Due to the evolving similarities for some functions, the representative evolving behavior comparison curves of six functions F3, F4, F6, F8, F18 and F19 are plotted and shown in Figure 4. This figure is based on 10 runs.

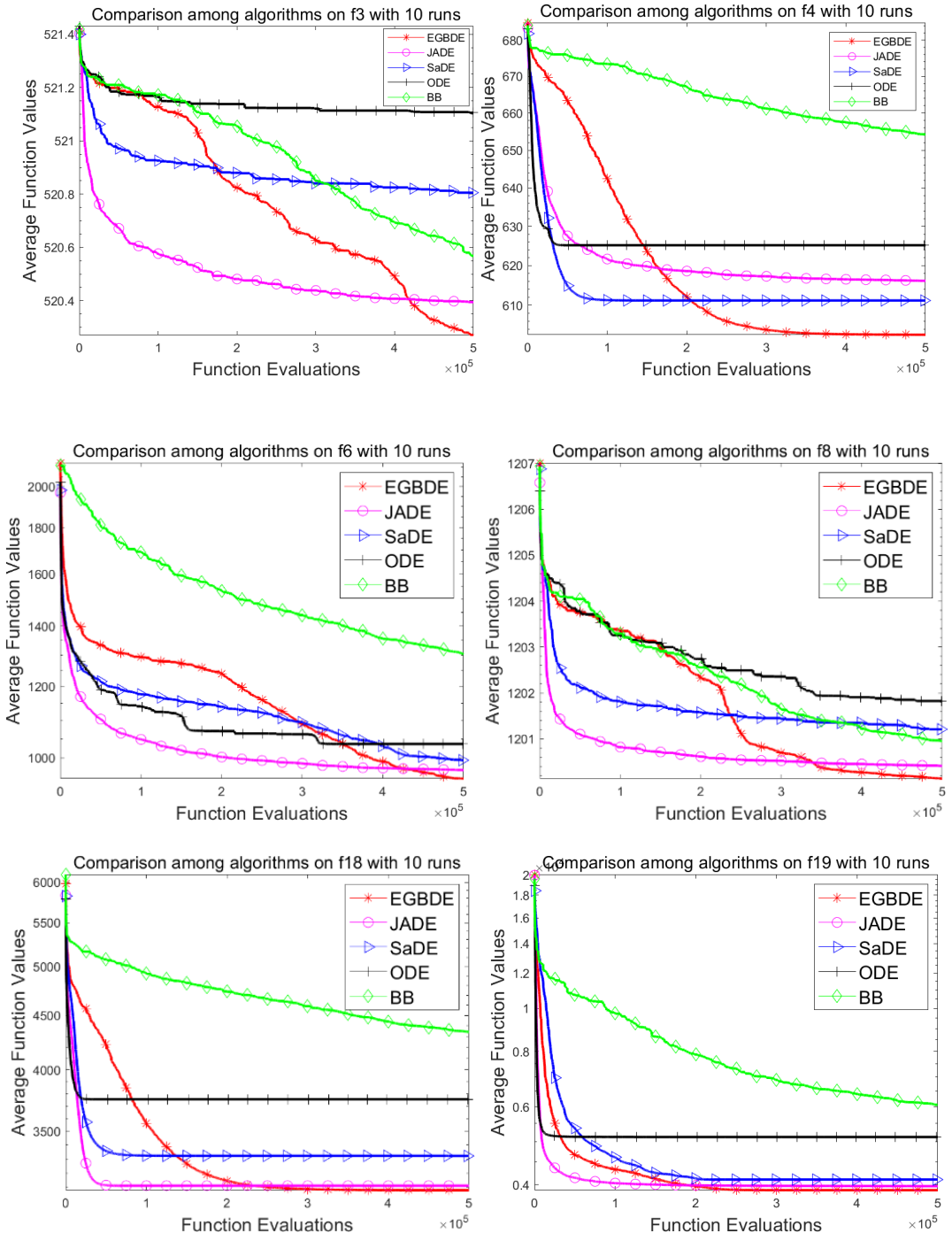

Figure 4. Evolving behavior comparison among algorithms.

Observed from Figure 4, the proposed EGBDE does not illustrate all time superiority to the other algorithms. EGBDE usually converges slower than competitors and surpassed them at the middle to later stage during the optimizing process. The inherent reason is that the proposed Gaussian mutation strategy only utilizes the elite individuals as the mean and generates a new individual randomly around the mean. It provides search direction for the algorithm in global exploration. However, the end stage of evolution, the probability of choosing Gaussian mutation approaches to zero and its variance is also usually too small to help the algorithm jump from local optimum.

In addition to the above numerical and convergence comparison, the boxplots of the same six representative functions are illustrated in Figure 5. It indicates that the first quantile, the middle value and the third quantile of EGBDE are better than other competitors. The solutions of EGBDE are relatively concentrated which shows the promising and robust property of EGBDE.

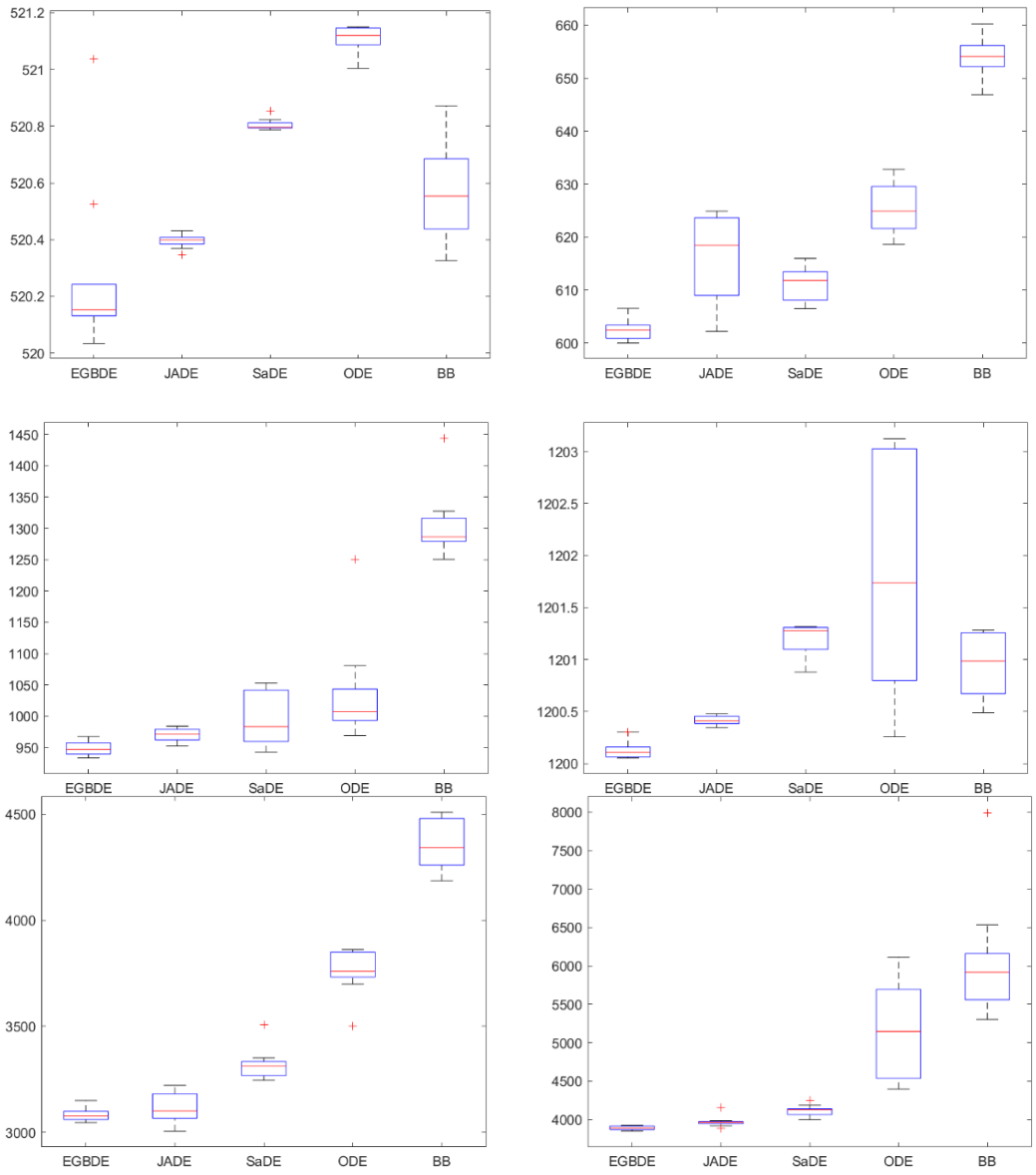

Figure 5. Boxplots of illustration comparison among algorithms.

Based on the above experimental comparison and analysis, it can conclude that the proposed elite Gaussian mutation strategy greatly enhances the performance of the algorithm. It makes the population evolve in a possible promising direction in a limited number of iterations and achieves a balance between exploration and exploitation.

5. Conclusions

An elite Gaussian mutation strategy is proposed in this paper inspired by bare-bones operation and differential mutation. Some elite individuals are firstly selected and then the mean and the variance of the bare-bones operation are calculated from the randomly selected elite individuals. This new mutation strategy enhances the global search ability and search accuracy of the canonical differential evolution algorithm. In addition, the improved Gaussian mutation strategy not only retains the characteristics of the original Gaussian mutation strategy, but also carries out global search in a larger space to reduce the possibility of falling into local extremum. The new algorithm randomly selects from both strategies which is adaptively adjusted by a proportion factor, and the proportion changes with the evolution of algorithm, aiming at the dynamic adaptive adjustment of the search range.

This paper also discusses the mean and variance of the improved elite Gaussian mutation strategy. Different mean and variance show different performance on different test functions. A set of mean and variance is selected with the best performance to improve the performance of the algorithm. From the simulation comparison, EGBDE performs very promising and has a good ability to balance between exploitation and exploration.

Just as there is no free lunch, the improved Gaussian mutation strategy still has room be improved. The improved Gaussian mutation strategy uses elite solutions frequently, so it will be a little difficult for algorithm to jump out from local optimum. In the future EGBDE or its variant is possible to be applied to multi-objective optimization and some practical problems like portfolio optimization problems [37].

Acknowledgements

This work is supported by Beijing Natural Science Foundation (1202020) and National Natural Science Foundation of China (61973042).

Conflict of interest

The authors declared no potential conflicts of interest with respect to the research, authorship and/or publication of this paper.

References

1. A. E. Eiben, J. E. Smith, *Introduction to evolutionary computing*, Springer, (2003), 15–30. http://dx.doi.org/10.1007/978-3-662-05094-1

2. A. W. Mohamed, A. A. Hadi, A. K. Mohamed, Gaining-sharing knowledge-based algorithm for solving optimization problems: a novel nature-inspired algorithm, *Int. J. Mach. Learn. Cybern.*, **11** (2020), 1501–1529. http://dx.doi.org/10.1007/s13042-019-01053-x
3. R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.*, **11** (1997), 341–359. http://dx.doi.org/10.1023/a:1008202821328
4. A. Qin, V. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Trans. Evol. Comput.*, **13** (2008), 398–417. http://dx.doi.org/10.1109/TEVC.2008.927706
5. A. K. Bhandari, A novel beta differential evolution algorithm-based fast multilevel thresholding for color image segmentation, *Neural Comput. Appl.*, **32** (2020), 4583–4613. http://dx.doi.org/10.1007/s00521-018-3771-z
6. W. Liu, Y. Gong, W. Chen, Z. Liu, H. Wang, J. Zhang, Coordinated charging scheduling of electric vehicles: a mixed-variable differential evolution approach, *IEEE Trans. Intell. Transp. Syst.*, **21** (2019), 5094–5109. http://dx.doi.org/10.1109/TITS.2019.2948596
7. E. N. Dragoi, V. Dafinescu, Parameter control and hybridization techniques in differential evolution: a survey, *Artif. Intell. Rev.*, **45** (2016), 447–470. http://dx.doi.org/10.1007/s10462-015-9452-8
8. Y. Kharchouf, R. Herbazi, A. Chahboun, Parameter's extraction of solar photovoltaic models using an improved differential evolution algorithm, *Energy Conv. Manag.*, **251** (2022), 114972. http://dx.doi.org/10.1016/j.enconman.2021.114972
9. D. Liu, Z. Hu, Q. Su, M. Liu, A niching differential evolution algorithm for the large-scale combined heat and power economic dispatch problem, *Appl. Soft. Comput.*, **133** (2021), 108017. https://doi.org/10.1016/j.asoc.2021.108017
10. S. Khalfi, A. Draa, G. Iacca, A compact compound sinusoidal differential evolution algorithm for solving optimization problems in memory-constrained environments, *Expert Syst. Appl.*, **186** (2021), 115705. http://dx.doi.org/10.1016/j.eswa.2021.115705
11. A. W. Mohamed, A. A. Hadi, A. K. Mohamed, Differential evolution mutations: taxonomy, comparison and convergence analysis, *IEEE Access*, **9** (2021), 68629–68662. https://doi.org/10.1109/ACCESS.2021.3077242
12. M. Yang, C. Li, Z. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, *IEEE Trans. Cybern.*, **45** (2014), 302–315. https://doi.org/10.1109/TCYB.2014.2339495
13. S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evol. Comput.*, **15** (2010), 4–31. http://dx.doi.org/10.1109/TEVC.2010.2059031
14. J. Zhang, A. C. Sanderson, JADE: Adaptive differential evolution with optional external archive, *IEEE Trans. Evol. Comput.*, **13** (2009), 945–958. https://doi.org/10.1109/TEVC.2009.2014613
15. A. Qin, V. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Trans. Evol. Comput.*, **13** (2008), 398–417. http://dx.doi.org/10.1109/TEVC.2008.927706
16. S. Rahnamayan, H. R. Tizhoosh, M. M. A. Salama, Opposition-based differential evolution, *IEEE Trans. Evol. Comput.*, **12** (2008), 64–79. http://dx.doi.org/10.1109/TEVC.2007.894200
17. M. A. Ahandani, H. Alavi-Rad, Opposition-based learning in the shuffled differential evolution algorithm, *Soft Comput.*, **16** (2012), 1303–1337. http://dx.doi.org/10.1007/s00500-012-0813-9

18. H. Liu, J. Han, L. Yuan, B. Yu, Self-adaptive bare-bones differential evolution based on bi-mutation strategy, *J. Commun.*, **38** (2017), 201–212. http://dx.doi.org/10.11959/j.issn.1000-436x.2017051
19. G. Xu, R. Li, J. Hao, X. Zhao, A new multi-stage perturbed differential evolution with multi-parameter adaption and directional difference, *Nat. Comput.*, **19** (2020), 683–698. http://dx.doi.org/10.1007/s11047-018-9692-z
20. J. Kennedy, Bare bones particle swarms, in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS03)*, (2003), 80–87. http://dx.doi.org/10.1109/SIS.2003.1202251
21. Y. Wang, Z. Cai, Combining multi-objective optimization with differential evolution to solve constrained optimization problems, *IEEE Trans. Evol. Comput.*, **16** (2012), 117–134. https://doi.org/10.1109/TEVC.2010.2093582
22. J. Chen, Y. Gong, W. Chen, M. Li, J. Zhang, Elastic differential evolution for automatic data clustering, *IEEE Trans. Cybern.*, **51** (2019), 4134–4147. https://doi.org/10.1109/TCYB.2019.2941707
23. K. S. Tey, S. Mekhilef, M. Seyedmahmoudian, B. Horan, A. T. Oo, A. Stojcevski, Improved differential evolution-based MPPT algorithm using SEPIC for PV systems under partial shading conditions and load variation, *IEEE Trans. Ind. Inform.*, **14** (2018), 4322–4333. https://doi.org/10.1109/TII.2018.2793210
24. M. G. H. Omran, A. P. Engelbrecht, A. Salman, Bare bones differential evolution, *Eur. J. Oper. Res.*, **196** (2009), 128–139. http://dx.doi.org/10.1016/j.ejor.2008.02.035
25. H. Wang, S. Rahnamayan, H. Sun, M. G. H. Omran, Gaussian bare-bones differential evolution, *IEEE Trans. Cybern.*, **43** (2013), 634–647. https://doi.org/10.1109/TSMCB.2012.2213808
26. H. Peng, Z. Wu, X. Zhou, C. Deng, Bare-bones differential evolution algorithm based on trigonometry, *J. Comput. Res. Dev.*, **52** (2015), 2776. http://dx.doi.org/10.7544/issn1000-1239.2015.20140230
27. S. Wang, H. Yang, Y. Li, S. Han, B. Yang, Multi-runways independent approach scheduling using self-adaptive differential evolution algorithm with elite archive, *Adv. Eng. Sci.*, **49** (2017), 153–161. http://dx.doi.org/10.15961/j.jsuese.201600468
28. Y. Li, Z. Zhan, Y. Gong, W. Chen, J. Zhang, Y. Li, Differential evolution with an evolution path: A DEEP evolutionary algorithm, *IEEE Trans. Cybern.*, **45** (2014), 1798–1810. http://dx.doi.org/10.1109/TCYB.2014.2360752
29. L. Cui, G. Li, Z. Zhu, Q. Lin, K. Wong, J. Chen, et al, Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism, *Inf. Sci.*, **422** (2018), 122–143. http://dx.doi.org/10.1016/j.ins.2017.09.002
30. X. Zhao, S. Feng, J. Hao, X. Zuo, Y. Zhang, Neighborhood opposition-based differential evolution with Gaussian perturbation, *Soft Comput.*, **25** (2021), 27–46. http://dx.doi.org/10.1007/s00500-020-05425-2
31. Y. He, X. Wang, K. Liu, Y. Wang, Convergent analysis and algorithmic improvement of differential evolution, *J. Softw.*, **21** (2010), 875–885. http://dx.doi.org/10.3724/SP.J.1001.2010.03486
32. R. Li, X. Zhao, X. Zuo, J. Yuan, X. Yao, Memetic algorithm with non-smooth penalty for capacitated arc routing problem, *Knowl.-Based Syst.*, **220** (2021), 106957. http://dx.doi.org/10.1016/j.knosys.2021.106957

33. Q. Fan, W. Wang, X. Yan, Differential evolution algorithm with strategy adaptation and knowledge-based control parameters, *Artif. Intell. Rev.*, **51** (2019), 219–253. http://dx.doi.org/10.1007/s10462-017-9562-6
34. R. D. Al-Dabbagh, F. Neri, N. Idris, M. S. Baba, Algorithmic design issues in adaptive differential evolution schemes: review and taxonomy, *Swarm Evol. Comput.*, **43** (2018), 284–311. http://dx.doi.org/10.1016/j.swevo.2018.03.008
35. Y. Zuo, F. Zhao, Z. Li, A knowledge-based differential covariance matrix adaptation cooperative algorithm, *Expert Syst. Appl.*, **184** (2021), 115495. https://doi.org/10.1016/j.eswa.2021.115495
36. J. Liang, B. Qu, P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, in *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University*, **635** (2013), 490. Available from: http://www5.zzu.edu.cn/cilab/fblw/jsbg.htm.
37. L. Ma, M. Huang, S. Yang, R. Wang, X. Wang, An adaptive localized decision variable analysis approach to large-scale multi-objective and many-objective optimization, *IEEE Trans. Cybern.*, 2021. https://doi.org/10.1109/TCYB.2020.3041212

AIMS Press