Multiple Topology SHADE with Tolerance-based Composite Framework for CEC2022 Single Objective Bound Constrained Numerical Optimization

Bo Sun
School of Information Engineering
JiangXi University of Science and Technology
Ganzhou, China
sunbo@mail.jxust.edu.cn

Yafeng Sun
School of Information Engineering
JiangXi University of Science and Technology
Ganzhou, China
sunyafeng_jx@mail.jxust.edu.cn

Wei Li
School of Information Engineering
JiangXi University of Science and Technology
Ganzhou, China
liwei@jxust.edu.cn

***Abstract*—To further enhance the convergence performance and accuracy of SHADE, a SHADE with tolerance-based multiple topology selection framework (MTT_SHADE) is proposed in this paper. In MTT_SHADE, three population topologies are established employing the *k*-nearest neighbor network, small-world network, and random network, respectively, and the evolution of individuals depends on the neighborhoods derived from different topologies. The tolerance-based composite framework is proposed to select the appropriate topology for an individual at the same time. Specifically, local tolerance and global tolerance are predetermined, corresponding to the tolerance for individuals and the population, respectively. The topology involved in the evolution of the individual is replaced when the individual does not progress after successive iterations. The population that does not improve in effect after successive iterations are considered to have exceeded the global tolerance and the three population topologies are reconstructed. The CEC2022 competition on single objective bound-constrained numerical optimization and four state-of-the-art DE variants are employed to investigate the effectiveness of the proposed algorithm. Experimental results show that MTT_SHADE is competitive in terms of accuracy and convergence.**

Index Terms—Differential evolution algorithm, k-nearest neighbor, small-world, random network, CEC2022

I. INTRODUCTION

Differential evolution algorithm (DE) was first proposed by Storn and Price [1]. DE is one of the well-known evolutionary algorithms. At the first IEEE International Conference on Evolutionary Computation, its outstanding performance drew the attention of a huge number of academics. However, the DE algorithm mainly focuses on exploitation ability, which may have different effects on solving different types of problems. Moreover, the problems of over-dependence on parameter selection and insufficient local exploitation ability of the DE algorithm bring new challenges to the performance of the algorithm. Therefore, a lot of research work based on DE has been carried out and has a certain improvement effect. Since the setting of parameters has a great influence on the DE algorithm, a novel JADE algorithm is proposed to improve the optimization performance of the algorithm [2]. In JADE, a novel mutation technique called 'DE/current-to-*pbest*' is implemented, which includes an optional external archive and adaptive control parameter updating. JADE has the ability to diversify the population while also improving convergence performance. Besides, a DE variant named composite DE (CoDE) is proposed in 2011[3]. CoDE generates trial vectors using three trial vector generation techniques and three control parameter settings, as well as a random combination of trial vector generation strategies and three control parameter values. CoDE can promote convergence speed and accuracy. Then, success-history based parameter adaptation for differential evolution (SHADE) is proposed [4] in CEC2013. In SHADE, it guides the selection of future control parameter values by using a historical recollection of successful control parameter settings. SHADE has some advantages in convergence accuracy.

The above-mentioned literature can promote DE to a certain extent. However, the lack of diversity in the later stage of the algorithm and the problem of easy premature convergence still exist. So a SHADE with tolerance-based multiple topology selection framework (MTT_SHADE) is proposed. MTT_SHADE mainly includes two parts: multiple topology SHADE and tolerance-based composite framework. Aiming at the problem of insufficient diversity in the later stage of DE, this study establishes three network topologies (*k*-nearest neighbor network, small-world network, and random network) to make the evolution between individuals depend on the neighborhood derived from different topologies so that the population maintains a degree of diversity from start to finish. Among them, the *k*-nearest neighbor network is used to enhance the local exploitation ability of the population, the random network is used to enhance the global exploration ability, and the small-world network is used to balance the above two abilities. Moreover, considering that CoDE is competitive in improving the overall performance of DE, it ignores the problem that the population is easy to be locally optimal in the later stage. Hence, a tolerance-based composite framework is proposed. Local tolerance and global tolerance are used to adjust the individual state, to reduce the probability of the population falling into local optimization. Finally, the tolerance-based multiple topology selection framework is embedded in SHADE to comprehensively improve the convergence accuracy and improve the lack of diversity in the later stage of the population.

The rest of this paper is organized as follows: the second part introduces the related work in detail, such as SHADE, three network topologies, et al. The third part gives a minute description of the proposed algorithm. The fourth part is the experiment, which tests the CEC2022 competition on single objective bound-constrained numerical optimization and records the results, and comparisons with well-known algorithm variants. The fifth part briefly analyzes the time complexity of the algorithm. The sixth part summarizes the work of this study and looks forward to future work.

II. BACKGROUND

A. SHADE

Tanabe and Fukunaga [4] proposed a success-history based adaptive DE (SHADE), an enhancement to JADE [2]. In SHADE, it uses a history-based parameter adaptation scheme. This scheme use a historical memory *MCR*, *MF* stores a set of *CR*, *F* values that have performed well in the past, and generate new *CR*, *F* pairs by directly sampling the parameter space close to one of these stored pairs. The mutation strategy used in SHADE current-to-*pbest*/1 can be described as follows:

$$v_{i,G} = x_{i,G} + F_i \cdot (x_{pbest,G} - x_{i,G}) + F_i \cdot (x_{r1,G} - x_{r2,G}) \tag{1}$$

where $x_{pbest,G}$ is randomly selected from the top $NP \times p$ ($p \in [0, 1]$) members in the *G-th* generation. F_i is the *F* parameter used by individual x_i. The control parameter p_i is set according to the equation by generation, which is associated with each individual x_i.

$$p_i = rand[p_{\min}, 0.2] \tag{2}$$

where p_{min} is set $p_{min}=2/NP$, at least two individuals are selected.

In the history-based parameter adaptation strategy, SHADE maintains a historical memory with *H* entries for both of the DE control parameters *CR* and *F*, *MCR*, *MF*. In each generation, select a parameter from *MCR* and MF as CR_i and F_i respectively. The random selection equation is as follows:

$$CR_i = randn_i(M_{CR,r_i}, 0.1) \tag{3}$$

$$F_i = randn_i(M_{F,r_i}, 0.1) \tag{4}$$

The contents of memory are updated as follows at the end of the generation:

$$M_{CR,k,G+1} = \begin{cases} mean_{WA}(S_{CR}) & S_{CR} \neq 0 \\ M_{CR,k,G} & otherwise \end{cases} \tag{5}$$

$$M_{F,k,G+1} = \begin{cases} mean_{WL}(S_F) & S_F \neq 0 \\ M_{F,k,G} & otherwise \end{cases} \tag{6}$$

where S_{CR}, S_F denote the CR_i and F_i values used by successful individuals.

The weighted mean $mean_{\mathrm{WA}}(S_{CR})$ [5] and the weighted Lehmer mean $mean_{WL}(S_F)$ are computed as follows:

$$mean_{WA}(S_{CR}) = \sum_{k=1}^{|S_{CR}|} \omega_k \cdot S_{CR,k} \tag{7}$$

$$\omega_k = \frac{\Delta f_k}{\sum_{k=1}^{|S_{CR}|} \Delta f_k} \tag{8}$$

$$mean_{WL}(S_F) = \frac{\sum_{k=1}^{|S_F|} \omega_k \cdot S_{F,k}^2}{\sum_{k=1}^{|S_F|} \omega_k \cdot S_{F,k}} \tag{9}$$

After obtaining the mutant, cross operate it with the parent x_i, *G* to obtain u_i, *G*. The crossover equation is as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & rand[0,1) \le CR, or, j = j_{rand} \\ x_{j,i,G} & otherwise \end{cases} \tag{10}$$

where [0, 1) denotes a uniformly selected random number from [0, 1). j_{rand} denotes a decision variable index which is uniformly randomly selected from [1, *D*].

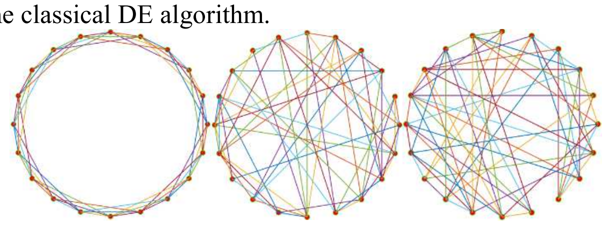
Finally, compare the offspring to the parent and keep the better ones for the future generation. The selection operation can be described as Eq. (11):

$$x_{i,G+1} = \begin{cases} u_{i,G} & f(u_{i,G}) \le f(x_{i,G}) \\ x_{i,G} & otherwise \end{cases} \tag{11}$$

B. Network topology

The proposal of the complex network provides a new method for research in the fields of physics, biology, chemistry, and computer science [6]. The *k*-nearest neighbor network is a common regular network, and the degree distribution is the *k*-centered δ function. A total number of nodes is *NP*, and each node in the network is connected to its *k* / 2 neighbors on the left and right. (*k* is an even number). The *k*-nearest neighbor network topology is shown in Fig. 1 (a). In 1998, the concept of the small-world network was defined mathematically for the first time [7]. The difference from the *k*-nearest neighbor network is that when *NP* nodes form a *k*-nearest neighbor network, each edge in the network is linked at random with probability *p*, and there is only one edge connecting any two nodes, and each node cannot be reunited with itself. When the number of network nodes *NP* is much greater than *k*, the network has sparsity and a high clustering coefficient. The randomized reconnection process reduces the average path length of the network and makes the network model have the characteristics of a small world. The small-world network is shown in Fig. 1 (b). The *k*-nearest neighbor network evolves into a random network when the probability *p* value is 1, and the probability of each edge in the random network is equal, as shown in Fig. 1 (c).

Fig. 1 is an example of three different network topologies. It can be seen from Fig.1 that the three network topologies have different characteristics. In Fig. 1 (a), the *k* value is set as 6 and the connection probability *p* is 0. Fig. 1 (b) sets *p* to 0.8 while keeping *k* at 6. Fig. 1 (c) is the figure drawn by setting the *p* value to 1, and *k* is still 6. If each node and other nodes connected to the node form a neighborhood, it is easy to find that the nodes in the *k*-nearest neighbor network are mainly affected by the left and right nodes. The nodes in the small-world network are affected by the left and right sides and some other nodes respectively. The neighborhood built by a small-world network for the population is relatively large. Nodes in a random network are equally affected by other nodes. Not hard to find that the function of a random network is equivalent to the way of selecting individuals by the classical DE algorithm.



(a) *k*-nearest neighbor network (b) Small-world network (c) Random network

Fig. 1. Network topology

III. PROPOSED MTT_SHADE

MTT_SHADE mainly includes two parts: multiple topology SHADE and tolerance-based composite framework. This section details the two strategies proposed.

A. Multiple Topology SHADE

The main idea of multiple topology SHADE is to construct multiple neighborhoods for individuals, to affect the mutation process and make individuals have broader choices in the process of evolution. Specifically, from the multi-topology of the population to the variation effect affecting individuals, there are four steps:

Step 1: Sorting. According to the individual fitness value, the population is sorted from better to worse. It is worth noting that the sorted results are only used in the second step to build the topology, not directly used in the evolution process.

Step 2: Building network topology. Firstly, according to the ranking results, taking the individual serial number as the node, the *k*-nearest neighbor network topology is constructed for the population, and then the small-world network topology is constructed based on the *k*-nearest neighbor network. There is no special operation in the construction of random network topology, and the adjacency relationship of nodes in the topology is random. This topology construction method not only takes into account the core characteristics of the network but also minimizes the complexity of the construction process.

Step 3: Composition neighborhood. For each individual, its position on the three network topologies is different, and its adjacency relationship with other individuals is also different. According to the adjacency relationship between individuals and other individuals in the three networks, three neighborhoods of individuals are established.

Step 4: Neighborhood individual participation variation. Using the tolerance-based composite framework, an appropriate neighborhood is selected for each individual. Further, the optimal individual in the neighborhood participates in the variation of the individual, as shown below:

$$v_{i,G} = x_{i,G} + F_i \cdot (x_{pbest,G} - x_{i,G}) + F_i \cdot (x_{nbest,G} - x_{r2,G}) \quad (12)$$

where, x_{nbest} is the best individual in the neighborhood of individual *i*. The tolerance-based composite framework will be introduced in Section III B detailed introduction.

The purpose of constructing three network topologies is to make similar individuals interact with each other after constructing *k*-nearest neighbor network, so that the population tends to local exploitation as a whole. Only a few individuals, such as those with poor tail, are affected by better individuals, making them tend to global exploration. The constructed random network topology is not affected by fitness ranking so each individual tends to global exploration. Small-world network topology is affected by fitness ranking, which can balance global exploration and local exploitation.

B. Tolerance-based Composite Framework

The fundamental goal of the tolerance-based composite framework is to make the improvement effect of the *i*+1 generation population as significant as possible compared with the *i* generation population. Under the guidance of this goal, select an appropriate network topology for individuals and make individuals move in a good direction. The framework consists of two parts: local tolerance and global tolerance. In the process of evolution, a local tolerance *TL* is set for each individual, i.e. *TL*(*i*), 1<=*i*<=*NP*. The global tolerance *TG* is designed for the overall effect of the population. Initially, the network topology of individual *i* is randomly selected and marked as *TS* (*i*). *TS*(*i*)=1, 2, 3, which represent *k*-nearest neighbor network, small-world network, and random network respectively. The tolerance-based composite framework is shown in algorithm 1.

Algorithm 1: The tolerance-based composite framework

```
Input: TS, Max_TG, Max_TL, fitness_pop, fitness_popold
For i=1:NP
   If fitness_pop(i)>=fitness_popold(i)
      TL(i)=TL(i)+1;
   Else
      TL(i)=0;
   End If
   If TL(i)>Max_TL
      TS(i) is randomly selected from two topologies other than the
      currently selected topology;
   End If
End For
If min(fitness_pop)>=min(fitness_popold)
   TG=TG+1;
Else
   TG=0;
End If
If TG>Max_TG
   Reconstruct three population topologies;
End If
Output: TS
```

In algorithm 1, when an individual has not been promoted for successive *TL* generations, the individual will abandon the currently selected network topology and choose randomly from the other two topologies. In addition, when the continuous *TG* generation of the population has not been improved, three network topologies are reconstructed. In this study, *Max_TG*=40， *Max_TL*=10. The two tolerances provide individuals and populations with a direction conducive to expanding to promising regions simply and efficiently.

C. MTT_SHADE

MTT_SHADE is motivated by the powerful improvement of EAs' local search ability brought by population topology, and the fusion of multiple population topologies has also been proved to be a powerful strategy. The tolerance-based composite framework provides the basis for the selection of topology. The pseudo-code of MTT_SHADE is shown in algorithm 2.

Algorithm 2: MTT_SHADE

```
1  Initialization population pop;
2  TS=randi(3,[NP,1]);
3  Sort the fitness by ascending;
4  Build three network topologies;
5  Construct three neighborhoods for each individual;
6  While nfes<max_nfes
7  The x_nbest of each individual is selected according to TS and neighborhood;
8     P=Eq. (2)
9     V<-Eq. (12);
10    u<- Binomial cross operation;
11    Popold=pop;
12    pop<- Selection operation;
13    Update TL, TG, TS using Algorithm 1;
14    Update F, CR using Eq. (3)-(4);
15    nfes=nfes+NP;
16 End While
```

In algorithm 2, in the initial stage, build a network topology. Then use a variable *TS* to randomly select a number for each individual from [1, 2, 3] as shown in the pseudo-code in line 2. The fitness values are sorted in ascending order. The purpose of this is to have an impact on the *k*-nearest neighbor network topology. Before entering the iteration process, three network topologies are constructed for the population, and a neighborhood is constructed for each individual. In the iterative process, x_{nbest} was selected to participate in the evolution of the population. Eq. (2) is used to determine the p value of each individual. Eq. (12) is used to mutate individuals, then cross and select the population, and algorithm 1 is used to update the network topology.

IV. EXPERIMENTAL STUDY

A. Experimental Setting

In this section, the performance of MTT_SHADE is evaluated on the CEC2022 test suits [12]. The test set includes twelve test functions, and contains four categories as TABLE I. TABLE I contains function categories, function names, and optimal values. Where F1 is an unimodal function, F2-F5 are basic functions, F6-F8 are hybrid functions, and F9-F12 are composition functions. The search scope is $[-100, 100]^D$. Test the functions in 10-D and 20-D respectively and the global optima of the functions are recorded in TABLE I [12]. In TABLE I, the F1-F5 functions are obtained by a function through a shift and partial or complete rotation. On the F6-F8 functions, the variables are randomly divided into some subcomponents and then different basic functions are used for different subcomponents. And the number of basic functions contained in each hybrid function is different. On the F9-F10 functions, they better combine the features of the sub-functions and keep the global or local optima consistent. And shifted and rotated functions are all the basic functions that have been employed in composition functions. This makes the functions more complex and poses a greater challenge to the performance of the algorithm.

Each function runs 30 times independently and the termination criteria to reach 200000 for 10-D and 1000000 for 20-D (*MaxFES*) evaluation of the objective function for each run. This paper uses a computer with a Win_64 bit, Intel (R) Core (TM) i5 - 6300hq CPU@ 2.30 GHz and 8 GB RAM. The version of MATLAB is MATLAB r2019a, which was used to follow through with the relevant experimental. The population size NP used MTT_SHADE is set according to most literature and historical experience. For this competition, the population size NP=60 for D=10 and NP=120 for D=20. It is found that the algorithm performs well when k value is 6 and p value is 0.8. Therefore, k=6 for three network and p=0.8 for small-world network in this study.

TABLE I. SUMMARY OF CEC2022 SINGLE OBJECTIVE BOUND CONSTRAINED NUMERICAL OPTIMIZATION

	No.	Functions	F_i^*
Unimodal Function	1	Shifted and full Rotated Zakharov Function	300
Basic Functions	2	Shifted and full Rotated Rosenbrock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's f6 Function	600
	4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
	5	Shifted and full Rotated Levy Function	900
Hybrid Functions	6	Hybrid Function 1 (N = 3)	1800
	7	Hybrid Function 2 (N = 6)	2000
	8	Hybrid Function 3 (N = 5)	2200
Composition Functions	9	Composition Function 1 (N = 5)	2300
	10	Composition Function 2 (N = 4)	2400
	11	Composition Function 3 (N = 5)	2600
	12	Composition Function 4 (N = 6)	2700
Search range: $[-100,100]^D$			

B. Optimization Performance

The performance of MTT_SHADE on the test set through error (the difference between the global optimal value determined by the algorithm and the objective function value) is analyzed in this section. The error values obtained for 10-D, and 20-D are tabulated in TABLE II and III. The error was evaluated as 0 when the difference between the ideal solution and the best solution found by the method was 1E-8 or less.

In TABLE II, the F1 function of MTT_SHADE on 10-D has good robustness and has excellent performance. In basic functions, the F2, F3, and F5 functions have excellent performance in both convergence accuracy and stability and can be solved to the optimal value. However, the average value shows that the stability of the F2 function is slightly worse than F3 and F5 functions. So the average value is slightly worse than theirs. The F4 function performs slightly worse than the other three basic functions. In hybrid functions, the F7 function can get the optimal value, but the average shows that its stability is insufficient. Although the F6 and F8 functions can not get the optimal value, they can improve the convergence accuracy. In composition functions, although the F11 function is composed of five complex functions, MTT_SHADE still shows excellent optimization ability. It can get the optimal value, and its stability is also better. The F12 function failed to get the optimal solution, mainly because the function form is particularly complex, which is composed of six functions, and there are many local optimal values. The F12 function still shows some advantages in global exploration ability.

In TABLE III, the F1 function of MTT_SHADE on 20-D still has good robustness. In basic functions, the F3 and F5 functions perform as well as on 10-D and can get the optimal value. The performance of the F2 function in 20-D is not as good as that in 10-D, which is mainly due to the characteristics of solving the problem. The F2 function is relatively smooth near the optimal value, and the difference between solutions is too small. The convergence accuracy of the F4 function is slightly worse than that of the other two functions. In hybrid functions, the performance of F7 function on 20-D is slightly worse than that on 10-D, and it can not get the optimal value. This also shows that the increase of dimension has a great impact on the optimization ability of the algorithm. The F6 function has a better effect on improving search accuracy. In composition functions, it is worth noting that the convergence accuracy of the F9 function on 20-D is better than that on 10-D. The F11 function fails to reach the optimal value like 10-D. The explanation for this could be that the composite function is more complex, preventing individuals from escaping the local optimal value.

TABLE II. PERFORMANCE ON 10-D

Func.	Best	Worst	Median	Mean	Std.
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	8.92E+00	3.99E+00	5.00E+00	2.31E+00

F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	1.99E+00	8.95E+00	3.98E+00	4.01E+00	1.56E+00
F5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	2.56E-02	5.00E-01	3.22E-01	3.10E-01	1.42E-01
F7	0.00E+00	3.91E-01	7.11E-02	8.47E-02	8.56E-02
F8	2.48E-02	2.03E+01	2.91E+00	6.43E+00	7.02E+00
F9	2.29E+02	2.29E+02	2.29E+02	2.29E+02	0.00E+00
F10	1.00E+02	2.05E+02	1.00E+02	1.04E+02	1.91E+01
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	1.59E+02	1.65E+02	1.62E+02	1.62E+02	1.66E+00

TABLE III. PERFORMANCE ON 20-*D*

Func.	Best	Worst	Median	Mean	Std.
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	4.49E+01	4.91E+01	4.91E+01	4.84E+01	1.59E+00
F3	0.00E+00	7.97E-08	0.00E+00	2.66E-09	1.46E-08
F4	4.98E+00	1.19E+01	7.96E+00	8.13E+00	1.65E+00
F5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	1.51E-01	5.24E+00	1.24E+00	1.45E+00	1.21E+00
F7	2.96E+00	2.16E+01	1.09E+01	1.20E+01	6.32E+00
F8	4.96E+00	2.09E+01	2.04E+01	1.98E+01	2.85E+00
F9	1.81E+02	1.81E+02	1.81E+02	1.81E+02	8.67E-14
F10	0.00E+00	1.01E+02	1.00E+02	9.70E+01	1.83E+01
F11	3.00E+02	4.00E+02	3.00E+02	3.03E+02	1.83E+01
F12	2.32E+02	2.41E+02	2.34E+02	2.35E+02	2.83E+00

C. Comparisons with advanced DE algorithms

To verify the performance of MTT_SHADE in solving the CEC2022 test suits, 4 advanced comparison algorithms are used to compare with MTT_SHADE. These algorithms are recently proposed state-of-art algorithms that attract many researchers' attention: CoDE [3], SHADE [4], RNDE [9], NBOLDE [10]. The parameters settings and the experimental setup for the peer algorithms are the same as that in the corresponding papers. For fairness, each algorithm was run 30 times independently and the mean and standard deviation were recorded.

TABLE IV. *COMPARISONS WITH COMPONENT ALGORITHMS ON 10-D*

Func		CoDE	SHADE	RNDE	NBOLDE	MTT_SHADE
F1	Best	2.6074E-10	**0.0000E+00**	**0.0000E+00**	1.0061E+03	**0.0000E+00**
	Mean	9.5132E-10	**0.0000E+00**	**0.0000E+00**	3.7559E+03	**0.0000E+00**
F2	Best	5.5372E-09	3.9866E+00	**0.0000E+00**	1.7198E+01	**0.0000E+00**
	Mean	2.2591E+00	5.6298E+00	**1.0631E+00**	6.7971E+01	5.0039E+00
F3	Best	1.9718E-09	**0.0000E+00**	**0.0000E+00**	3.5609E+00	**0.0000E+00**
	Mean	9.8950E-09	**0.0000E+00**	**0.0000E+00**	1.2952E+01	**0.0000E+00**
F4	Best	5.4813E+00	**9.9671E-01**	3.9798E+00	1.5097E+01	1.9901E+00
	Mean	1.4278E+01	**3.1204E+00**	7.3881E+00	2.9019E+01	4.0135E+00
F5	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	1.3243E+01	**0.0000E+00**
	Mean	1.5158E-14	**0.0000E+00**	**0.0000E+00**	7.2886E+01	**0.0000E+00**
F6	Best	1.6328E-01	**1.1978E-02**	1.8236E-02	2.3146E+02	2.5588E-02
	Mean	3.2561E-01	**2.7403E-01**	2.7797E-01	4.2057E+03	3.0966E-01
F7	Best	2.6245E-03	2.2082E-03	**0.0000E+00**	2.1146E+01	**0.0000E+00**
	Mean	**7.5538E-03**	1.2247E+01	2.0803E-02	3.8329E+01	8.4713E-02
F8	Best	3.6355E+00	2.9379E-01	3.4629E-02	1.2368E+01	**2.4831E-02**
	Mean	6.3811E+00	2.1697E+00	**1.5752E-01**	2.3722E+01	6.4347E+00
F9	Best	**2.2928E+02**	**2.2928E+02**	**2.2928E+02**	2.3199E+02	**2.2928E+02**
	Mean	**2.2928E+02**	**2.2928E+02**	**2.2928E+02**	2.9971E+02	**2.2928E+02**
F10	Best	1.0024E+02	**1.0015E+02**	1.0019E+02	1.0059E+02	1.0017E+02
	Mean	1.0032E+02	**1.0024E+02**	1.0702E+02	1.1937E+02	1.0375E+02
F11	Best	5.4570E-12	**0.0000E+00**	**0.0000E+00**	8.6083E+01	**0.0000E+00**
	Mean	1.1081E-11	**0.0000E+00**	**0.0000E+00**	1.6914E+02	**0.0000E+00**
F12	Best	1.5862E+02	**1.5862E+02**	1.6270E+02	1.6446E+02	**1.5862E+02**
	Mean	**1.6005E+02**	1.6128E+02	1.6453E+02	1.7512E+02	1.6193E+02
Wilcoxon test(Best)		8+/4≈/0-	3+/8≈/1-	6+/6≈0-	12+0≈0-	
Wilcoxon test(Mean)		5+/4≈/3-	1+/8≈/3-	3+/6≈/3-	12+/0≈/0-	

TABLE V. *COMPARISONS WITH COMPONENT ALGORITHMS ON 20-D*

Func		CoDE	SHADE	RNDE	NBOLDE	MTT_SHADE
F1	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	7.7335E+03	**0.0000E+00**
	Mean	**0.0000E+00**	1.8948E-15	1.8948E-15	1.9302E+04	**0.0000E+00**
F2	Best	**4.4895E+01**	**4.4895E+01**	**4.4895E+01**	1.7872E+02	**4.4895E+01**
	Mean	**4.5314E+01**	4.8805E+01	4.7967E+01	2.9857E+02	4.8386E+01
F3	Best	1.1369E-13	**0.0000E+00**	**0.0000E+00**	2.0152E+01	**0.0000E+00**
	Mean	1.1369E-13	2.6359E-07	**4.9264E-14**	3.2911E+01	2.6572E-09
F4	Best	7.9597E+00	5.9699E+00	8.9546E+00	8.2275E+01	**4.9753E+00**
	Mean	1.8606E+01	8.6896E+00	2.0706E+01	1.1529E+02	**8.1262E+00**
F5	Best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	4.4256E+02	**0.0000E+00**
	Mean	**0.0000E+00**	2.9843E-03	7.5791E-15	7.8285E+02	**0.0000E+00**
F6	Best	4.0306E-01	1.6254E-01	1.9391E-01	1.1211E+05	**1.5144E-01**
	Mean	**4.9959E-01**	1.9242E+01	9.6933E+00	5.9316E+06	1.4544E+00
F7	Best	1.3305E+00	4.0362E+00	**0.0000E+00**	4.2890E+01	2.9571E+00
	Mean	1.1130E+01	1.9644E+01	**2.3381E+00**	8.9855E+01	1.2012E+01
F8	Best	1.8495E+01	1.4221E+01	**4.7604E-01**	2.8207E+01	4.9588E+00
	Mean	2.2896E+01	1.9888E+01	**1.8157E+01**	3.7818E+01	1.9752E+01
F9	Best	**1.8078E+02**	**1.8078E+02**	**1.8078E+02**	2.1492E+02	**1.8078E+02**
	Mean	**1.8078E+02**	**1.8078E+02**	**1.8078E+02**	2.7942E+02	**1.8078E+02**
F10	Best	1.0033E+02	1.0027E+02	1.0024E+02	1.0231E+02	**0.0000E+00**
	Mean	1.0042E+02	1.0798E+02	1.0028E+02	1.5711E+02	**9.7033E+01**
F11	Best	4.0000E+02	**3.0000E+02**	**3.0000E+02**	8.7522E+02	**3.0000E+02**
	Mean	4.0000E+02	3.0333E+02	**3.0000E+02**	1.6526E+03	3.0333E+02
F12	Best	**2.2886E+02**	2.3174E+02	2.3053E+02	3.0591E+02	2.3186E+02
	Mean	**2.3101E+02**	2.3528E+02	2.3283E+02	3.6653E+02	2.3533E+02
Wilcoxon test(Best)		7+/4≈/1-	3+/9≈/0-	4+/5≈/3-	12+/0≈/0-	
Wilcoxon test(Mean)		4+/4≈/4-	3+/9≈/0-	3+/5≈/4-	12+/0≈/0-	

TABLE IV is the comparison result of 5 algorithms in 10-*D*, TABLE V is the comparison result of 20-*D*. Statistical analysis of the best and mean values was performed using the Wilcoxon test with a significance of 0.05. The angle of the 10-dimensional optimal value shows that MTT_SHADE search effect ranks first overall. From an average point of view, MTT_SHADE is slightly worse than SHADE. This shows that MTT_SHADE has a certain role in improving the solution accuracy. From the statistical results in TABLE V, it is obvious that MTT_SHADE shows the overall ranking first in terms of the best value, and there is 1 function that performs slightly worse than RNDE in terms of the average value. To sum up, the overall performance of MTT_SHADE in 10- and 20-*D* ranks first, which strongly proves that the algorithm has a great improvement effect.

In short, MTT_SHADE has certain robustness in optimization. The reason is that the proposal of local tolerance can improve individual quality. When individuals may fall into local optimization, local tolerance can replace other network neighborhood topologies for them. The global topology can improve the state of the whole population. When the overall optimization effect of the population is poor, the network neighborhood topology is reconstructed to improve the situation of local optimization. This also shows that the framework proposed in this study can promote the improvement of global exploration ability and local exploitation ability.

D. Discussion of Convergence and Stability

To verify the convergence speed and stability of MTT-SHADE, we select one of the four types of functions for analysis. In Fig. 2, the F1, F2, and F11 functions converge significantly faster than other algorithms and reach the best value in the early stage. Although the convergence speed of MTT_SHADE on the F6 function is inferior to SHADE and RNDE in the later stage, it can converge quickly in the early stage. In Fig. 3, MTT_SHADE converges significantly faster on the F2 function than the other four variants. MTT_SHADE behaves similarly to the 10-*D* case on the F6 function but converges faster than SHADE. And the convergence accuracy of all algorithms in this function is not good on the F6 function, the reason should be attributed to the characteristics of the function so that the algorithm cannot escape the local optimum. The convergence speed of MTT_SHADE on the F8 and F11 functions is almost the same as SHADE, but the convergence accuracy is better than SHADE.

Figs. 4 and 5 clearly show us the stability of MTT_SHADE in the optimization process. In Fig. 4, on the F1 function, except for NBOLDE, the convergence effects of the other four algorithms are relatively stable, because the function is relatively simple. SHADE and RNDE are more stable on the F4 function, and the other three algorithms have 1 abnormal point. Although CoDE, RNDE, MTT_SHADE all have 1 abnormal point on the F7 function, the overall stability is better. CoDE is more stable on the F10 function. RNDE, NBOLDE, and MTT_SHADE all have abnormal points, but MTT_SHADE is the least and more stable. In Fig. 5, both SHADE and RNDE have abnormal points on the F7 function, but SHADE has more. Both SHADE and NBOLDE have abnormal points on the F10 function, but SHADE is more stable than NBOLDE. In summary, it is not difficult to find that MTT_SHADE has almost no abnormal points and is more stable.

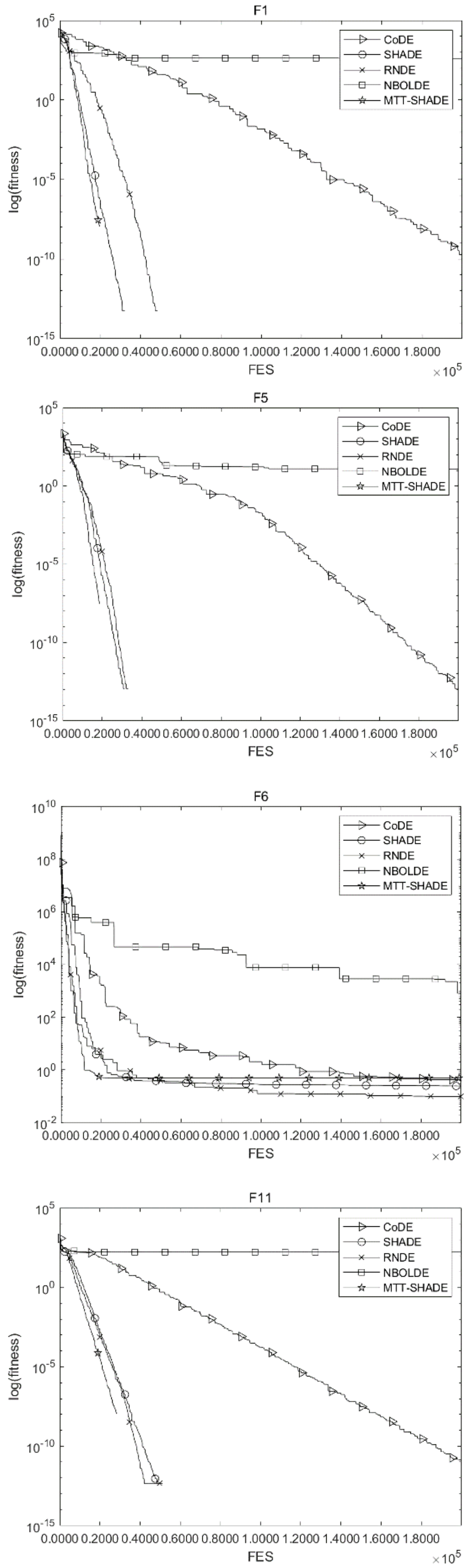


Fig. 2. Convergence effect of 5 algorithms on 10-*D*

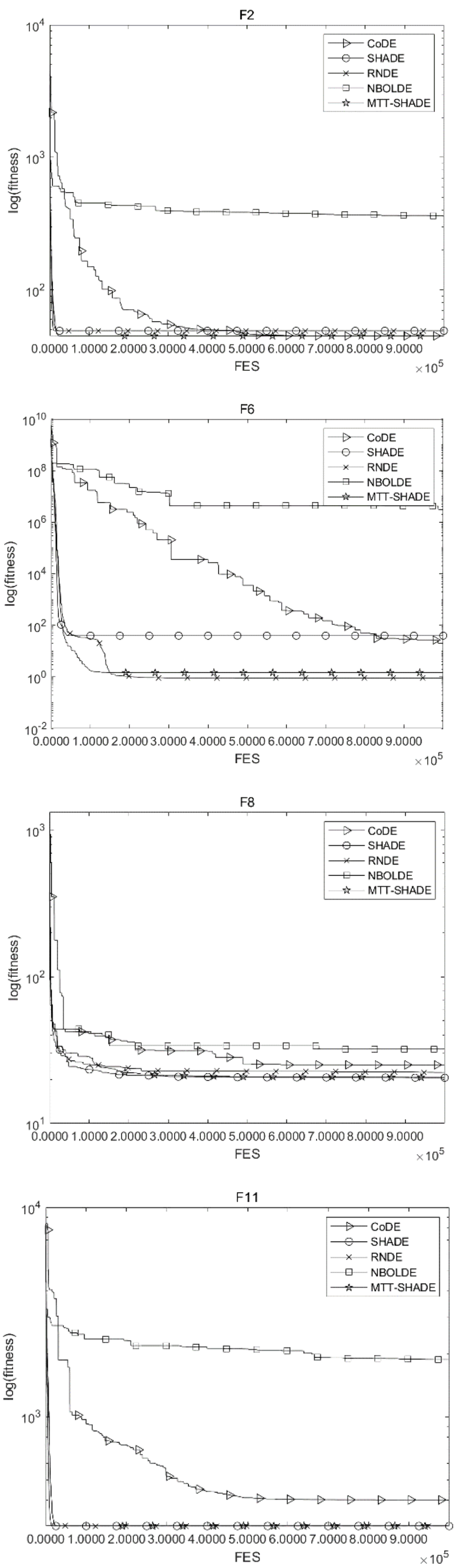


Fig. 3. Convergence effect of 5 algorithms on 20-*D*

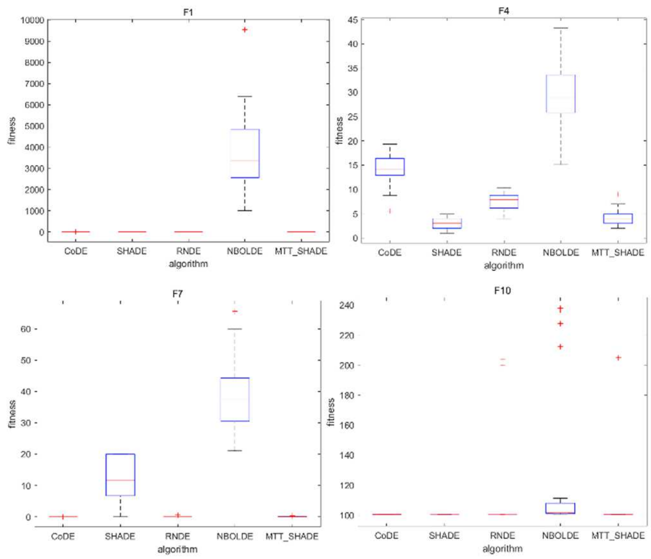


Fig. 4. Stability effect of 5 algorithms on 10-*D*

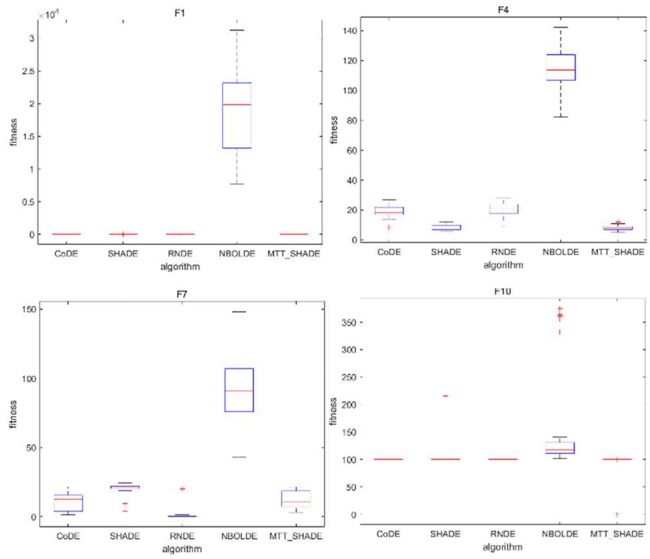


Fig. 5. Stability effect of 5 algorithms on 20-*D*

E. Time Complexity

For the calculation of algorithm time complexity, reference [7] gives detailed calculation steps, and the calculation rules are as follows:

1. Run a test program [8] and calculate a time for the above = *T0*:

2. Evaluate the computing time just for Function 1. The number of evaluations is 200000 with *D*-dimension, it is *T1*;

3. Evaluate the complete computing time for MTT_SHADE. The number of evaluations is 200000 with *D*-dimension, it is *T2*;

4. Execute step 3 five times. The mean of the obtained values of *T2*=*mean* (*T2*).

The calculation of time complexity has three parameters: *T2*, *T1*, *T0*, and (*T2*-*T1*)/*T0*. The computational complexity is tabulated in TABLE VI. In TABLE VI, *T0* is small, and *T2* is larger than *T1* because of the increased time complexity in building the network topology. The value of *T2* does not increase with the increase of dimension. In addition, Time complexity increases with the increase of dimension and evaluation times, but the increase is not obvious, which proves that MTT_SHADE can be used as an efficient algorithm to solve a variety of difficult problems.

TABLE VI. TIME COMPLEXITY

	T0	*T1*	*T2*	(*T2-T1*)/*T0*
D=10	0.0625	0.5344	6.7625	99.6496
D=20	0.0625	0.5708	6.8781	100.9168

V. CONCLUSION

In this study, MTT_SHADE is proposed to test CEC2022 single objective suits. A tolerance-based multiple topology selection framework is shallow into SHADE to increase population diversity and reduce the probability of the population falling into local optimization. The framework consists of two parts, a multiple topology selection strategy, and a tolerance-based composite framework. Multiple topology selection strategy constructs neighborhoods for individuals through *k*-neighborhood network, small-world network, and random network so that the excellent individuals in the neighborhood can participate in population evolution. Using the differences between the three neighborhoods, we can enhance the individual's exploitation ability and exploration ability， and balance the two. The tolerance-based composite framework selects the appropriate network topology for each individual through local tolerance and global tolerance. Through the analysis of the experiment results, it is concluded that MTT_SHADE has certain competitiveness in improving the convergence accuracy and speed. However, because the algorithm's strong performance comes at the expense of time complexity, future work will focus on reducing the algorithm's time complexity.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science F oundation of China (Grant Nos. 61903089, 62066019), the Natural Science Foundation of Jiangxi Province (Grant Nos. 20202BABL202020, 20202BAB202014) and the National Key Research and Development Program of China (Grant N o. 2020YFB1713700) and the Graduate Innovation Foundation of JiangXi University of Science and Technology (Grant No. XY2021-S094).

REFERENCES

[1] R. Storn, K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11(4), pp. 341–359, 1997. https://doi.org/10.1023/A:1008202821328.

[2] J. Zhang A. C. Sanderson, “JADE: Adaptive differential evolution with optional external archive,” IEEE Tran. Evol. Comput., vol. 13, no. 5, pp. 945–958, 2009.

[3] Y. Wang, Z. Cai, Q. Zhang, “Differential evolution with composite trial vector generation strategies and control parameters,” IEEE Tran. Evol. Comput., vol. 15, no. 1, pp. 55–66, 2011.

[4] R. Tanabe, A. Fukunaga, “Success-history based parameter adaptation for differential evolution,” 2013 IEEE congress on evolutionary computation, IEEE, 2013, pp. 71-78.

[5] F. Peng, K. Tang, G. Chen, X. Yao,“Multi-start JADE with knowledge transfer for numerical optimization,” in IEEE CEC, 2009, pp. 1889–1895.

[6] D. Xu, C. Yang, W. Gui, “Pinning control of the nearest-neighbor coupled networks based on topology structure,” 2010 8th World Congress on Intelligent Control and Automation. IEEE, 2010, pp. 3040-3045.

[7] D. J. Watts, S. H. Strogatz, “Collective dynamics of ‘small-world’ network model,” Physics Letters A, vol. 263(4-6), 1999, pp. 341-346.

[8] A. Kumar, V. K. Price, A. W. Mohamed, A. A. Hadi, P. N. Suganthan, “Problem Definitions and Evaluation Criteria for the CEC 2022 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization”.

[9] H. Peng, Z. Guo, C. Deng, Z. Wu. Enhancing differential evolution with random neighbors based strategy. Journal of Computational Science, 26, 501–511, 2018.

[10] W. Deng, S. Shang, X. Cai, H. Zhao, Y. Song, J. Xu. An improved differential evolution algorithm and its application in optimization problem. Soft Computing, 25(7), 5277–5298, 2021.