

Differential evolution optimization with time-frame strategy adaptation

Shih-Chang Wang¹ 

Published online: 11 January 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract In differential evolution (DE) research, many successful empirical guidelines in selecting appropriate trial vector generation strategies and control parameter values for various problems have been investigated. The comprehensive exploration of the experience can be an effective way to develop an advanced DE variant. In this paper, an improved DE approach with time-frame strategy adaptation called the time-frame adaptive differential evolution (TFADE) is proposed. It employs diverse trial vector generation strategies with various control parameter values that can be adaptively determined to generate promising solutions and dynamically adjusted to deal with premature convergence during evolution, according to successful experience over a period of preceding generations called the time frame. In the experimental study, TFADE is compared with 4 commonly used conventional DEs, 3 outstanding state-of-the-art adaptive DEs, and 2 novel non-DE approaches, evaluated by a test suite of 25 benchmark functions. The experimental results show that the performance of TFADE is significantly superior to these competitors.

Keywords Differential evolution · Evolutionary computation · Soft computing · Computational intelligence

1 Introduction

Among contemporary heuristic techniques, evolutionary algorithms (EAs), inspired by biological evolution, have been increasingly important in recent decades. They have been the kernel of the field known as evolutionary computation (EC) in artificial intelligence and succeeded in solving various problems. Nowadays, using EAs to solve optimization problems has become a common practice due to their efficient and effective performance in solving complex problems that cannot be easily solved by conventional techniques.

The differential evolution (DE) algorithm, proposed by [Storn and Price \(1995\)](#), is a popular evolutionary approach and a promising global optimizer in the continuous search domain. It is a population-based stochastic search technique employing mutation, crossover, and selection operations to move a population towards promising solutions. DE is a simple yet powerful search technique revealing remarkable performance in a wide variety of problems. It has been successfully utilized in diverse fields such as scheduling and planning problems ([Cheng et al. 2014](#); [Damak et al. 2009](#); [Wang and Yeh 2014](#)), data clustering ([Dong et al. 2014](#); [Xiang et al. 2015](#)), pattern recognition ([Das and Konar 2009](#); [Du et al. 2007](#); [Maulik and Saha 2009](#)), and power dispatch ([Chiou 2009](#); [Varadarajan and Swarup 2008](#)).

The performance of DE highly depends on two parts. One is the trial vector generation strategy, i.e., the mutation and crossover operations; the other is the control parameter setting, i.e., the scaling factor (F), the crossover rate (CR), and the population size (NP). In DE research, many encouraging trial vector generation strategies have been reported, different strategies can be suitable for solving different kinds of problems ([Qin et al. 2009](#)). However, the performance of a given trial vector generation strategy is still significantly influenced by its control parameter setting. Although there are many

Communicated by V. Loia.

✉ Shih-Chang Wang
scwang@mail.lhu.edu.tw

¹ Department of Business Administration, Lughwa University of Science and Technology, 300, Sec. 1, Wanshou Rd., Guishan District, Taoyuan City 33306, Taiwan

important guidelines for the choice of an appropriate control parameter setting (Gämperle et al. 2002; Mezura-Montes et al. 2006; Price et al. 2005), the interaction between control parameter settings and DE performance is still complicated and not completely understood.

Therefore, in order to successfully solve a practical problem by DE, as the characteristics of the given problem are usually unknown, it is commonly required to perform a trial-and-error procedure to choose an appropriate strategy and fine-tune its corresponding parameter setting. However, such trial-and-error search process is really a time-consuming and error-prone task requiring high computational costs. Besides, the population of target vectors may move through different kinds of regions during evolution. Using diverse trial vector generation strategies coupled with various control parameter settings at different stages of evolution can be more effective than just single one (Mallipeddi et al. 2011; Qin et al. 2009; Yong et al. 2011). Therefore, for DE it is preferable to adaptively determine an appropriate strategy and its parameter setting during evolution. There also has been an increasing interest in designing new DE variants with adaptive ability in the recent decade.

In this paper, we propose a novel adaptive DE variant called the time-frame adaptive differential evolution (TFADE) algorithm to avoid the time-consuming trial-and-error search process for seeking a suitable strategy and its parameter values. Both of which can be adaptively determined to generate promising solutions and dynamically adjusted to deal with the difficulty while premature convergence or stagnation occurs during evolution, according to successful experience over a period of preceding generations called the time frame. The rest of this paper is organized as follows. Section 2 reviews related work on conventional DE and state-of-the-art adaptive DE variants. Section 3 introduces the details of the proposed time-frame adaptive differential evolution algorithm. Some numerical experiments, results and discussions are reported in Sect. 4. Section 5 concludes this paper.

2 Related work

2.1 The differential evolution (DE) algorithm

An initialized population of NP D -dimensional parameter vectors, $\mathbf{x}_{i,G} = (x_{1i,G}, x_{2i,G}, \dots, x_{Di,G})$ where $i = 1, 2, \dots, NP$ and G be a generation, called target vectors encodes some candidate solutions towards the global optimum. The target vectors in the population are randomly generated from a uniform distribution with minimum bounds of $\mathbf{x}_{\min} = \{x_{1\min}, x_{2\min}, \dots, x_{D\min}\}$ and maximum bounds of $\mathbf{x}_{\max} = \{x_{1\max}, x_{2\max}, \dots, x_{D\max}\}$ within a search space. After initialization the population performs a series of evolutionary

operations, which are iterations of mutation, crossover, and selection. The mutation and crossover operations are utilized to generate new vectors called trial vectors. Both operations are so-called the trial vector generation strategy. After that, the selection operation then determines whether each of the new trial vectors will survive into next generation. The common evolutionary process of DE is described as follows.

1. Mutation operation

For each target vector $\mathbf{x}_{i,G}$, DE employs the following mutation operation to produce its corresponding mutant vector $\mathbf{v}_{i,G}$ (for the case of DE/rand/1):

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}), \tag{1}$$

where the indices $r1, r2$ and $r3$ are mutually exclusive integers randomly selected from the set of $\{1, 2, \dots, NP\} \setminus \{i\}$. F is a mutation factor called the scaling factor for scaling the difference vector $(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G})$. It is a positive real number where $F \in (0, 1+)$ that controls the rate at which the population evolves. It is generally the mutation operation that separates one DE variant from another. The mutation operation given in (1), which uses a randomly selected vector $\mathbf{x}_{r1,G}$ and one weighted difference vector $F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G})$, is denoted as DE/rand/1. This is the idea how the different DE variants are named. The other commonly used conventional DE variants (Das et al. 2009; Price et al. 2005; Price 1999) are listed as follows:

$$\text{DE/rand/2} : \mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}) + F(\mathbf{x}_{r4,G} - \mathbf{x}_{r5,G}) \tag{2}$$

$$\text{DE/best/1} : \mathbf{v}_{i,G} = \mathbf{x}_{\text{best},G} + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) \tag{3}$$

$$\text{DE/best/2} : \mathbf{v}_{i,G} = \mathbf{x}_{\text{best},G} + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) + F(\mathbf{x}_{r3,G} - \mathbf{x}_{r4,G}) \tag{4}$$

$$\text{DE/rand - to - best/1} : \mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F(\mathbf{x}_{\text{best},G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) \tag{5}$$

$$\text{DE/rand - to - best/2} : \mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F(\mathbf{x}_{\text{best},G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) + F(\mathbf{x}_{r3,G} - \mathbf{x}_{r4,G}) \tag{6}$$

$$\text{DE/current - to - rand/1} : \mathbf{v}_{i,G} = \mathbf{x}_{i,G} + K(\mathbf{x}_{r1,G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}) \tag{7}$$

2. Crossover operation

Once the mutation operation is finished, the crossover operation is then employed to each pair of the target vector $\mathbf{x}_{i,G}$ with its corresponding mutant vector $\mathbf{v}_{i,G}$, to generate a new vector $\mathbf{u}_{i,G} = (u_{i1,G}, u_{i2,G}, \dots, u_{iD,G})$ called the trial vector. The binomial crossover operation is the basic version defined as follows:

$$\mathbf{u}_{i,G} = u_{ij,G} = \begin{cases} v_{ij,G}, & \text{if } (\text{rand}_j \leq CR) \text{ or } (j = j_{\text{rand}}) \\ x_{ij,G}, & \text{otherwise} \end{cases} \quad (8)$$

where $j = 1, 2, \dots, D$, rand_j is a uniform random number generated within the range of $[0, 1]$. $CR \in [0, 1]$ is the crossover probability called the crossover rate, which is a user-defined value controls the fraction of parameter values that are copied from mutant vectors. $j_{\text{rand}} \in \{1, 2, \dots, D\}$ is a randomly chosen index ensures that the trial vector $\mathbf{u}_{i,G}$ does not duplicate the target vector $\mathbf{x}_{i,G}$.

3. Selection operation

After the mutation and crossover operations, the selection operation selects a better one between the target vector $\mathbf{x}_{i,G}$ and the trial vector $\mathbf{u}_{i,G}$ into next generation according to the following fitness measurement of a greedy selection scheme:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G}, & \text{if } \text{fitness}(\mathbf{u}_{i,G}) \text{ is better than } \text{fitness}(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G}, & \text{otherwise} \end{cases} \quad (9)$$

2.2 DE with adaptive scheme

The performance of DE is significantly influenced by its control parameter settings, but the interaction between both is still complicated and not completely understood. Many adaptive DE variants then have been proposed to dynamically adjust control parameter values without involving the knowledge of the relationship among strategies, control parameters, and the characteristics of given problems.

Abbass (2002) adapted the crossover rate CR by encoding it into each individual to simultaneously evolve with other parameter for multi-objective optimization problems. The scaling factor F is generated for each variable from a Gaussian distribution $N(0, 1)$. Teo (2006) then proposed a DE variant with a dynamic population sizing strategy called DESAP based on Abbass' self-adaptive Pareto DE. He implemented two versions of DESAP using absolute and relative encoding methodologies for dynamically self-adapting the population size parameter. Das et al. (2005) linearly reduced the scaling factor F with increasing generation count from a maximum to a minimum value. They also employed a uniform distribution between 0.5 and 1.5 with a mean value of 1 to obtain a new hybrid DE variant. Brest et al. (2006) encoded control parameters F and CR into the individuals and adjusted by introducing two new parameters τ_1 and τ_2 . In

their algorithm (called jDE), a set of F values are assigned to individuals in the population. Then, a random number rand is uniformly generated in the range of $[0, 1]$. If $\text{rand} < \tau_1$, the F is re-initialized to a new random value in the range of $[0, 1]$, otherwise it is kept unchanged. The CR is adapted in the same manner but with a different re-initialized range of $[0, 1]$.

Later, adapting not only parameter values but also trial vector generation strategies may bring great performance for DE, some outstanding contemporary adaptive DE schemes focused on both adaptations have been proposed. Qin et al. (2009) proposed a self-adaptive DE algorithm named SaDE, in which both trial vector generation strategies and their associated control parameter values are gradually self-adapted by learning from their previous experience in generating promising solutions. As a result, more suitable strategies with its control parameter settings can be determined adaptively to match different phases of search process. The performance of SaDE is extensively evaluated by 26 bound-constrained numerical optimization problems and compared with several conventional DEs and state-of-the-art adaptive DE variants. Zhang and Sanderson (2009) proposed an adaptive differential evolution called JADE to improve the performance of DE by implementing an efficient mutation scheme of "DE/current-to-pbest" with the optional external archive. The "DE/current-to-pbest" is a generalization of the conventional "DE/current-to-best" and the optional archive operation saves historical information which has the potential to provide promising direction of search for evolution progress. The simulation results show that JADE can not only diversify the population, but also improve convergence performance. Liu and Lampinen (2005) introduced a fuzzy adaptive differential evolution (FADE) using fuzzy logic controllers into DE, the inputs of which incorporate relative function values and individuals of successive generations to adapt the parameters of mutation and crossover operations. The experimental results show that the FADE algorithm outperforms several conventional DEs on higher dimensional problems. Mallipeddi et al. (2011) proposed an ensemble of trial vector generation strategies and control parameters called EPSDE. It contains a pool of distinct trial vector generation strategies and a pool of control parameter values competing to produce offspring during evolution process. Yong et al. (2011) combined several effective trial vector generation strategies with some control parameter settings, proposed a method called composite DE (CoDE). It uses three trial vector generation strategies and three control parameter settings randomly combining them to generate promising solutions.

In recent years, an increasing number of researchers have contributed their effort to enhance the efficiency and effectiveness of DE. Zou et al. (2013) introduced a modified differential evolution algorithm (MDE), in which two distributions

of Gauss and Uniform are employed to adjust the scaling factor F and crossover rate CR, and two mutation strategies are adopted to produce better solutions. To guarantee the quality of the population, MDE uses an external archive, and some solutions of high quality in the external archive can be selected for candidate solutions. Lu et al. (2014) proposed a framework called DE with surrogate-assisted self-adaptation (DESSA) to deal with computationally expensive problems. DESSA generates multiple trial vectors using different trial vector generation strategies and parameter settings, and then employs a surrogate model to identify the potentially best trial vector to undergo real fitness evolution. The surrogate model acts like a strategy and parameter setting selector which aims to identify the most suitable strategy and parameter setting for each target vector. Yeh et al. (2014) proposed a novel grey adaptive DE algorithm (GADE) by updating the control parameters in adaptive manner with the help of grey relational analysis. In the algorithm, each target vector using strategies has its own values of scaling factor F and crossover rate CR depending on corresponding grey relational grade. Since the relational grade of an individual is varied over the generations, those two control parameters can be adaptive time-varying. Based on the works coupled with grey evolutionary analysis (GEA), they also proposed a GEA-based DE algorithm (GEA-DE) which enables the control parameters F and CR to adapt in the evolution process by introducing a GEA-based parameter automation approach (Yeh et al. 2015). The simulation results show that GEA-DE with the best mutation strategy (GEA/best) outperforms DE/rand/1, DE/best/1, jDE, and SaDE on most of the test functions in terms of solution accuracy and convergence speed.

3 The time-frame adaptive differential evolution (TFADE) algorithm

To solve a given problem by DE, using diverse trial vector generation strategies coupled with various control parameter settings at different stages of evolution can be more effective than just single one (Mallipeddi et al. 2011; Qin et al. 2009; Yong et al. 2011). Moreover, the performance of DE highly depends on the selection of the trial vector generation strategy and its control parameter setting. To choose a suitable trial vector generation strategy coupled with an appropriate control parameter setting for a given problem depends on the characteristics of the problem. It is common to perform a trial-and-error search for choosing a trial vector generation strategy and fine-tune its control parameter values, which would be really a time-consuming and error-prone task. In the last decade, DE researchers have investigated many empirical guidelines in selecting suitable trial vector generation strategies and control parameter settings for various problems. However, when solving a practical problem,

the characteristics of the problem are usually ambiguous, resulting in easily choosing the bad ones. An inappropriate choice of the strategy and control parameter setting may lead to premature convergence or stagnation during evolution, which has been extensively surveyed (Gämperle et al. 2002; Lampinen and Zelinka 2000; Price et al. 2005; Zaharie 2003). In order to reduce the risk of occurring premature convergence during evolution, a stagnation breaking scheme embedded in DE would be able to improve its performance.

Towards this direction, a DE variant with time-frame strategy adaptation called the time-frame adaptive differential evolution (TFADE) algorithm is proposed. In TFADE, several specific trial vector generation strategies and their control parameter values diverse and restricted are embedded. It can be gradually adapted to select suitable strategies to produce promising solutions according to successful experience over a period of preceding generations. Besides, a stagnation breaking scheme possessing extreme features of control parameter settings is also introduced to deal with the situation of premature convergence or stagnation during evolution. The idea of TFADE is described as follows.

3.1 The time-frame strategy candidate pool

A time-frame strategy candidate pool containing several diverse and effective strategies is maintained. Many different characteristics of trial vector generation strategies have been vastly investigated (Mallipeddi et al. 2011; Qin et al. 2009; Yong et al. 2011). Putting more strategies into the pool might cause unfavorable and ineffective influence, which can reduce the chance of generating promising offspring. A good candidate pool should be restricted, but exhibits effective and diverse characteristics. In TFADE, several trial vector generation strategies with effective and diverse characteristics are selected into the time-frame strategy candidate pool, as follows.

“DE/rand/1/bin” is the most commonly used trial vector generation strategy in DE literature. It is robust and has no bias to any search direction that can demonstrate various searches in a random manner. Its extension, “DE/rand/2/bin”, can lead to a better perturbation improving diversity and generate a wide variety of trial vectors because two difference vectors are added to the base vector (Gämperle et al. 2002; Storn and Price 1997). They are robust, diverse and can provide broad exploration capability to any problem. Especially it is suitable for solving multimodal problems (Qin et al. 2009). For these reasons, both strategies are selected into the time-frame strategy candidate pool.

“DE/current-to-rand/1” is a rotation-invariant strategy, which is kind of special than others and more effective for rotated problems. It commonly uses the rotation-invariant arithmetic crossover rather than the binomial to generate trial

vectors (Das et al. 2009; Price 1999). As its effectiveness and robustness, especially to the rotated problems, it is selected into the pool.

The strategies of “DE/best/1/bin”, “DE/best/2/bin”, “DE/rand-to-best/1/bin”, and “DE/rand-to-best/2/bin” employ the information of the best individual found so far. They are suitable for solving unimodal problems but less effective to highly multimodal problems, especially they are easier to get stuck at local optimum. For these reasons, we do not consider these strategies putting into the pool.

All strategies selected into the time-frame strategy candidate pool are shown in Table 1. Due to the popularity in DE research, the binomial-type crossover operation is employed to the strategies in the pool except the strategy of “DE/current-to-rand/1”.

3.2 The parameter candidate pool

Two kinds of parameter candidate pools are maintained in TFADE. One is called the time-frame parameter candidate pool; the other the extreme parameter candidate pool. Some empirical guidelines in choosing suitable parameter values for various problems also have been investigated. A detailed examination of important guidelines will lead

us to select a good combination of parameters into the pool.

Storn and Price (1997) suggested that a reasonable value for the population size NP should be between 5D and 10D. A good initial choice of F is 0.5 and the effective range of F values is suggested between 0.4 and 1. If the population converges prematurely, either F or NP can be increased. First reasonable attempt of choosing CR value can be 0.1. However, because the larger CR value can speed up the convergence, the value of 0.9 for CR may be a good initial choice if the problem is near unimodal or fast convergence is desired. Gämperle et al. (2002) examined different parameter settings for DE on Sphere, Rosenbrock, and Rastrigin functions. Their experimental results showed that the searching capability and convergence speed are very sensitive to the choice of NP, F and CR. They recommended that the population size NP should be between 3D and 8D, the scaling factor F equals to 0.6, and the crossover rate CR should be between 0.3 and 0.9. Ronkkonen et al. (2005) suggested that the population size NP should be between 2D and 4D. Using F values between 0.4 and 0.95, and F = 0.9 as an initial choice is a good way for DE. CR values should lie in [0, 0.2] for separable functions and in [0.9, 1] for multimodal and non-separable functions. Runarsson and Xin (2000) showed that although it has been

Table 1 Three trial vector generation strategies in the time-frame strategy candidate pool

“DE/rand/1/bin”:

$$\mathbf{u}_{i,G} = u_{ij,G} = \begin{cases} x_{r1j,G} + F(x_{r2j,G} - x_{r3j,G}), & \text{if } (\text{rand}_j \leq \text{CR}) \text{ or } (j = j_{\text{rand}}) \\ x_{ij,G}, & \text{otherwise} \end{cases}$$

“DE/rand/2/bin”:

$$\mathbf{u}_{i,G} = u_{ij,G} = \begin{cases} x_{r1j,G} + F(x_{r2j,G} - x_{r3j,G}) + F(x_{r4j,G} - x_{r5j,G}), & \text{if } (\text{rand}_j \leq \text{CR}) \text{ or } (j = j_{\text{rand}}) \\ x_{ij,G}, & \text{otherwise} \end{cases}$$

“DE/current-to-rand/1”:

$$\mathbf{u}_{i,G} = \mathbf{x}_{i,G} + K(\mathbf{x}_{r1,G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G})$$

Table 2 Guidelines for choosing control parameter values

Researches	NP	F	CR
Storn and Price (1997)	[5D, 10D]	Init.: 0.5 [0.4, 1]	Init.: 0.1 or 0.9
Gämperle et al. (2002)	[3D, 8D]	0.6	[0.3, 0.9]
Ronkkonen et al. (2005)	[2D, 4D]	Init.: 0.9 [0.4, 0.95]	Unimodal: [0, 0.2] Multimodal: [0.9, 1]
Mezura-Montes et al. (2003)	–	[0.3, 1]	–
Montgomery and Chen (2010)	–	–	Low and high values are effective; around 0.5 are not
Coello Coello (2002)	–	–	0.9
Yong et al. (2011)	–	0.8, 1.0	0.1, 0.2, 0.9
Mallipeddi et al. (2011)	–	–	Unimodal: low values Multimodal: high values

Fig. 1 The process of TFADE

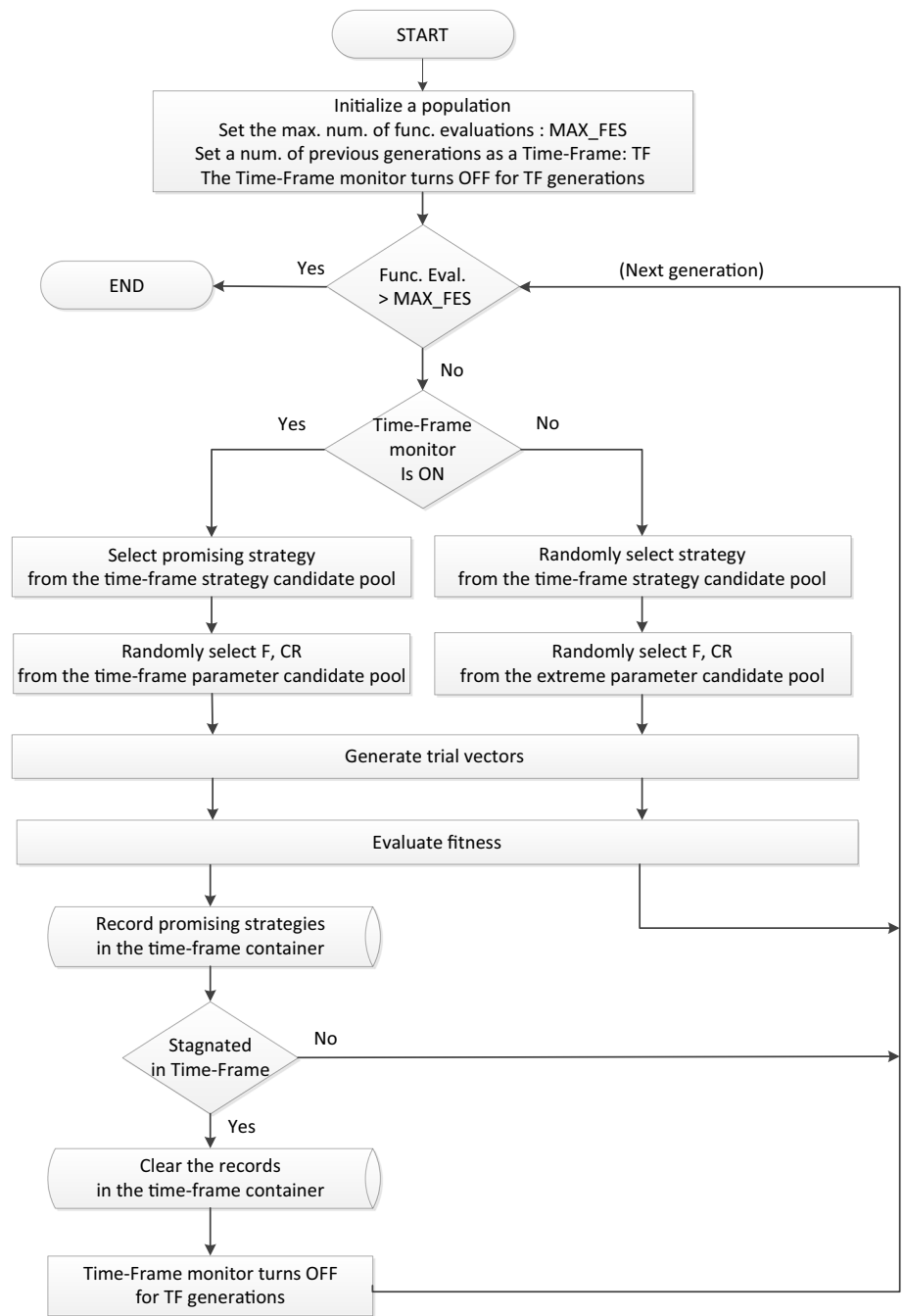


Fig. 2 A time-frame container of size TF

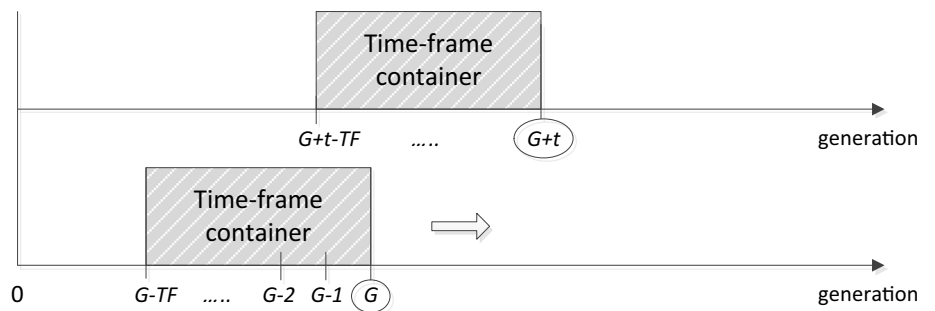


Table 3 Algorithmic description of TFADE

```

1  Input
2  randomly initialize a population of  $NP$  target vectors ;
3  the max. number of function evaluations ( $FES$ ) =  $MAX\_FES$  ;
4  the time-frame strategy candidate pool :  $S = \{ "DE/rand/1/bin",$ 
      " $DE/rand/2/bin$ ", " $DE/current-to-rand/1$ " } ;
5  the time-frame parameter candidate pool :  $Pt = \{ (F, CR) | \text{a number of } F \text{ and}$ 
       $CR \text{ within } [0.3, 1] \text{ and } [0.1, 0.9] \text{ in an equal step respectively } \}$  ;
6  the extreme parameter candidate pool :  $Pe = \{ (F, CR) \} = \{ (1, 0.1), (1, 0.9) \}$  ;
7  the time-frame MONITOR sets OFF for  $TF$  generations ;
8  generation=0 ;  $FES=0$  ;
9  do {
10     generation++ ;
11     if MONITOR = ON
12         select a promising strategy from  $S$  ;
13         randomly select  $F$  and  $CR$  from  $Pt$  ;
14         generate trial vectors ;
15         evaluate fitness ;
16         record data ;
17         if stagnated for  $TF$  generations
18             clear all recording data ;
19             MONITOR sets OFF for  $TF$  generations ;
20         end
21     else
22         randomly select a strategy from  $S$  ;
23         randomly select  $F$  and  $CR$  from  $Pe$  ;
24         generate trial vectors ;
25         evaluate fitness ;
26     end
27      $FES = FES + NP$  ;
28 } while ( $FES < MAX\_FES$ )

```

found that $F \in [0.3, 1]$ can be effective (Mezura-Montes and Coello Coello 2003), the use of a large population in conjunction with large F will cause the algorithm to converge too slowly to produce a good result by the end of its run.

Montgomery and Chen (2010) pointed out an analysis of DE why low and high values of CR are effective while values around 0.5 are not. At its extremes, CR leads to vastly different search behaviors. Low values of CR result in a search that is not just aligned with a small number of search space axes, but is gradual, slow and robust. High values of CR result in a search where fewer solutions generated may be improving, but the improvements can be larger. Coello Coello (2002) showed that a key parameter that affects DE's performance is the crossover rate CR. The crossover rate $CR = 0.9$ has been found to work well across a large range of problem domains. It is often used as a default value in many DE implementations. Yong et al. (2011) showed that three trial vector generation strategies, "rand/1/bin", "rand/2/bin", and "current-to-rand/1", and three control parameter settings,

$[F = 1.0, CR = 0.1]$, $[F = 1.0, CR = 0.9]$, and $[F = 0.8, CR = 0.2]$ are frequently used in many DE variants, the properties of which have been well studied. Mallipeddi et al. (2011) suggested that separable unimodal problems require lower CR values while parameter-linked multimodal problems require higher CR values. These guidelines are summarized in Table 2.

Based on these investigated guidelines, as the parameter values should be restricted to enhance the speed and effectiveness, and with diverse characteristics for solving various problems; therefore, choosing m of F and CR values within the range of $[0.3, 1]$ and $[0.1, 0.9]$ in an equal step, respectively, nearly covered above-mentioned guidelines, is considered putting into the time-frame parameter candidate pool.

However, once premature convergence or stagnation occurred during evolution, some extraordinary parameter values need to be employed instead to ruffle the situation. From these guidelines, it is shown that F is closely related to convergence speed. The higher F value can avoid pre-

Table 4 Experimental results of TFADE with conventional DEs on D10 functions

Fcn.	TFADE (D10)		Rand/1/bin ($F = 0.9CR = 0.1$)		
	Mean	Std	Mean	Std	
<i>f</i> 1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
<i>f</i> 2	0.0000E+00	0.0000E+00	7.1386E-01	2.9891E-01	-
<i>f</i> 3	9.9954E-11	5.3051E-10	9.0394E+05	4.3160E+05	-
<i>f</i> 4	0.0000E+00	0.0000E+00	1.2506E+01	7.7841E+00	-
<i>f</i> 5	0.0000E+00	0.0000E+00	3.2065E-01	5.6331E-01	-
<i>f</i> 6	0.0000E+00	0.0000E+00	1.2447E+00	9.6495E-01	-
<i>f</i> 7	3.2424E-02	2.2547E-02	3.4254E-01	6.9934E-02	-
<i>f</i> 8	2.0251E+01	1.4636E-01	2.0374E+01	6.0727E-02	-
<i>f</i> 9	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
<i>f</i> 10	5.4059E+00	2.2580E+00	1.3940E+01	3.0646E+00	-
<i>f</i> 11	5.4774E-01	8.3834E-01	5.6154E+00	7.2540E-01	-
<i>f</i> 12	3.3343E+00	4.7961E+00	2.7603E+02	1.5677E+02	-
<i>f</i> 13	6.1659E-01	8.8993E-02	3.5026E-01	9.6068E-02	+
<i>f</i> 14	3.0512E+00	3.4065E-01	3.3561E+00	1.7929E-01	-
<i>f</i> 15	9.9110E+01	1.6003E+02	6.1218E+00	1.0980E+01	+
<i>f</i> 16	1.0308E+02	6.2399E+00	1.3553E+02	1.1132E+01	-
<i>f</i> 17	1.1308E+02	1.2019E+01	1.5750E+02	1.3716E+01	-
<i>f</i> 18	5.1667E+02	2.5200E+02	7.0196E+02	1.4335E+02	-
<i>f</i> 19	4.3333E+02	2.2489E+02	7.0298E+02	1.3141E+02	-
<i>f</i> 20	4.8333E+02	2.4507E+02	7.0734E+02	1.1620E+02	-
<i>f</i> 21	5.0000E+02	9.0972E+01	4.9029E+02	8.4839E+01	-
<i>f</i> 22	6.9718E+02	1.5874E+02	7.3799E+02	1.3649E+02	-
<i>f</i> 23	6.0013E+02	9.2879E+01	5.6275E+02	4.9645E+01	+
<i>f</i> 24	2.0000E+02	0.0000E+00	2.0000E+02	1.2496E-05	-
<i>f</i> 25	3.7823E+02	3.2321E+00	3.8931E+02	3.3775E+00	-
+			3		
=			2		
-			20		
%adv			87 %		

Fcn.	Rand/1/bin ($F = 0.9 CR = 0.9$)			Rand-to-best/1/bin ($F = 0.9, CR = 0.1$)		
	Mean	Std		Mean	Std	
<i>f</i> 1	8.8063E-14	1.1328E-13	-	0.0000E+00	0.0000E+00	=
<i>f</i> 2	1.9801E-06	1.6602E-06	-	2.5507E-02	1.6573E-02	-
<i>f</i> 3	8.0037E+00	8.9668E+00	-	6.6890E+05	2.9375E+05	-
<i>f</i> 4	3.1539E-05	2.4257E-05	-	1.5316E+00	9.6998E-01	-
<i>f</i> 5	1.7675E-10	1.4893E-10	-	5.4501E-05	1.4513E-04	-
<i>f</i> 6	5.8404E-04	4.1872E-04	-	2.2762E+00	2.7233E+00	-
<i>f</i> 7	2.8356E-01	2.2547E-01	-	1.9864E-01	5.5191E-02	-
<i>f</i> 8	2.0361E+01	8.0029E-02	-	2.0331E+01	7.0380E-02	-
<i>f</i> 9	7.2278E+00	4.9349E+00	-	0.0000E+00	0.0000E+00	=
<i>f</i> 10	1.4461E+01	9.9643E+00	-	1.2218E+01	3.0397E+00	-
<i>f</i> 11	4.3800E+00	3.0166E+00	-	5.6032E+00	6.7710E-01	-
<i>f</i> 12	8.5177E+00	3.8677E+01	-	1.8367E+02	1.0542E+02	-
<i>f</i> 13	1.4638E+00	9.9716E-01	-	3.0161E-01	9.4975E-02	+
<i>f</i> 14	3.3894E+00	5.3696E-01	-	3.2168E+00	2.3044E-01	-

Table 4 continued

Fcn.	Rand/1/bin ($F = 0.9$ CR = 0.9)			Rand-to-best/1/bin ($F = 0.9$, CR = 0.1)		
	Mean	Std		Mean	Std	
<i>f</i> 15	2.8252E+02	1.7376E+02	–	2.0063E+00	1.0900E+01	+
<i>f</i> 16	1.2166E+02	2.7222E+01	–	1.1896E+02	8.9088E+00	–
<i>f</i> 17	1.5748E+02	3.2811E+01	–	1.4478E+02	1.3430E+01	–
<i>f</i> 18	4.0000E+02	2.0342E+02	+	6.0956E+02	2.2613E+02	–
<i>f</i> 19	3.5000E+02	1.5256E+02	+	5.5740E+02	2.2734E+02	–
<i>f</i> 20	3.6667E+02	1.7287E+02	+	6.4282E+02	2.1353E+02	–
<i>f</i> 21	5.0000E+02	2.7465E–13	=	5.0775E+02	1.2594E+02	–
<i>f</i> 22	7.5792E+02	8.6805E+01	–	6.8179E+02	1.8873E+02	=
<i>f</i> 23	5.5947E+02	1.5457E–12	+	5.8653E+02	6.1928E+01	=
<i>f</i> 24	2.0000E+02	0.0000E+00	=	2.0214E+02	1.1702E+01	–
<i>f</i> 25	3.8360E+02	2.6908E+00	–	3.8716E+02	3.2921E+00	–
	4			4		
	2			2		
	19			19		
	83 %			83 %		
Fcn.	Current-to-rand/1 ($F = 0.9$, CR = 0.1)			Rand/2/bin ($F = 0.9$ CR = 0.1)		
	Mean	Std		Mean	Std	
<i>f</i> 1	0.0000E+00	0.0000E+00	=	0.0000E+00	0.0000E+00	=
<i>f</i> 2	0.0000E+00	0.0000E+00	=	1.9069E+00	8.7628E–01	–
<i>f</i> 3	2.2042E–25	5.0775E+02	+	1.0356E+06	4.1549E+05	–
<i>f</i> 4	0.0000E+00	0.0000E+00	=	2.0916E+01	9.4511E+00	–
<i>f</i> 5	3.9836E–11	5.5378E–11	–	3.1246E+00	2.7061E+00	–
<i>f</i> 6	1.3289E–01	7.2785E–01	–	2.0484E+00	1.2726E+00	–
<i>f</i> 7	3.0179E–02	2.9212E–02	=	3.9397E–01	9.5693E–02	–
<i>f</i> 8	2.0355E+01	7.2572E–02	–	2.0364E+01	6.2087E–02	–
<i>f</i> 9	4.8441E+00	2.1977E+00	–	0.0000E+00	0.0000E+00	=
<i>f</i> 10	4.5436E+00	2.2580E+00	=	1.4713E+01	2.8859E+00	–
<i>f</i> 11	2.0885E–01	5.7409E–01	+	5.6189E+00	7.1365E–01	–
<i>f</i> 12	5.5198E+01	2.8394E+02	–	2.7380E+02	1.2454E+02	–
<i>f</i> 13	1.2271E+00	4.8736E–01	–	4.1024E–01	6.9666E–02	+
<i>f</i> 14	3.1866E+00	2.2304E–01	–	3.3007E+00	1.7795E–01	–
<i>f</i> 15	2.3939E+02	1.3448E+02	–	2.3931E+01	1.8123E+01	+
<i>f</i> 16	9.8863E+01	6.0044E+00	+	1.4243E+02	9.1794E+00	–
<i>f</i> 17	1.0030E+02	5.6604E+00	=	1.6416E+02	1.5041E+01	–
<i>f</i> 18	5.3333E+02	2.5371E+02	–	6.7054E+02	1.4087E+02	–
<i>f</i> 19	4.3333E+02	2.2489E+02	=	6.3538E+02	1.3683E+02	–
<i>f</i> 20	4.1667E+02	2.1509E+02	=	6.4797E+02	1.4006E+02	–
<i>f</i> 21	4.9333E+02	3.6515E+01	=	4.5213E+02	8.7733E+01	+
<i>f</i> 22	7.2051E+02	1.4266E+02	=	7.3670E+02	1.3915E+02	–
<i>f</i> 23	5.6486E+02	2.9532E+01	=	5.4692E+02	3.8484E+01	+
<i>f</i> 24	2.0000E+02	0.0000E+00	=	2.0002E+02	3.4156E–02	–
<i>f</i> 25	3.7419E+02	2.7879E+00	+	3.9224E+02	4.7261E+00	–
	3			4		
	12			2		
	10			19		
	77 %			83 %		

Table 5 Experimental results of TFADE with conventional DEs on D30 functions

Fcn.	TFADE (D30)		Rand/1/bin ($F = 0.9$ CR = 0.1)		
	Mean	Std	Mean	Std	
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
f_2	1.1986E-13	2.2297E-13	3.2067E+03	5.8949E+02	-
f_3	9.6157E+04	6.0922E+04	2.6029E+07	5.5824E+06	-
f_4	4.4048E-03	7.2269E-03	1.1882E+04	2.1995E+03	-
f_5	9.9731E+02	5.9929E+02	4.6671E+03	4.8732E+02	-
f_6	1.3655E+00	1.7700E+00	3.9356E+01	1.8469E+01	-
f_7	1.6656E-02	9.7616E-03	2.9366E-01	5.1647E-02	-
f_8	2.0780E+01	2.8213E-01	2.0953E+01	4.4810E-02	-
f_9	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	=
f_{10}	3.4689E+01	9.6632E+00	1.5406E+02	1.1681E+01	-
f_{11}	9.6896E+00	2.6178E+00	3.0179E+01	1.4498E+00	-
f_{12}	4.4116E+03	6.0927E+03	4.8701E+04	9.0748E+03	-
f_{13}	4.5088E+00	5.1959E-01	3.8104E+00	2.6284E-01	+
f_{14}	1.2581E+01	3.3822E-01	1.3010E+01	1.7126E-01	-
f_{15}	4.3013E+02	7.0357E+01	1.0697E+02	5.9078E+01	+
f_{16}	6.1639E+01	1.9447E+01	2.3032E+02	1.6195E+01	-
f_{17}	7.1491E+01	1.8618E+01	2.7011E+02	2.0706E+01	-
f_{18}	9.0541E+02	1.2457E+00	9.0990E+02	1.0930E+00	-
f_{19}	8.9798E+02	2.6662E+01	9.0968E+02	1.0138E+00	-
f_{20}	9.0533E+02	1.5492E+00	9.0989E+02	9.2508E-01	-
f_{21}	5.0000E+02	2.3126E-13	5.0000E+02	2.8363E-13	-
f_{22}	8.9168E+02	1.7764E+01	9.5867E+02	8.4360E+00	-
f_{23}	5.3425E+02	4.7011E-01	5.3416E+02	1.3528E-04	=
f_{24}	2.0000E+02	9.4982E-13	2.0000E+02	5.7188E-13	=
f_{25}	2.1043E+02	7.1642E-01	2.2607E+02	2.2325E+00	-
+			2		
=			4		
-			19		
%adv			90 %		

Fcn.	Rand/1/bin ($F = 0.9$ CR = 0.9)			Rand-to-best/1/bin ($F = 0.9$, CR = 0.1)		
	Mean	Std		Mean	Std	
f_1	4.1850E-03	4.7372E-03	-	0.0000E+00	0.0000E+00	=
f_2	3.8241E+02	2.1228E+02	-	1.4359E+03	2.9626E+02	-
f_3	5.9732E+06	2.7505E+06	-	2.4456E+07	6.0641E+06	-
f_4	2.3886E+03	1.6100E+03	-	7.1697E+03	1.5615E+03	-
f_5	1.1064E+03	3.9899E+02	-	3.9247E+03	5.8725E+02	-
f_6	6.2485E+01	5.8268E+01	-	2.8513E+01	1.4595E+01	-
f_7	1.0071E+00	9.5931E-02	-	1.4229E-01	4.5337E-02	-
f_8	2.0933E+01	5.0823E-02	-	2.0948E+01	4.8053E-02	-
f_9	3.6629E+01	1.5555E+01	-	0.0000E+00	0.0000E+00	=
f_{10}	2.3510E+02	1.4550E+01	-	1.3612E+02	1.6125E+01	-
f_{11}	3.9054E+01	1.2541E+00	-	2.9369E+01	1.4169E+00	-
f_{12}	7.7481E+03	5.5238E+03	-	3.7040E+04	9.3873E+03	-
f_{13}	1.0984E+01	3.7301E+00	-	3.6831E+00	3.7603E-01	+
f_{14}	1.3461E+01	1.3854E-01	-	1.2993E+01	2.0201E-01	-

Table 5 continued

Fcn.	Rand/1/bin ($F = 0.9$ CR = 0.9)			Rand-to-best/1/bin ($F = 0.9$, CR = 0.1)		
	Mean	Std		Mean	Std	
f_{15}	3.6766E+02	8.6483E+01	+	8.1528E+01	7.5353E+01	+
f_{16}	2.5098E+02	1.3036E+01	-	1.6746E+02	2.5790E+01	-
f_{17}	2.8699E+02	2.1748E+01	-	2.3560E+02	2.7995E+01	-
f_{18}	9.0608E+02	7.9527E-01	-	9.0811E+02	7.3733E-01	-
f_{19}	9.0634E+02	6.8217E-01	-	9.0833E+02	8.8879E-01	-
f_{20}	9.0618E+02	6.6252E-01	-	9.0829E+02	8.8833E-01	-
f_{21}	5.0000E+02	8.2477E-04	=	5.0000E+02	2.3054E-13	=
f_{22}	9.1211E+02	1.2002E+01	-	9.4221E+02	1.3540E+01	-
f_{23}	5.3416E+02	4.6787E-04	=	5.3416E+02	2.0700E-04	=
f_{24}	2.0000E+02	1.1555E-03	=	2.0000E+02	1.0782E-12	=
f_{25}	2.1285E+02	1.5246E+00	-	2.2176E+02	2.1537E+00	-
1				2		
3				5		
21				18		
95 %				90 %		
Fcn.	Current-to-rand/1 ($F = 0.9$, CR = 0.1)			Rand/2/bin ($F = 0.9$ CR = 0.1)		
	Mean	Std		Mean	Std	
f_1	1.2878E+04	5.5219E+03	-	0.0000E+00	0.0000E+00	=
f_2	1.3956E+04	4.7027E+03	-	4.1963E+03	6.7195E+02	-
f_3	7.5583E+06	4.4230E+06	-	2.8660E+07	6.6460E+06	-
f_4	1.5370E+04	5.0916E+03	-	1.3450E+04	2.5224E+03	-
f_5	8.2942E+03	1.7565E+03	-	4.8911E+03	4.4267E+02	-
f_6	1.0322E+09	9.6315E+08	-	4.2348E+01	1.6990E+01	-
f_7	3.4087E+03	1.1053E+03	-	4.9474E-01	9.0619E-02	-
f_8	2.0928E+01	6.0418E-02	-	2.0944E+01	6.2024E-02	-
f_9	1.2661E+02	1.7753E+01	-	0.0000E+00	0.0000E+00	=
f_{10}	1.6128E+02	2.6821E+01	-	1.5946E+02	1.4810E+01	-
f_{11}	2.3837E+01	3.0731E+00	-	2.9770E+01	1.3873E+00	-
f_{12}	1.2836E+05	4.0425E+04	-	5.2851E+04	9.2869E+03	-
f_{13}	7.6610E+00	1.7127E+00	-	4.1885E+00	2.8813E-01	+
f_{14}	1.2692E+01	3.0330E-01	=	1.3077E+01	1.8181E-01	-
f_{15}	5.4802E+02	4.2621E+01	-	1.4040E+02	2.9469E+01	+
f_{16}	1.6277E+02	3.5246E+01	-	2.3387E+02	2.5746E+01	-
f_{17}	1.6225E+02	3.2720E+01	-	2.8621E+02	2.1928E+01	-
f_{18}	9.4668E+02	2.7105E+01	-	9.1034E+02	9.4438E-01	-
f_{19}	9.4427E+02	1.9698E+01	-	9.1042E+02	9.6452E-01	-
f_{20}	9.4567E+02	2.1965E+01	-	9.1041E+02	1.1092E+00	-
f_{21}	1.0197E+03	1.3982E+02	-	5.0000E+02	1.4240E-13	=
f_{22}	9.3441E+02	2.7572E+01	-	9.6000E+02	1.1262E+01	-
f_{23}	9.9244E+02	1.6612E+02	-	5.3416E+02	4.2828E-04	=
f_{24}	7.6608E+02	1.4658E+02	-	2.0000E+02	4.0539E-14	=
f_{25}	8.2111E+02	4.1262E+02	-	2.2911E+02	1.9157E+00	-
0				2		
1				5		
24				18		
100 %				90 %		

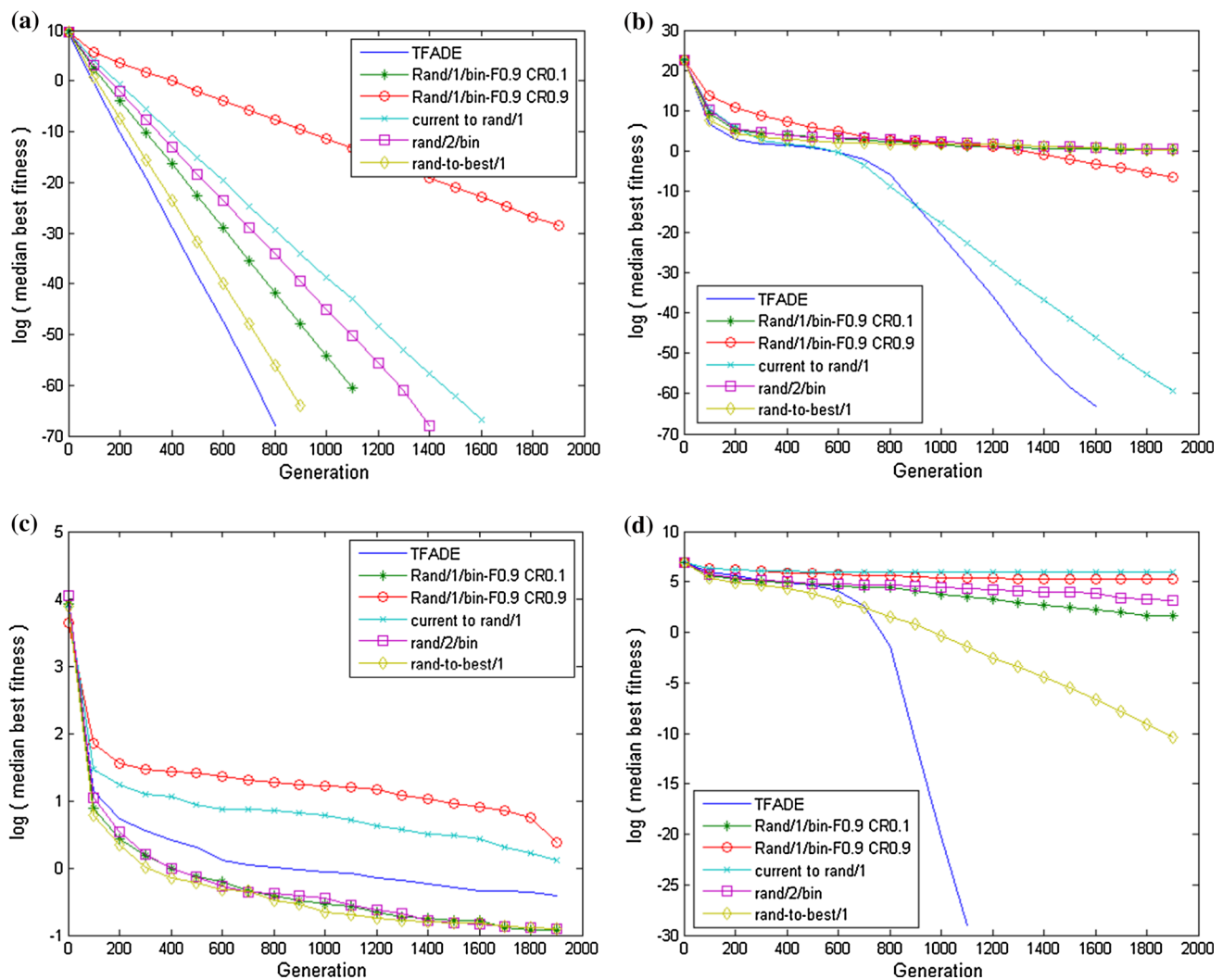


Fig. 3 The median convergence characteristics of TFADE with conventional DEs on D10 test functions of **a** the unimodal function of f_1 **b** the basic multimodal function of f_6 **c** the expanded multimodal function of f_{13} **d** the hybrid composition function of f_{15}

mature convergence or stagnation, while the lower F may increase convergence speed. CR is usually more sensitive to specific problems. Separable unimodal problems need lower CR value while multimodal ones require higher CR. Therefore, an extreme parameter candidate pool is developed that only extreme values of $F = \{1\}$ and $\text{CR} = \{0.1, 0.9\}$ are selected to increase the chance for escaping the stagnation. Thus, in the stagnation breaking scheme of TFADE, once the premature convergence occurred during evolution, the extreme parameter candidate pool will be enabled instead of the time-frame parameter candidate pool to deal with the difficulty. As for the parameter NP, it does not need to be fine-tuned because just a few typical values can be tried according to the complexity of a given problem (Qin et al. 2009). Therefore, it is left as a user-defined parameter.

3.3 The TFADE algorithm

The process of TFADE is shown in Fig. 1. The first important scheme is called the time-frame container, which maintains crucial information to be monitored during evolution, such as strategy/parameter candidate pools, the selection probability of strategies, and the best fitness of individuals (discussed later), over a given period of preceding generations called the time frame. Concerning each target vector in the current population, one trial vector generation strategy is selected from the time-frame strategy candidate pool in the container, according to the selection probability of the strategy which is calculated by generating promising solutions within the time frame. The selection probability of a strategy is defined as a normalized success rate that a trial vector generated by the strategy which can successfully enter next generation within

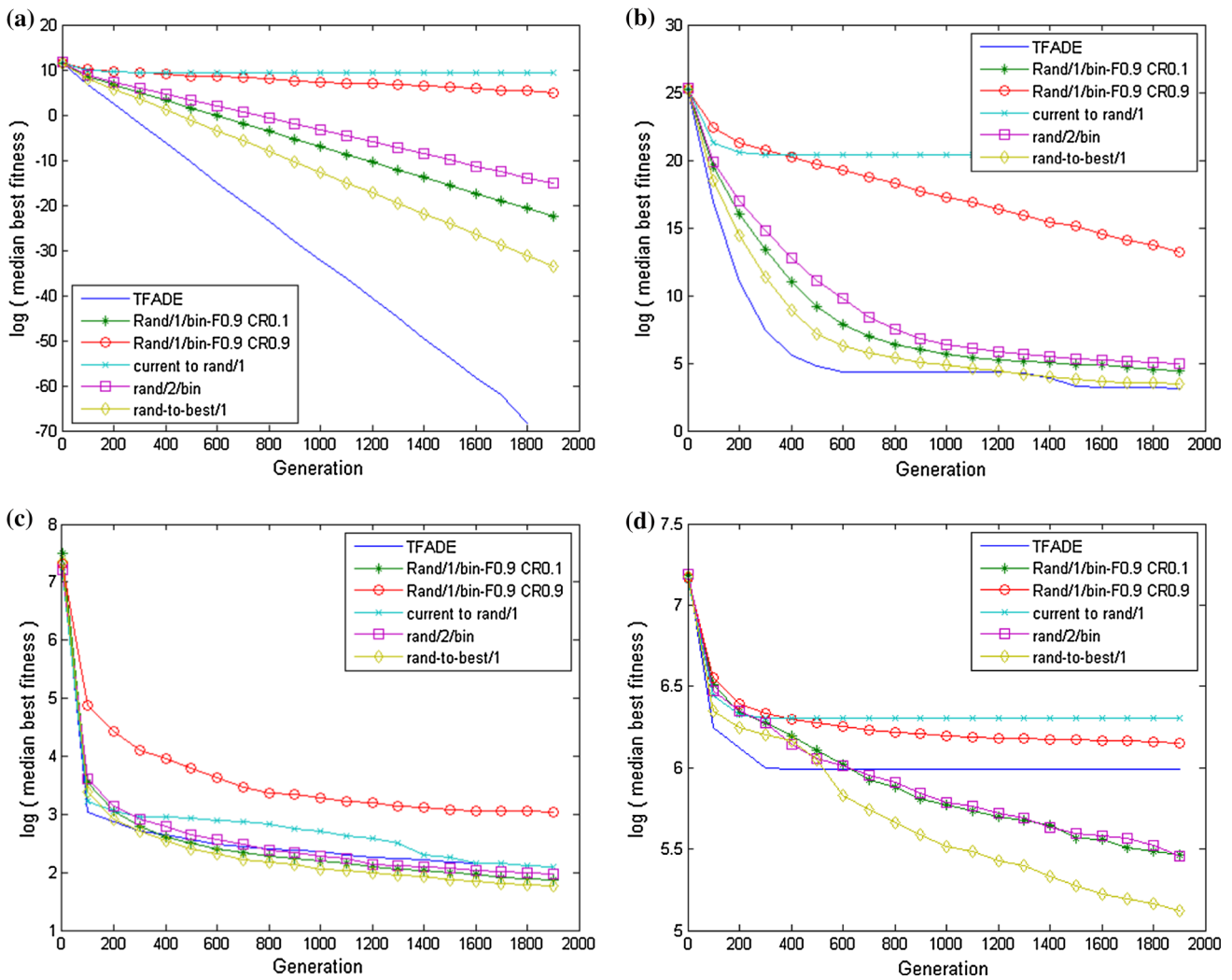


Fig. 4 The median convergence characteristics of TFADE with conventional DEs on D30 test functions of **a** the unimodal function of f_1 **b** the basic multimodal function of f_6 **c** the expanded multimodal function of f_{13} **d** the hybrid composition function of f_{15}

the time frame. During evolution, the time frame moves forward along time horizon of generations, the selection probability of each strategy and the best fitness of individuals can be gradually updated and stored in the container by experiencing different generations. A time-frame container of size TF from generation G evolving to $G+t$ is illustrated in Fig. 2.

More specifically, to the selection probability of a strategy, assume that we discuss a time frame of size TF and S strategies maintained in the time-frame strategy candidate pool. The selection probability with respect to each strategy for a target vector \mathbf{x}_i , where $i = 1, 2, \dots, NP$, in the initialized population is $1/S$, i.e., all strategies have the same probability to be selected at the beginning.

During evolution at current generation G after TF generations, the success rate of a strategy s that a trial vector \mathbf{u}_i generated by the strategy which can enter next generation can be calculated by

$$\text{SucRate}_{i,s,G} = \frac{\sum_{g=G-\text{TF}}^G \text{Success}_{i,s,g}}{\sum_{g=G-\text{TF}}^G \text{Selection}_{i,s,g}} + \varepsilon, \tag{10}$$

where $\text{Selection}_{i,s,g} = \{0, 1\}$ means whether the strategy s is selected to generate \mathbf{u}_i at the generation g , and $\text{Success}_{i,s,g} = \{0, 1\}$ stands for whether it is successful to enter next generation at generation g . $g = (G - \text{TF}), (G - \text{TF} + 1), \dots, G$ be the generations within the time frame. Note that a small value ε is placed to ensure nonzero success rate and that all strategies in the time-frame strategy candidate pool always have a chance to be selected.

Therefore, the selection probability of the strategy s at generation G for a target vector \mathbf{x}_i can be denoted as the normalized success rate of the strategy s as

$$\text{SelProb}_{i,s,G} = \frac{\text{SucRate}_{i,s,G}}{\sum_{s=1}^S \text{SucRate}_{i,s,G}}, \tag{11}$$

Table 6 Experimental results of TFADE with state-of-the-art DEs on D10 functions

Fcn.	TFADE (D10)			SaDE			EPSDE (LP50)			CoDE		
	Mean	Std		Mean	Std		Mean	Std		Mean	Std	
<i>f</i> 1	0.0000E+00	0.0000E+00		0.0000E+00	0.0000E+00	=	0.0000E+00	0.0000E+00	=	4.1505E-28	5.7129E-28	-
<i>f</i> 2	0.0000E+00	0.0000E+00		2.5244E-29	5.4263E-29	-	3.8707E-29	5.2507E-29	-	2.0748E-13	2.0412E-13	-
<i>f</i> 3	9.9954E-11	5.3051E-10		3.5097E+03	7.0375E+03	-	1.2460E+01	5.6609E+01	-	2.3848E-07	2.4077E-07	-
<i>f</i> 4	0.0000E+00	0.0000E+00		1.0098E-29	3.8428E-29	-	6.9420E-29	1.0606E-28	-	1.9252E-11	1.5298E-11	-
<i>f</i> 5	0.0000E+00	0.0000E+00		1.2127E-13	6.6420E-13	-	5.8782E+00	1.8869E+01	-	1.8438E-10	9.8575E-11	-
<i>f</i> 6	0.0000E+00	0.0000E+00		1.7323E+00	1.2970E+00	-	5.4403E-02	2.8723E-01	-	3.7258E-08	1.0716E-07	-
<i>f</i> 7	3.2424E-02	2.2547E-02		2.5365E-02	2.0306E-02	=	2.9909E-02	1.3913E-02	=	8.2854E-02	5.4072E-02	-
<i>f</i> 8	2.0251E+01	1.4636E-01		2.0342E+01	7.7987E-02	-	2.0382E+01	7.4526E-02	-	2.0341E+01	7.4911E-02	-
<i>f</i> 9	0.0000E+00	0.0000E+00		0.0000E+00	0.0000E+00	=	0.0000E+00	0.0000E+00	=	0.0000E+00	0.0000E+00	=
<i>f</i> 10	5.4059E+00	2.2580E+00		5.1075E+00	1.7279E+00	=	5.6135E+00	1.6884E+00	-	7.2175E+00	2.3677E+00	-
<i>f</i> 11	5.4774E-01	8.3834E-01		1.0095E+00	7.5414E-01	-	6.4746E+00	1.1845E+00	-	2.2954E+00	1.8463E+00	-
<i>f</i> 12	3.3343E+00	4.7961E+00		7.4760E+00	6.9442E+00	-	3.0421E+02	1.3568E+02	-	4.9475E+00	7.6742E+00	-
<i>f</i> 13	6.1659E-01	8.8993E-02		5.6518E-01	6.7346E-02	+	2.9863E-01	4.7283E-02	+	7.0221E-01	1.1024E-01	-
<i>f</i> 14	3.0512E+00	3.4065E-01		2.8333E+00	3.2456E-01	+	3.5476E+00	2.7797E-01	-	3.3161E+00	2.0303E-01	-
<i>f</i> 15	9.9110E+01	1.6003E+02		6.6502E+01	1.3560E+02	=	1.1452E+02	1.1718E+02	-	3.0836E+00	1.1834E+01	+
<i>f</i> 16	1.0308E+02	6.2399E+00		9.8027E+01	6.0009E+00	+	1.0262E+02	6.8159E+00	=	1.0348E+02	6.8432E+00	-
<i>f</i> 17	1.1308E+02	1.2019E+01		1.0088E+02	6.9149E+00	+	1.1654E+02	9.4381E+00	-	1.2934E+02	1.3539E+01	-
<i>f</i> 18	5.1667E+02	2.5200E+02		7.4606E+02	1.8153E+02	-	8.1281E+02	1.8263E+02	-	4.3333E+02	2.2489E+02	=
<i>f</i> 19	4.3333E+02	2.2489E+02		7.5000E+02	1.5256E+02	-	7.4315E+02	2.0515E+02	-	3.5000E+02	1.5256E+02	+
<i>f</i> 20	4.8333E+02	2.4507E+02		7.1236E+02	2.1301E+02	-	8.0140E+02	1.1265E+02	-	3.3333E+02	1.2685E+02	+
<i>f</i> 21	5.0000E+02	9.0972E+01		5.6333E+02	1.9384E+02	-	8.2968E+02	2.9532E+02	-	5.0000E+02	3.4689E-13	=
<i>f</i> 22	6.9718E+02	1.5874E+02		6.8201E+02	1.7427E+02	=	6.9367E+02	1.3879E+02	=	7.3431E+02	1.1813E+02	=
<i>f</i> 23	6.0013E+02	9.2879E+01		7.7085E+02	1.7467E+02	-	1.0284E+03	1.5829E+02	-	5.5947E+02	1.2935E-12	+
<i>f</i> 24	2.0000E+02	0.0000E+00		2.0000E+02	0.0000E+00	=	3.7356E+02	3.0768E+00	-	2.0000E+02	0.0000E+00	=
<i>f</i> 25	3.7823E+02	3.2321E+00		3.7816E+02	4.0345E+00	=	3.7877E+02	2.3074E+01	=	3.7809E+02	2.7298E+00	=
	+			4			1			4		
	=			7			6			5		
	-			14			18			16		
	%adv			78 %			95 %			80 %		

Table 7 Experimental results of TFADE with state-of-the-art DEs on D30 functions

Fcn.	TFADE (D30)		SaDE		EPSDE (LP50)		CoDE		
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	
<i>f</i> 1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.6829E-30	9.2176E-30	-	2.7492E-28	2.3453E-28
<i>f</i> 2	1.1986E-13	2.2297E-13	3.2780E-06	5.3257E-06	2.9037E-25	1.3555E-24	+	3.6643E-08	4.7126E-08
<i>f</i> 3	9.6157E+04	6.0922E+04	5.4255E+05	2.2676E+05	2.2052E+06	5.0891E+06	-	2.5500E+05	1.9148E+05
<i>f</i> 4	4.4048E-03	7.2269E-03	1.0070E+02	1.1113E+02	4.2239E+00	9.8869E+00	-	2.5844E-02	3.7700E-02
<i>f</i> 5	9.9731E+02	5.9929E+02	3.3539E+03	3.9709E+02	1.5000E+03	7.4917E+02	-	5.2683E+01	9.4293E+01
<i>f</i> 6	1.3655E+00	1.7700E+00	5.2653E+01	3.1266E+01	2.6579E-01	1.0114E+00	+	3.0467E+00	1.3380E+00
<i>f</i> 7	1.6656E-02	9.7616E-03	1.5261E-02	1.2526E-02	1.8295E-02	1.3897E-02	=	2.3815E-03	4.8167E-03
<i>f</i> 8	2.0780E+01	2.8213E-01	2.0923E+01	5.5560E-02	2.0952E+01	3.7629E-02	-	2.0931E+01	8.9877E-02
<i>f</i> 9	0.0000E+00	0.0000E+00	6.6331E-02	2.5243E-01	0.0000E+00	0.0000E+00	=	1.0196E-11	1.7043E-11
<i>f</i> 10	3.4689E+01	9.6632E+00	4.7061E+01	9.4178E+00	4.7214E+01	1.1871E+01	-	3.4118E+01	1.1303E+01
<i>f</i> 11	9.6896E+00	2.6178E+00	1.6883E+01	2.8396E+00	3.5205E+01	3.8488E+00	-	6.9831E+00	3.2580E+00
<i>f</i> 12	4.4116E+03	6.0927E+03	3.7031E+03	2.9060E+03	3.6472E+04	7.4106E+03	-	2.9893E+03	3.4508E+03
<i>f</i> 13	4.5088E+00	5.1959E-01	3.9954E+00	3.2051E-01	1.9129E+00	1.5845E-01	+	5.7041E+00	4.4011E-01
<i>f</i> 14	1.2581E+01	3.3822E-01	1.2657E+01	2.6839E-01	1.3499E+01	2.3833E-01	-	1.3014E+01	3.3999E-01
<i>f</i> 15	4.3013E+02	7.0357E+01	3.7391E+02	8.7211E+01	2.2193E+02	4.4542E+01	+	4.0667E+02	2.5371E+01
<i>f</i> 16	6.1639E+01	1.9447E+01	9.5640E+01	9.8850E+01	1.1277E+02	8.8584E+01	-	5.7618E+01	1.8814E+01
<i>f</i> 17	7.1491E+01	1.8618E+01	7.1142E+01	1.8305E+01	1.3511E+02	6.2829E+01	-	5.9264E+01	1.2751E+01
<i>f</i> 18	9.0541E+02	1.2457E+00	8.6548E+02	6.2444E+01	8.2156E+02	4.2945E+00	+	9.0384E+02	1.7021E-01
<i>f</i> 19	8.9798E+02	2.6662E+01	8.8686E+02	5.8334E+01	8.2125E+02	3.4979E+00	+	9.0386E+02	2.4743E-01
<i>f</i> 20	9.0533E+02	1.5492E+00	8.8146E+02	5.8781E+01	8.2275E+02	4.7779E+00	+	9.0380E+02	3.0503E-01
<i>f</i> 21	5.0000E+02	2.3126E-13	5.2206E+02	1.2082E+02	8.5024E+02	6.6183E+01	-	5.0000E+02	1.7568E-13
<i>f</i> 22	8.9168E+02	1.7764E+01	9.3611E+02	1.4906E+01	5.0748E+02	5.9335E+00	+	8.9695E+02	1.3602E+01
<i>f</i> 23	5.3425E+02	4.7011E-01	5.8950E+02	1.7228E+02	8.4695E+02	8.5074E+01	-	5.3416E+02	2.8768E-04
<i>f</i> 24	2.0000E+02	9.4982E-13	2.0000E+02	2.8908E-14	2.2250E+02	5.2430E+01	-	2.0000E+02	2.8908E-14
<i>f</i> 25	2.1043E+02	7.1642E-01	2.1281E+02	1.3258E+00	2.1344E+02	2.4402E+00	-	2.1043E+02	6.4797E-01
	+		4		8			5	
	=		6		2			8	
	-		15		15			12	
	%adv		79 %		65 %			71 %	

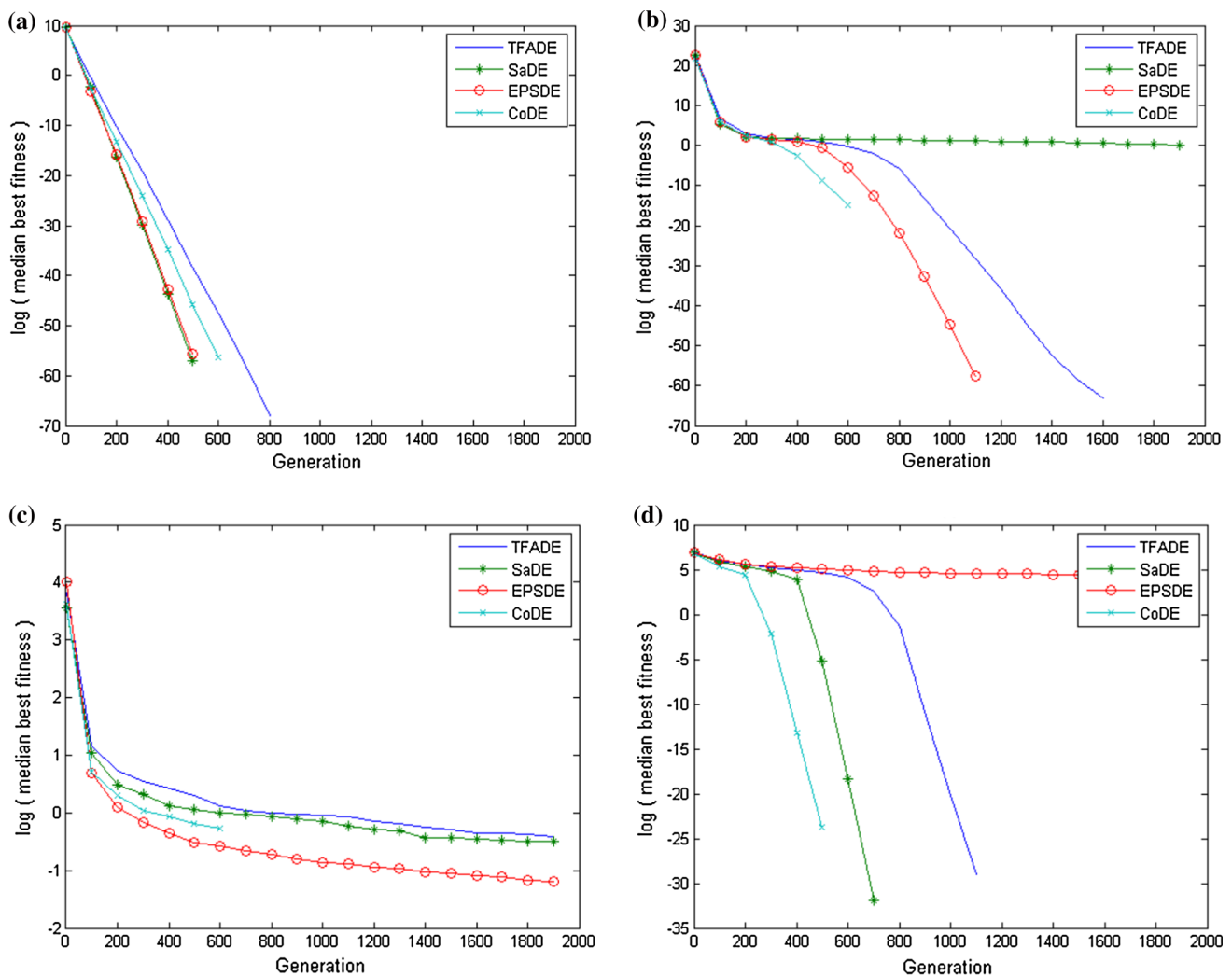


Fig. 5 The median convergence characteristics of TFADE with state-of-the-art adaptive DEs on D10 test functions of **a** the unimodal function of f_1 **b** the basic multimodal function of f_6 **c** the expanded multimodal function of f_{13} **d** the hybrid composition function of f_{15}

where strategy $s = 1, 2, \dots, S$. At this stage, the roulette wheel selection method is then employed to select a promising trial vector generation strategy for target vector \mathbf{x}_i .

After a trial vector generation strategy is selected, the values of its control parameters, F and CR , are then randomly chosen from the time-frame parameter candidate pool. Both the selected strategy and control parameter values are subsequently employed to corresponding target vector to generate a trial vector.

All these dynamic data are maintained in the time-frame container and gradually adapted during evolution. Once the records overflow over TF generations, the earliest record will be removed, keeping the most up-to-date records of TF generations in the container to be monitored.

The above procedure works as the typical evolution process of TFADE. However, during evolution once prema-

ture convergence or stagnation occurred throughout the time frame of size TF, it means that the typical evolution process in pursuit of promising solutions might be dysfunctional at this evolution stage. In order to escape this situation, the mechanism of the stagnation breaking scheme will immediately take control over the strategy/parameter selection. This mechanism destroys the current promising selection records in the container at once, stop monitoring and recording for TF generations, and randomly selecting strategies and parameters from the time-frame strategy candidate pool and the extreme parameter candidate pool, respectively for TF generations to ruffle the difficulty. After TF generations, the typical evolution process will then automatically switch on to adapt a better solution again in the current population. The algorithmic description of TFADE is illustrated in Table 3.

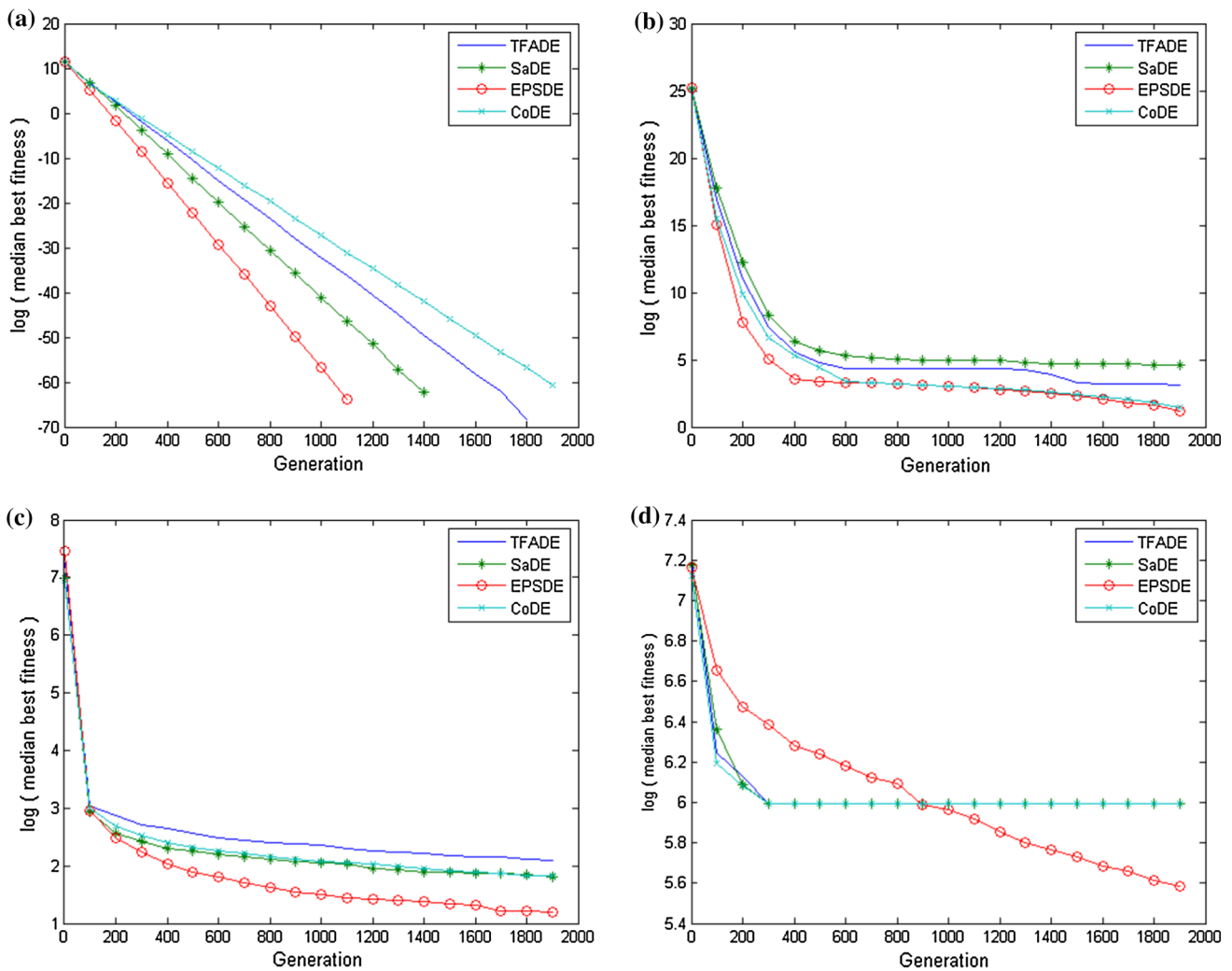


Fig. 6 The median convergence characteristics of TFADE with state-of-the-art adaptive DEs on D30 test functions of **a** the unimodal function of f_1 **b** the basic multimodal function of f_6 **c** the expanded multimodal function of f_{13} **d** the hybrid composition function of f_{15}

4 The experimental study

Three kinds of experiments were conducted in the experimental study. First, TFADE was compared with 4 commonly used conventional DEs coupled with widely suggested parameter settings. Next, with 3 outstanding state-of-the-art adaptive DEs proposed recently, i.e., SaDE (Qin et al. 2009), EPSDE (Mallipeddi et al. 2011), and CoDE (Yong et al. 2011). Last, 2 novel non-DE algorithms, i.e., CLPSO (Liang et al. 2006) and CMA-ES (Hansen and Ostermeier 2001), were also compared. To evaluate the performance of TFADE with these approaches, a test suite of benchmark functions (Suganthan et al. 2005) reported in 2005 IEEE Congress on Evolutionary Computation (CEC2005) was conducted. The test suite comprises 25 benchmark functions that can be divided into 4 groups:

1. 5 unimodal functions: $f_1 - f_5$,

2. 7 basic multimodal functions: $f_6 - f_{12}$,
3. 2 expanded multimodal functions: $f_{13} - f_{14}$,
4. 11 hybrid composition functions: $f_{15} - f_{25}$.

Both 10-dimensional (D10) and 30-dimensional (D30) of the benchmark functions were considered. 30 independent runs were conducted for each algorithm to each function. The population size NP was set to 50. The maximum number of function evaluations was set to multiplying function dimensions by 10,000 as the termination criteria, which are 100,000 for D10 functions and 300,000 for D30, respectively. The mean and standard deviation of the function error $f(x_{\text{best}}) - f(x^*)$ were calculated for measuring the performance of each algorithm to a function f , where x_{best} stands for the best solution of the function found by an algorithm in an independent run and x^* the global optimum of the function. The number of preceding generations for the time frame in TFADE was set to 10, and $m = 5$ was chosen for the

Table 8 Experimental results of TFADE with non-DE approaches on D10 functions

Fcn.	TFADE (D10)		CLPSO			CMA-ES		
	Mean	Std	Mean	Std		Mean	Std	
<i>f</i> 1	0.0000E+00	0.0000E+00	7.3678E−25	7.2245E−25	−	8.2473E−28	1.9406E−28	−
<i>f</i> 2	0.0000E+00	0.0000E+00	1.9571E−01	1.9433E−01	−	5.3996E−27	2.2468E−27	−
<i>f</i> 3	9.9954E−11	5.3051E−10	3.2308E+05	1.4269E+05	−	3.7344E−23	1.5679E−23	+
<i>f</i> 4	0.0000E+00	0.0000E+00	4.8630E+00	6.4251E+00	−	3.9035E+04	8.2610E+04	−
<i>f</i> 5	0.0000E+00	0.0000E+00	5.8506E+00	5.3746E+00	−	4.9113E−11	6.8728E−12	−
<i>f</i> 6	0.0000E+00	0.0000E+00	1.4583E+00	1.0769E+00	−	9.3020E−01	1.7150E+00	−
<i>f</i> 7	3.2424E−02	2.2547E−02	1.2670E+03	3.8000E−13	−	1.2067E−02	9.8299E−03	+
<i>f</i> 8	2.0251E+01	1.4636E−01	2.0362E+01	6.2865E−02	−	2.0003E+01	1.5515E−02	+
<i>f</i> 9	0.0000E+00	0.0000E+00	8.8818E−16	4.5408E−15	−	1.1780E+02	4.4885E+01	−
<i>f</i> 10	5.4059E+00	2.2580E+00	9.7514E+00	2.4955E+00	−	4.8487E+01	7.6068E+01	−
<i>f</i> 11	5.4774E−01	8.3834E−01	4.8648E+00	7.5520E−01	−	2.3323E+00	1.7622E+00	−
<i>f</i> 12	3.3343E+00	4.7961E+00	1.1731E+02	6.1480E+01	−	6.4601E+03	1.1014E+04	−
<i>f</i> 13	6.1659E−01	8.8993E−02	5.2624E−01	6.3763E−02	+	1.0885E+00	4.0723E−01	−
<i>f</i> 14	3.0512E+00	3.4065E−01	3.2388E+00	1.7283E−01	−	4.8339E+00	2.2808E−01	−
<i>f</i> 15	9.9110E+01	1.6003E+02	7.5398E+00	1.5844E+01	+	6.4309E+02	3.6453E+02	−
<i>f</i> 16	1.0308E+02	6.2399E+00	1.1653E+02	7.3206E+00	−	2.1094E+02	2.2024E+02	−
<i>f</i> 17	1.1308E+02	1.2019E+01	1.3505E+02	8.5211E+00	−	4.7987E+02	3.9998E+02	−
<i>f</i> 18	5.1667E+02	2.5200E+02	6.5008E+02	1.5316E+02	−	8.0270E+02	3.5166E+02	−
<i>f</i> 19	4.3333E+02	2.2489E+02	6.3734E+02	1.4854E+02	−	7.5181E+02	3.6851E+02	−
<i>f</i> 20	4.8333E+02	2.4507E+02	6.9029E+02	1.3510E+02	−	7.1652E+02	3.9735E+02	−
<i>f</i> 21	5.0000E+02	9.0972E+01	4.4291E+02	7.3646E+01	+	8.8728E+02	2.6814E+02	−
<i>f</i> 22	6.9718E+02	1.5874E+02	7.3579E+02	1.0162E+02	−	7.7471E+02	2.7702E+01	−
<i>f</i> 23	6.0013E+02	9.2879E+01	5.3894E+02	4.5506E+01	+	1.0873E+03	1.3708E+02	−
<i>f</i> 24	2.0000E+02	0.0000E+00	2.0000E+02	2.2742E−04	−	3.4667E+02	1.6965E+02	−
<i>f</i> 25	3.7823E+02	3.2321E+00	1.7487E+03	4.7715E+00	−	4.6196E+02	2.0570E+02	−
+			4			3		
=			0			0		
−			21			22		
%adv			84 %			88 %		

number of F and CR putting into the time-frame parameter candidate pool.

To present statistically significant difference between two algorithms, the statistical t test with a significance level of $\alpha = 0.05$ was performed on each pair of experimental results between TFADE and a compared algorithm on each function. The indicators of “+”, “=”, and “−” define that the out-

come of the corresponding algorithm is statistically better than, similar to, and worse than that of TFADE on a function, respectively. In order to show the overall excellence degree of TFADE against a competitor on the test suite of all 25 functions, we define a quality measurement called the advantage percentage of TFADE, denoted by %adv, as the equation (12).

$$\%adv = \frac{\text{num. of worse performance func. ("−")} - \text{num. of better performance func. ("+"})}{\text{total func. (25)} - \text{num. of similar performance func. ("=")}} \quad (12)$$

Table 9 Experimental results of TFADE with non-DE approaches on D30 functions

Fcn.	TFADE (D30)		CLPSO			CMA-ES		
	Mean	Std	Mean	Std		Mean	Std	
<i>f</i> 1	0.0000E+00	0.0000E+00	5.8455E−24	3.3523E−24	−	1.7678E−25	3.1240E−26	−
<i>f</i> 2	1.1986E−13	2.2297E−13	1.0182E+03	2.0127E+02	−	6.5127E−25	2.0394E−25	+
<i>f</i> 3	9.6157E+04	6.0922E+04	1.5216E+07	4.2249E+06	−	5.1462E−21	1.5288E−21	+
<i>f</i> 4	4.4048E−03	7.2269E−03	7.0268E+03	1.3502E+03	−	5.7489E+04	1.3241E+05	−
<i>f</i> 5	9.9731E+02	5.9929E+02	3.9057E+03	4.0797E+02	−	3.2770E−10	7.1499E−11	+
<i>f</i> 6	1.3655E+00	1.7700E+00	7.6528E+00	5.7193E+00	−	1.1960E+00	1.8581E+00	=
<i>f</i> 7	1.6656E−02	9.7616E−03	4.6964E+03	1.6972E−02	−	1.4780E−03	4.1207E−03	+
<i>f</i> 8	2.0780E+01	2.8213E−01	2.0939E+01	6.5672E−02	−	2.0194E+01	4.6028E−01	+
<i>f</i> 9	0.0000E+00	0.0000E+00	1.7408E−14	3.1761E−14	−	4.4818E+02	8.7924E+01	−
<i>f</i> 10	3.4689E+01	9.6632E+00	1.0404E+02	1.3044E+01	−	4.8123E+01	1.4132E+01	−
<i>f</i> 11	9.6896E+00	2.6178E+00	2.5309E+01	1.3404E+00	−	6.4331E+00	2.3835E+00	+
<i>f</i> 12	4.4116E+03	6.0927E+03	1.3404E+00	5.0041E+03	+	1.4658E+04	1.3738E+04	−
<i>f</i> 13	4.5088E+00	5.1959E−01	2.5910E+00	3.3296E−01	+	3.2799E+00	7.2171E−01	+
<i>f</i> 14	1.2581E+01	3.3822E−01	1.2665E+01	2.8821E−01	−	1.4736E+01	2.6255E−01	−
<i>f</i> 15	4.3013E+02	7.0357E+01	5.5254E+01	3.5407E+01	+	4.1337E+02	2.1633E+02	=
<i>f</i> 16	6.1639E+01	1.9447E+01	1.7222E+02	2.3516E+01	−	3.2533E+02	2.9275E+02	−
<i>f</i> 17	7.1491E+01	1.8618E+01	2.3387E+02	3.4092E+01	−	4.8539E+02	3.3290E+02	−
<i>f</i> 18	9.0541E+02	1.2457E+00	9.0158E+02	3.1314E+01	+	9.1650E+02	6.8482E+01	−
<i>f</i> 19	8.9798E+02	2.6662E+01	9.0430E+02	2.9665E+01	−	9.0394E+02	2.7591E−01	−
<i>f</i> 20	9.0533E+02	1.5492E+00	9.1206E+02	9.4690E+00	−	9.0399E+02	2.9426E−01	+
<i>f</i> 21	5.0000E+02	2.3126E−13	5.0000E+02	2.6932E−13	−	5.0000E+02	2.6665E−12	=
<i>f</i> 22	8.9168E+02	1.7764E+01	9.6457E+02	1.4170E+01	−	8.2366E+02	1.7569E+01	−
<i>f</i> 23	5.3425E+02	4.7011E−01	5.3416E+02	1.4753E−04	=	5.5012E+02	7.0156E+01	−
<i>f</i> 24	2.0000E+02	9.4982E−13	2.0000E+02	1.8902E−12	−	2.0000E+02	7.1005E−14	=
<i>f</i> 25	2.1043E+02	7.1642E−01	1.6620E+03	5.3338E+00	−	2.0804E+02	5.3688E+00	+
+			4			9		
=			1			4		
−			20			12		
%adv			83 %			57 %		

All these performance records were listed in the last 4 rows of experimental result tables for comparison.

4.1 Comparison with conventional DEs

Five test instances of four conventional DEs with fixed control parameter settings, “Rand/1/bin ($F = 0.9, CR = 0.1$)”, “Rand/1/bin ($F = 0.9, CR = 0.9$)”, “Rand-to-best/1/bin ($F = 0.9, CR = 0.1$)”, “Current-to-rand/1 ($F = 0.9, CR = 0.1$)” and “Rand/2/bin ($F = 0.9, CR = 0.1$)”, were compared with TFADE. The experimental results on D10 and D30 functions are illustrated in Tables 4 and 5, respectively. Figures 3 and 4 present the convergence characteristics of each test instance for 4 functions in terms of the median of the best fitness in 30 independent runs on D10 and D30 functions, respectively. The 4 functions, one is chosen from each

group, are (a) the unimodal function of f_1 , (b) the basic multimodal function of f_6 , (c) the expanded multimodal function of f_{13} , and (d) the hybrid composition function of f_{15} .

In the experimental results, obviously, TFADE is outstanding among these test instances on most functions of D10 and D30, especially on the groups of unimodal functions ($f_1 - f_5$) and basic multimodal functions ($f_6 - f_{12}$), because no one can be significantly better than TFADE on test functions of D10 and D30 except “Current-to-rand/1 ($F = 0.9, CR = 0.1$)” on 2 functions of D10 in the two groups. The advantage percentages of TFADE on the test suite to each are 87, 83, 83, 77, and 83 % for D10 functions and 90, 95, 90, 100, and 90 % for D30, respectively. Overall, the performance of TFADE is significantly better than that of these competitors; especially it outstandingly outperforms the competitors on both groups of the unimodal functions and basic multi-

Table 10 The advantage percentages of TFADE

	Rand/1/bin $F = 0.9$ CR = 0.1	Rand/1/bin $F =$ 0.9 CR = 0.9	Rand-to-best/ 1/bin $F = 0.9$ CR = 0.1	Current-to-rand/1 $F = 0.9$ CR = 0.1	Rand/2/bin $F = 0.9$ CR = 0.1	Average
(1) %adv to conventional DEs						
D10	87 %	83 %	83 %	77 %	83 %	82 %
D30	90 %	95 %	90 %	100 %	90 %	93 %
	SaDE	EPSDE (LP50)	CoDE	Average		
(2) %adv to state-of-the-art DEs						
D10	78 %	95 %	80 %	84 %		
D30	79 %	65 %	71 %	72 %		
	CLPSO	CMA-ES	Average			
(3) %adv to non-DE approaches						
D10	84 %	88 %	86 %			
D30	83 %	57 %	70 %			

modal functions, and on the functions with high dimensions (30-dimensional functions).

4.2 Comparison with state-of-the-art DEs

TFADE was also compared with three state-of-the-art adaptive DE variants proposed recently, i.e., SaDE, EPSDE, and CoDE. In these approaches, the control parameter settings were employed as in their original papers. The experimental results on D10 and D30 functions are presented in Tables 6 and 7, respectively. Figures 5 and 6 also illustrate the convergence characteristics of each approach for the 4 functions aforementioned in terms of the median of the best fitness in 30 independent runs on D10 and D30 functions, respectively.

Clearly, in the experimental results, TFADE also outperforms the state-of-the-art adaptive DE variants on all functions of D10 and D30 in the groups of the unimodal functions ($f_1 - f_5$) and basic multimodal functions ($f_6 - f_{12}$). No one is significantly better than TFADE on the test functions of D10 and D30 except EPSDE on 2 functions of D30 and CoDE on 2 functions of D30. The advantage percentages of TFADE on the test suite to each approach are 78, 95, and 80 % for D10 functions and 79, 65, and 71 % for D30, respectively. In summary, the performance of TFADE is greater than that of SaDE, EPSDE, and CoDE, especially on the groups of the unimodal functions and basic multimodal functions.

4.3 Comparison with non-DE approaches

Two novel non-DE algorithms, CLPSO and CMA-ES, were compared with TFADE in this experiment. CLPSO proposed by Liang et al. (2006) is an outstanding PSO variant, and CMA-ES proposed by Hansen and Ostermeier (2001) is a

very efficient and famous evolution strategy. Both algorithms with control parameter settings used in their original papers were conducted for comparison. The experimental results on D10 and D30 functions are shown in Tables 8 and 9, respectively.

In the experimental results, TFADE performs better than CLPSO and CMA-ES on 21 and 22 out of 25 functions of D10, and on 20 and 12 out of 25 functions of D30, respectively. The advantage percentages of TFADE to both are 84 and 88 % for D10 functions and 83 and 57 % for D30, respectively. In summary, TFADE has an absolute advantage than CLPSO and CMA-ES except that it has a slight advantage than CMA-ES on the functions with high dimensions (30-dimensional functions).

5 Conclusion

The performance of DE is extremely determined by the trial vector generation strategy and control parameter setting. The effectiveness of a strategy is also significantly influenced by its control parameter values. In recent years, DE researchers have reported many empirical guidelines in selecting suitable trial vector generation strategies with control parameter settings for solving various features of problems. The comprehensive exploration of the experience can be a good way to develop an advanced DE variant.

Towards this direction, in this paper, an improved DE variant with time-frame strategy adaptation called the time-frame adaptive differential evolution (TFADE) algorithm is proposed. It employs diverse trial vector generation strategies with various control parameter values that exhibit distinct characteristics from each other. Both strategies and control

parameter values can be adaptively determined to generate promising solutions and dynamically adjusted to deal with the difficulty while premature convergence or stagnation occurs during evolution, according to successful experience over a period of preceding generations called the time frame.

Although TFADE is simple and easy to be implemented, its performance is significantly outstanding. In the experimental study, the performance of TFADE was compared with that of 5 test instances of 4 commonly used conventional DEs, i.e., “Rand/1/bin ($F = 0.9$, $CR = 0.1$)”, “Rand/1/bin ($F=0.9$, $CR = 0.9$)”, “Rand-to-best/1/bin ($F = 0.9$, $CR = 0.1$)”, “Current-to-rand/1 ($F = 0.9$, $CR = 0.1$)” and “Rand/2/bin ($F = 0.9$, $CR = 0.1$)”, 3 outstanding state-of-the-art adaptive DEs, i.e., SaDE, EPSDE, and CoDE, and 2 novel non-DE approaches, i.e., CLPSO and CMA-ES, evaluated by a test suite of 25 benchmark functions reported in CEC2005. The experimental results show that its comprehensive performance is significantly superior to these competitors. The overall excellence degree of TFADE against these competitors, i.e., the advantage percentage (% adv), also shows that TFADE has an excellent advantage than these competitors. The advantage percentages of TFADE on the experimental results are summarized in Table 10.

Related work is considered for further research. From the literature review, we conclude that putting more strategies and parameter values into the strategy/parameter candidate pool will lead to an unfavorable and ineffective influence, which could reduce the performance of DE. Therefore, the strategies and parameter settings involved in the time-frame container should be restricted and preserve diversity. It is desirable to study whether the performance of TFADE can be significantly improved if the number of strategies and parameter values putting into the container can be dynamically adapted during evolution. What and how many strategies and parameter values should be combined can be more effective? Moreover, the discussion on the practical application of TFADE is also an important issue. In recent years, many distinguishing topics have been reported, such as the industrial applications on wireless sensor networks and RFID (Zhang et al. 2014a, b, 2015b), the web-based mobile application (Zhang 2012; Zhang and Liang 2013; Zhang et al. 2015a; Zhang and Zhang 2011), the image de-noising (Zhang et al. 2012b), and Internet of Things (Zhang et al. 2012a). The practical application of TFADE on these subjects is really worth investigating in the future.

Compliance with ethical standards

Conflict of interest The author declares that he has no conflict of interest in this work.

References

- Abbass HA (2002) The self-adaptive Pareto differential evolution algorithm. In: Proceedings of the 2002 congress on evolutionary computation, pp 831–836
- Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* 10(6):646–657
- Chakraborty UK, Das S, Konar A (2006) Differential evolution with local neighborhood. In: IEEE congress on evolutionary computation, pp 2042–2049
- Cheng M-Y, Tran D-H, Wu Y-W (2014) Using a fuzzy clustering chaotic-based differential evolution with serial method to solve resource-constrained project scheduling problems. *Autom Constr* 37:88–97
- Chiou JP (2009) A variable scaling hybrid differential evolution for solving large-scale power dispatch problems generation. *Transm Distrib* 3(2):154–163
- Coello Coello CA (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods Appl Mech Eng* 191(11–12):1245–1287
- Damak N, Jarbouli B, Siarry P, Loukil T (2009) Differential evolution for solving multi-mode resource-constrained project scheduling problems. *Comput Oper Res* 36(9):2653–2659
- Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. *IEEE Trans Evol Comput* 13(3):526–553
- Das S, Konar A (2009) Automatic image pixel clustering with an improved differential evolution. *Appl Soft Comput* 9(1):226–236
- Das S, Konar A, Chakraborty UK (2005) Two improved differential evolution schemes for faster global search. In: Proceedings of the 7th annual conference on genetic and evolutionary computation, pp 991–998
- Dong CR, Ng WWY, Wang XZ, Chan PPK, Yeung DS (2014) An improved differential evolution and its application to determining feature weights in similarity-based clustering. *Neurocomputing* 146:95–103
- Du J-X, Huang D-S, Wang X-F, Gu X (2007) Shape recognition based on neural networks trained by differential evolution algorithm. *Neurocomputing* 70(4):896–903
- Gämperle R, Müller SD, Koumoutsakos P (2002) A parameter study for differential evolution. In: International conference on advances in intelligent systems, fuzzy systems, evolutionary computation, pp 293–298
- Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. *Evol Comput* 9(2):159–195
- Lampinen J, Zelinka I (2000) On stagnation of the differential evolution algorithm. In: 6th International Mendel conference on soft computing, pp 76–83
- Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295
- Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. *Soft Comput* 9(6):448–462
- Lu X, Tang K, Sendhoff B, Yao X (2014) A new self-adaptation scheme for differential evolution. *Neurocomputing* 146:2–16
- Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* 11(2):1679–1696
- Maulik U, Saha I (2009) Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. *Pattern Recognit* 42(9):2135–2149

- Mezura-Montes E, Coello Coello CA (2003) Adding a diversity mechanism to a simple evolution strategy to solve constrained optimization problems. In: The 2003 congress on evolutionary computation, 8–12 Dec, pp 6–13
- Mezura-Montes E, Velazquez-Reyes J, Coello CAC (2006) A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th annual conference on genetic and evolutionary computation, pp 485–492
- Montgomery J, Chen S (2010) An analysis of the operation of differential evolution at high and low crossover rates. In: 2010 IEEE congress on evolutionary computation, 18–23, pp 1–8
- Price K, Storn RM, Lampinen J (2005) *Differential evolution: a practical approach to global optimization*. Springer, Berlin
- Price KV (1999) *An introduction to differential evolution. New ideas in optimization*. McGraw-Hill Ltd., UK, pp 79–108
- Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* 13(2):398–417
- Ronkkonen J, Kukkonen S, Price KV (2005) Real-parameter optimization with differential evolution. In: The IEEE congress on evolutionary computation (CEC-2005), pp 506–513
- Runarsson TP, Xin Y (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans Evol Comput* 4(3):284–294
- Storn R, Price K (1995) Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
- Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization
- Teo J (2006) Exploring dynamic self-adaptive populations in differential evolution. *Soft Comput* 10(8):673–686
- Varadarajan M, Swarup KS (2008) Differential evolution approach for optimal reactive power dispatch. *Appl Soft Comput* 8(4):1549–1561
- Wang S-C, Yeh M-F (2014) Applying differential evolution to aggregate production planning. *Univ J Ind Bus Manag* 2(7):164–172
- Xiang WI, Zhu N, Ma SF, Xi Meng, An MQ (2015) A dynamic shuffled differential evolution algorithm for data clustering. *Neurocomputing* 158:144–154
- Yeh M-F, Leu M-S, Wang S-C, Wang W-J (2014) Grey adaptive differential evolution algorithm. *J Grey Syst UK* 17(2):67–74
- Yeh M-F, Wang S-C, Leu M-S (2015) Differential evolution with grey evolutionary analysis. *J Grey Syst* 27(2):38–46
- Yong W, Zixing C, Qingfu Z (2011) Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans Evol Comput* 15(1):55–66
- Zaharie D (2003) Control of population diversity and adaptation in differential evolution algorithms. In: Proceedings of Mendel 2003, 9th international conference on soft computing, pp 41–46
- Zhang DG (2012) A new approach and system for attentive mobile learning based on seamless migration. *Appl Intell* 36(1):75–89
- Zhang DG, Liang YP (2013) A kind of novel method of service-aware computing for uncertain mobile applications. *Math Computer Model* 57(3–4):344–356
- Zhang DG, Song XD, Wang X, Li K, Li WB, Ma Z (2015a) New agent-based proactive migration method and system for big data environment (BDE). *Eng Comput* 32(8):2443–2466
- Zhang DG, Zhang XD (2011) Design and implementation of embedded un-interruptible power supply system (EUPSS) for web-based mobile application. *Enterp Inf Syst* 6(4):473–489
- Zhang DG, Zheng K, Zhang T, Wang X (2015b) A novel multicast routing method with minimum transmission for WSN of cloud computing service. *Soft Comput* 19(7):1817–1827
- Zhang DG, Zhu YN, Zhao CP, Dai WB (2012a) A new constructing approach for a weighted topology of wireless sensor networks based on local-world theory for the internet of things (IOT). *Computers Math Appl* 64(5):1044–1055
- Zhang D, Kang X, Wang J (2012) A novel image de-noising method based on spherical coordinates system. *J Adv Signal Process* 2012(1):110
- Zhang D, Li G, Zheng K, Ming X, Pan Z-H (2014a) An energy-balanced routing method based on forward-aware factor for wireless sensor networks. *IEEE Trans Ind Inf* 10(1):766–773
- Zhang D, Wang X, Song X, Zhao D (2014b) A novel approach to mapped correlation of ID for RFID anti-collision. *IEEE Trans Serv Comput* 7(4):741–748
- Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. *IEEE Trans Evol Comput* 13(5):945–958
- Zou D, Wu J, Gao L, Li S (2013) A modified differential evolution algorithm for unconstrained optimization problems. *Neurocomputing* 120:469–481