Triangular Gaussian Mutation to Differential Evolution

Jinglei Guo · Yong Wu* · Wei Xie · Shouyong Jiang

Abstract Differential evolution (DE) has been a popular algorithm for its simple structure and few control parameters. However, there are some open issues in DE regrading its mutation strategies. An interesting one is how to balance the exploration and exploitation behavior when performing mutaion, and this has attracted a growing number of research interests over a decade. To address this issue, this paper presents a triangular Gaussian bare-bones mutation strategy. This strategy utilizes the physical positions and the fitness differences of the vertices in the triangular structure. Based on this strategy, a triangular Gaussian bare-bones DE (TGBDE) and its improved version (ITGBDE) are suggested. Empirical studies are carried out on the 20 benchmark functions and show that, in the comparison with several state-of-the-art DE variants, ITGBDE obtains significantly better or at least comparable results, suggesting the proposed mutation strategy is promising for DE.

Keywords Differential evolution(DE) · Gaussian distribution · triangular structure · global optimum

1 Introduction

Swarm algorithms, inspired by the Darwinian's evolution theory (Darwin C, 2009), are well known to solve complex optimization problems without any requirements of problems' intrinsic differentiability and continuity. Differential evolution (DE), proposed by Storn and Price (Price K et al, 2006; Storn R and Price K, 1995), is a population-based stochastic optimization algorithm. DE differs from other standard evolutionary algorithms (EAs) in the way that DE generates offspring by mutating the solutions with a scaled difference of two randomly selected population vectors. Due to simple structure and few control parameters, DE has attracted a lot of researchers' attention. It is worth noting that some variants of DE have been highly ranked in the competition of the IEEE Congress on Evolutionary Computation (CEC) conference series. Up to date, DE and its variants have become one of the most powerful and competitive algorithms in evolutionary computation and they have been successfully applied to numerous real world areas.

However, the performance of DE is greatly influenced by two components: one is the the new vector generation mutation and crossover strategies; the other is the control parameters setting. In the past decades, a number of DE variants that employ new mutation strategies and/or adapt the control parameters have been proposed to improve the performance of DE. Since 2000, researchers have devoted much effort to designing new mutation schemes and combining existing mutation schemes, such as ODE(Rahnamayan S et al, 2008), TDE(Fan HY et al, 2003), DEGL(Das S et al, 2009),CoDE (Wang Y et al, 2011), OXDE(Wang Y et al, 2012), EPSDE (Mallipeddi R et al, 2011) and so on. A large number of studies also focus on guided adaptation of three primary control parameters of DE, e.g., jDE(Brest J et al, 2006), JADE(Zhang J and Sanderson AC, 2009), ADE(Yu WJ et al, 2014), SHADE(Tenabe R and Fukunaga AS, 2013), and SaDE (Qin AK et al, 2009). Although these strategies are helpful to improve the DE performance, many of them introduce extra complex operations which are not easy to implement.

The rest of this paper is organized as follows. Section 2 briefly introduces the classical DE algorithm. Section 3 sur-

Jinglei Guo
School of Computer Science, Central China Normal University, Wuhan 430079, China
Yong Wu* (Corresponding Author)
school of Automation, Wuhan University of Technology, Wuhan 430070, China
Wei Xie
School of Computer Science, Central China Normal University, Wuhan 430079, China
Shouyong Jiang
School of Computing Science,University of Lincoln, Lincoln, LN6 7TS, UK

E-mail: jingleiguo@mail.ccnu.edu.cn, wuyong@whut.edu.cn, xw@mail.ccnu.edu.cn, math4neu@gmail.com

veys related work on improving DE. In section 4, the proposed approach is described in detail. Experimental results and discussion on the well-known 20 functions are reported in section 5. Finally, section 6 concludes this paper.

2 Classical DE Algorithm

Like other EAs, DE starts with a population of NP random individuals ,$P^g = \{\overrightarrow{X}_i^g | i = 1, 2, \cdots, NP\}$, where g is the generation number, and NP is the population size. In P^g, each individual is called the target vector, i.e. $\overrightarrow{X}_i^g = (x_{i1}^g \cdots x_{iD}^g)$, where D is the dimension of the search space. In the initialization stage ($g = 0$), the jth dimension of vector i is initialized as follows:

$$x_{ij}^0 = L_j + rand() \cdot (U_j - L_j) \quad (1)$$

where $rand()$ generates a uniformly distributed random number in the range $[0, 1]$, L_j and U_j are the lower and upper bounds of the *j*th component of a vector, respectively.

After that, DE iteratively performs three basic operations: mutation, crossover and selection, to evolve the population towards the global optimum. In the evolution of DE, trial vectors are generated through the combination of mutation operation and crossover operation.

2.1 Mutation

At this stage, a mutant vector $\overrightarrow{V}_i^g = \{v_{i1}^g, \cdots, v_{iD}^g\}$ is generated by a mutation operator. Unlike traditional mutation in EA, DE employs the difference of two vectors with a scaling factor F to explore different search regions. The following are the most common mutation operators in DE:

- DE/rand/1:

$$\overrightarrow{V}_i^g = \overrightarrow{X}_{r1}^g + F \cdot (\overrightarrow{V}_{r2}^g - \overrightarrow{V}_{r3}^g) \quad (2)$$

- DE/best/1:

$$\overrightarrow{V}_i^g = \overrightarrow{X}_{best}^g + F \cdot (\overrightarrow{X}_{r1}^g - \overrightarrow{X}_{r2}^g) \quad (3)$$

- DE/current-to-best/1:

$$\overrightarrow{V}_i^g = \overrightarrow{X}_i^g + F \cdot (\overrightarrow{X}_{best}^g - \overrightarrow{X}_i^g) + F \cdot (\overrightarrow{X}_{r1}^g - \overrightarrow{X}_{r2}^g) \quad (4)$$

- DE/rand/2:

$$\overrightarrow{V}_i^g = \overrightarrow{X}_i^g + F \cdot (\overrightarrow{X}_{r2}^g - \overrightarrow{X}_{r3}^g) + F \cdot (\overrightarrow{X}_{r4}^g - \overrightarrow{X}_{r5}^g) \quad (5)$$

- DE/best/2:

$$\overrightarrow{V}_i^g = \overrightarrow{X}_{best}^g + F \cdot (\overrightarrow{X}_{r1}^g - \overrightarrow{X}_{r2}^g) + F \cdot (\overrightarrow{X}_{r3}^g - \overrightarrow{X}_{r4}^g) \quad (6)$$

where F is the scaling factor and the indices r_1, r_2, r_3, r_4 and r_5 are distinct integers randomly selected from $\{1, 2, \cdots, NP\}$, which are different from the base index i. $\overrightarrow{X}_{best}^g$ is the best vector in the current population.

2.2 Crossover

After mutation, a trial vector $\overrightarrow{U}_i^g = \{u_{i1}^g, \cdots, u_{iD}^g\}$ is generated by a binomial crossover operator on $\overrightarrow{X}_i^g$ and $\overrightarrow{V}_i^g$ as follows:

$$u_{ij}^g = \begin{cases} v_{ij}^g & if\ rand() \leq CR\ or\ j = j_{rand} \\ x_{ij}^g & otherwise \end{cases} \quad (7)$$

where CR is the crossover control parameter, and similar to Eq. (1), $rand()$ is a random number in $[0, 1]$, and $j_{rand} \in \{1, 2, \ldots, D\}$ is a randomly chosen index. The condition $j = j_{rand}$ ensures that the trial vector $\overrightarrow{U}_i^g$ is different from the corresponding target vector $\overrightarrow{X}_i^g$ on at least one dimension.

2.3 Selection

Selection operator in DE is a one-to-one greedy replacement between the trial vector and the corresponding vector. For minimization problems, if the trial vector $\overrightarrow{U}_i^g$ is smaller than or equal to the target vector $\overrightarrow{X}_i^g$, the trial vector survives to next generation; otherwise, the target vector enters the next generation:

$$\overrightarrow{X}_i^{g+1} = \begin{cases} \overrightarrow{U}_i^g & if\ f(\overrightarrow{U}_i^g) \leq f(\overrightarrow{X}_i^g) \\ \overrightarrow{X}_i^g & \text{otherwise} \end{cases} \quad (8)$$

where f is the objective function .

3 Related Work

In the past two decades, continuous effort has been made to improving DE , as presented in the comprehensive survey made by Das and Suganthan(Das S and Suganthan PN, 2011; Das S et al, 2016). The improvement on DE can be classified into two categories: 1) dynamic adaptation or self-adaptation of control parameters; 2) new strategies to generate trial vectors.

3.1 Adaptation of control parameters

There are three control parameters: scaling factor(F), crossover rate(CR) and population size(NP) in DE. The performance of DE is sensitive to the associated parameters, especially for complex problems(Gamperle R et al, 2002). Thus, a good volume of research work has been undertaken to tune the control parameters. Storn and Price(Storn R and Price K, 1995) suggested that NP should be between [5D,10D] and F should be in the range of [0.4,1]. Brest *et al.* (Brest J et al, 2006) proposed a self-adaptation scheme(jDE) for DE in which the control parameters F and CR are encoded into the the individuals and adjusted by two factors τ_1 and τ_2. Qin *et al.* (Qin AK et al, 2009) introduced SaDE, in which each mutation strategy associated with parameters F and CR is self-adapted by the experience of previous LP generations. In SaDE, F is approximated by the normal distribution with mean value 0.5 and standard deviation 0.3, denoted by $N(0.5, 0.3^2)$ and CR obeys the normal distribution with the mean value 0.5 and standard deviation 0.1, denoted by $N(0.5, 0.1^2)$. FiADE(Ghosh A et al, 2011), proposed by Ghosh *et al.*, is a simple yet effective parameter adaptation based on the objective function value. The presented results showed that the performance of FiADE is very competitive with the best-known DE variants. Zhang and Sanderson (Zhang J and Sanderson AC, 2009) found better control parameters tend to generate more successful individuals and they made use of the most recent successful mutation information to propagate the new F and CR in JADE. Tanabe and Fukunaga (Tenabe R and Fukunaga AS, 2013) proposed a success-history based parameter adaptation DE (SHADE) to revise the selection of future control parameter values in JADE . Awad *et al.* (Awad NH et al, 2016) used a sinusoidal function to automatically adapt the values of the scaling factor F to further improve LSHADE. The population size NP plays an important role on the number of function evaluations and converge rate. Mallipeddi *et al.* (Mallipeddi R et al, 2011) suggested that separable and unimodal functions require smaller population sizes to speed up the convergence, while parameter-linked multimodal functions require large populations to avoid premature convergence. Yu *et al.* (Yu WJ et al, 2014) studied the relationship between the population state and control parameters and proposed an adaptive DE (ADE) algorithm with a two-level adaptive parameter control scheme. In ADE, the population-level parameters F_p and CR_p for the whole population are adaptively controlled according to the optimization states, namely, the exploration state and the exploitation state in each generation. The individual-level parameters F_i and CR_i for each individual are generated by adjusting the population-level parameters. Experimental results showed that ADE generally outperforms four state-of-the-art DE variants on different kinds of optimization problems. Brest and Maucec(Brest J and Maucec MS, 2009) found that a higher population diversity is needed at the beginning of the evolution and the exploitative ability is more favourable in the last stage. Dynamic population size reduction techniques (Brest J and Maucec MS, 2009) were proposed to improve the efficiency and robustness of the DE algorithm. Tanabe and Fukunaga (Tanabe R and Fukunaga AS, 2014) improved SHADE with linear population size reduction and named it LSHADE, which was the winner in the CEC 2014 competition on real parameter singe objective optimization.

3.2 New Trial Vector Generation Strategies

Despite the combination of existing mutation strategies in DE, researchers have made several attempts to design new mutation and crossover schemes for generation new trial vector. Zhang and Sanderson (Zhang J and Sanderson AC, 2009) proposed a new mutation strategy named "DE/current-to-pbest" with an optional external archive which utilizes historical data to provide information of progress direction. Due to the external archive, JADE showed promising results for relatively high dimensional problems. Das *et al.* (Das S et al, 2009) described an improved variant of the DE/current-to-best/1 scheme, which utilizes the concept of the physical neighborhood of population members, to balance the exploration and exploitation abilities of DE without imposing the function evaluation burden. Epitropakis *et al.* (Epitropakis MG et al, 2011) proposed a proximity-based mutation operator which selects parents based on the pairwise distance possibility, with the closest neighbor of a vector having the highest probability to be a parent. Authors in (Epitropakis MG et al, 2011) incorporated this proximity-based mutation operator in the original DE algorithm, as well as other recently proposed DE variants to enhance performance for the majority of the benchmark problems studied. Fan and Lampinen (Fan HY et al, 2003) proposed trigonometric mutation and embedded it into DE. This operator forms a hypergeometric triangle in the search space where the selected individuals are viewed as the vertices, and utilizes the selected individuals' objective function values to ensure the perturbations can be biased towards the best points. Wang *et al.* (Wang Y et al, 2012) proposed orthogonal crossover (OX) operators to make a systematic and rational search in a region defined by the parent solutions. Gong and Cai (Gong W and Cai Z, 2013) proposed a ranking-based mutation operator for DE, where the individuals are proportionally selected according to their rankings in the current population. In the ranking-based mutation operator, the individuals with better rankings have more opportunity to be selected as parents. Guo *et al.* (Guo SM et al, 2014) proposed a successful-parent-selection framework to improve the performance of DE by providing alternatives for the selection of parents during mutation and crossover. The recently updated solutions are stored into an archive and some parents in the achieve are chosen to perform mutation and crossover so as to alleviate evolutionary stagnation. Wang *et al.* (Wang Y et al, 2016) proposed a cumulative population distribution framework called CPI-DE. In CPI-DE, two trial vectors are created based on the original coordinate system and eigen coordinate system. Wang *et al.* (Wang H et al, 2013) proposed the Gaussian bare-bones DE (GBDE) and modified GBDE (MGBDE), in which the Gaussian mutation strategy generates trial vectors randomly from a Gaussian distribution with mean μ and standard deviation σ. The proposed Gaussian mutation strategy eliminates the parameter F, and μ and σ are calculated from the current individual and the best individual from the population.

4 Triangle Brare-Bones DE

Despite various improvements, most DE variants are still not free from control parameters, which means they have to be provided optimal parameter settings in order to perform to the best of their ability. Also, optimal parameter values are often problem dependent. Inspired by the idea of the bare-bone particle swarm algorithm(Kennedy J, 2003), Wang (Wang H et al, 2013) proposed a two-fold modification of DE, named them GBDE and Modified GBDE(MGBDE), to ease the effects of control parameters.

4.1 GBDE and MGBDE

Unlike common mutation operator, the mutant vectors in GBDE are generated by sampling randomly from Gaussian distribution with mean value $(\overrightarrow{X}^g_{best} + \overrightarrow{X}^g_i)/2$ and standard deviation $|\overrightarrow{X}^g_{best} - \overrightarrow{X}^g_i|$ as follows,

$$\overrightarrow{V}^g_i = N(\frac{\overrightarrow{X}^g_{best} + \overrightarrow{X}^g_i}{2}, |\overrightarrow{X}^g_{best} - \overrightarrow{X}^g_i|^2) \quad (9)$$

As the mean value and standard deviation are calculated from the current population, the parameter F is omitted in GBDE. The binomial crossover operator Eq. (7) is used to generate the trial vector, but the crossover possibility CR is chosen from the Gaussian distribution with mean value 0.5 and standard deviation 0.1. The CR value is individual independent and will keep unchanged if the solution has been updated in the current generation, otherwise it will be set to a value generated by the Gaussian distribution $N(0.5, 0.1^2)$. Thus, CR is self-adapted as follows:

$$CR^{g+1}_i = \begin{cases} CR^g_i, & if\ f(U^g_i) \leq f(X^g_i) \\ N(0.5,\ 0.1^2), & \text{otherwise} \end{cases} \quad (10)$$

In the modified GBDE, named MGBDE, each individual is assigned to a mutation strategy (either DE/best/1 or the Gaussian mutation) during the population initialization and sticks to the mutation strategy during the search process.

4.2 TGBDE

4.2.1 Motivation

In either GBDE or MGBDE, only the best vector $\overrightarrow{X}^g_{best}$ in the current population plays an attractor role in the searching process, that is all the individuals move towards $\overrightarrow{X}^g_{best}$.

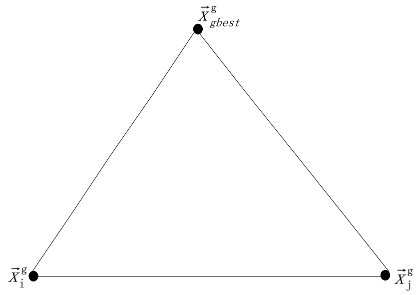


Fig. 1 Triangular topology of TGBDE.

It is obvious that the standard deviation $|\overrightarrow{X}^g_{best} - \overrightarrow{X}^g_i|$ in Eq.(9) is small when the current vector $\overrightarrow{X}^g_i$ is very close to the best vector $\overrightarrow{X}^g_{best}$. In this situation, the Gaussian mutation in Eq. (9) only produces a slight vibration surrounding $\overrightarrow{X}^g_{best}$ due to the small $|\overrightarrow{X}^g_{best} - \overrightarrow{X}^g_i|$ value. This is very likely to cause evolutionary stagnation if $\overrightarrow{X}^g_{best}$ is a local optimum rather than global optimum. In view of this drawback, we suggest a new triangular Gaussian bare-bones mutation strategy. The new strategy employ an extra vector, which is selected randomly in the current population, to enhance the perturbation. The best vector $\overrightarrow{X}^g_{best}$, the current vector $\overrightarrow{X}^g_i$ and the randomly selected vector $\overrightarrow{X}^g_j$ together form a hypergometric triangle in the search apace, as shown in Figure 1.

Although the Gaussian bare-bones mutation in GBDE eliminates the scaling factor F, it only utilizes the physical position of the vectors and ignores the fitness status of the vectors. In contrast to GBDE, TGBDE takes into consideration both the physical position information and objective function values of the candidate vectors when creating perturbations, which is detailed in the following subsection.

4.2.2 Proposed TGBDE

Based on the analysis of the search behaviour of GBDE and MGBDE, we propose a new triangular Gaussian bare-bones mutation strategy. In this strategy, three vectors, i.e., the global best vector $\overrightarrow{X}^g_{best}$, the current vector $\overrightarrow{X}^g_i$ and the random vector $\overrightarrow{X}^g_j$ in the population, are considered vertices to construct a triangle. The center of the triangle's vertices is viewed as the mean value μ and the weighted sum of the three vector differences as the deviation value σ in Gaussian distribution function. Therefore, the triangular Gaussian bare-bones mutation strategy is defined by

$$\overrightarrow{V}^g_i = N(\mu, \sigma^2) \quad (11)$$

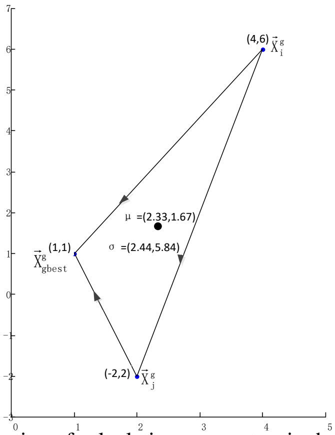


Fig. 2 An illustration of calculating parameters in the triangular Gaussian bare-bones mutation strategy for the 2-D Sphere problem.

where

$$\mu = (\overrightarrow{X}^g_{best} + \overrightarrow{X}^g_i + \overrightarrow{X}^g_j)/3.0 \quad (12)$$

$$\sigma = (p_2 - p_1)(\overrightarrow{X}^g_{best} - \overrightarrow{X}^g_i) + (p_3 - p_1)(\overrightarrow{X}^g_{best} - \overrightarrow{X}^g_j) + (p_3 - p_2)(\overrightarrow{X}^g_i - \overrightarrow{X}^g_j) \quad (13)$$

with

$$p_1 = |f(\overrightarrow{X}^g_{best})|/sum\_p$$
$$p_2 = |f(\overrightarrow{X}^g_i)|/sum\_p$$
$$p_3 = |f(\overrightarrow{X}^g_j)|/sum\_p$$
$$sum\_p = |p_2 - p_1| + |p_3 - p_1| + p_2 - p_1|$$

The calculation of μ and σ for the 2-D Sphere problem $f(x) = \sum_{i=1}^{D} x_i^2$ is illustrated in Figure 2. in the figure, the mutant vector $\overrightarrow{X}^g_i = (4, 6)$, the random selected vector $\overrightarrow{X}^g_i = (-2, 2)$ and the best vector of current population $\overrightarrow{X}^g_{gbest} = (1, 1)$ are chosen to produce a perturbed vector. Thus, we can easily derive $\mu = (2.33, 1.67)$ and $\sigma = (2.44, 5.84)$ using Eq.(12) and Eq.(13), respectively.

It can be seen that the standard deviation part in Eq.(12) is associated with the three edges of the triangle, i.e., $(\overrightarrow{X}^g_{best} - \overrightarrow{X}^g_i)$, $(\overrightarrow{X}^g_{best} - \overrightarrow{X}^g_j)$ and $(\overrightarrow{X}^g_i - \overrightarrow{X}^g_j)$. $(p_2 - p_1)$, $(p_3 - p_1)$ and $(p_2 - p_3)$ in Eq.(13) are weights. The signs of weights determine the shifting direction of the vector difference and the absolute values of $(p_2 - p_1)$, $(p_3 - p_1)$ and $(p_2 - p_3)$ determine the step-size. The weights enable a preference to produce a new high-quality vector because their signs cause the donors to move towards a vertex with a better objective value. The values of the weights can automatically magnify the contribution of the vector differentials to the standard deviation σ such that the greater the difference in the fitness values between the vertices that form a triangular structure, the larger contribution the corresponding vector difference.

It is noticeable from the above analysis that the triangular topology directs the search to promising areas since it tends to generate new trial solutions towards the better ones of three individuals in the triangular topology. Compared with GBDE, the TGBDE mutation strategy utilizes the triangular structure which introduces an extra vector to make the current vector $\overrightarrow{X}^g_i$ not only learn the information of global best individual but also gain the learning experience of another random vector $\overrightarrow{X}^g_j$.

Furthermore, the search behaviour of TGBDE is a fine-tuned procedure which emphasizes exploration at the early stage and then gradually switches to exploitation at the late stage. At the early stage, the distances between vectors are large because the vectors are randomly distributed in the search space. Thus the search process focuses on exploration due to the large deviation σ in (11). As the evolution precedes, the distances between vectors become smaller due to the attraction of outstanding vectors, which makes the search process focus on exploitation.

Similar to the classical DE, TGBDE also utilizes the binomial crossover scheme in Eq.(7). Obviously, the parameter CR greatly affects the performance of TGBDE. When the crossover probability CR is large, the trial vector $\overrightarrow{U}^g_i$ inherits more components from the triangular Gaussian mutant vector $\overrightarrow{V}^g_i$ which is generated by the three vertices (the current vector $\overrightarrow{X}^g_i$, the random vector $\overrightarrow{X}^g_j$ and global best vector $\overrightarrow{X}^g_{best}$). It means the trial vector will learn more information from the random vector $\overrightarrow{X}^g_j$ and global best vector $\overrightarrow{X}^g_{best}$. When CR is small, the trial vector $\overrightarrow{U}^g_i$ inherits more components from current vector $\overrightarrow{X}^g_i$. This means the trial vector will be similar to the current vector, and the convergence speed will decrease. Each individual $\overrightarrow{X}^g_i$ is considered to have its own CR_i parameter. CR_i is set by normal distribution with mean μ_{CR} and standard variance σ_{CR}:

$$CR_i = N(\mu_{CR}, \sigma^2_{CR}) \quad (14)$$

where σ_{CR} is set to 0.1, as suggested in GBDE or MGBDE (Wang H et al, 2013).

The pseudocode of TGBDE is described in Algorithm 1. Again, we use the 2-D Sphere problem as an example to illustrate the search behavior of TGBDE. The plots in Figure 3 show the population distribution at generation g=0, 10, 20, and 30, respectively. At generation $g = 0$, population individuals are widely distributed in the search space $[-100, 100]^2$. At this stage, the population has a high level of diversity and TGBDE mainly focuses on exploration. With an increase in the number of generations (i.e., g increases from 0 to 30), the population diversity decreases gradually and the search behaviour transforms from exploration to exploitation.

4.2.3 Improved TGBDE (ITGBDE)

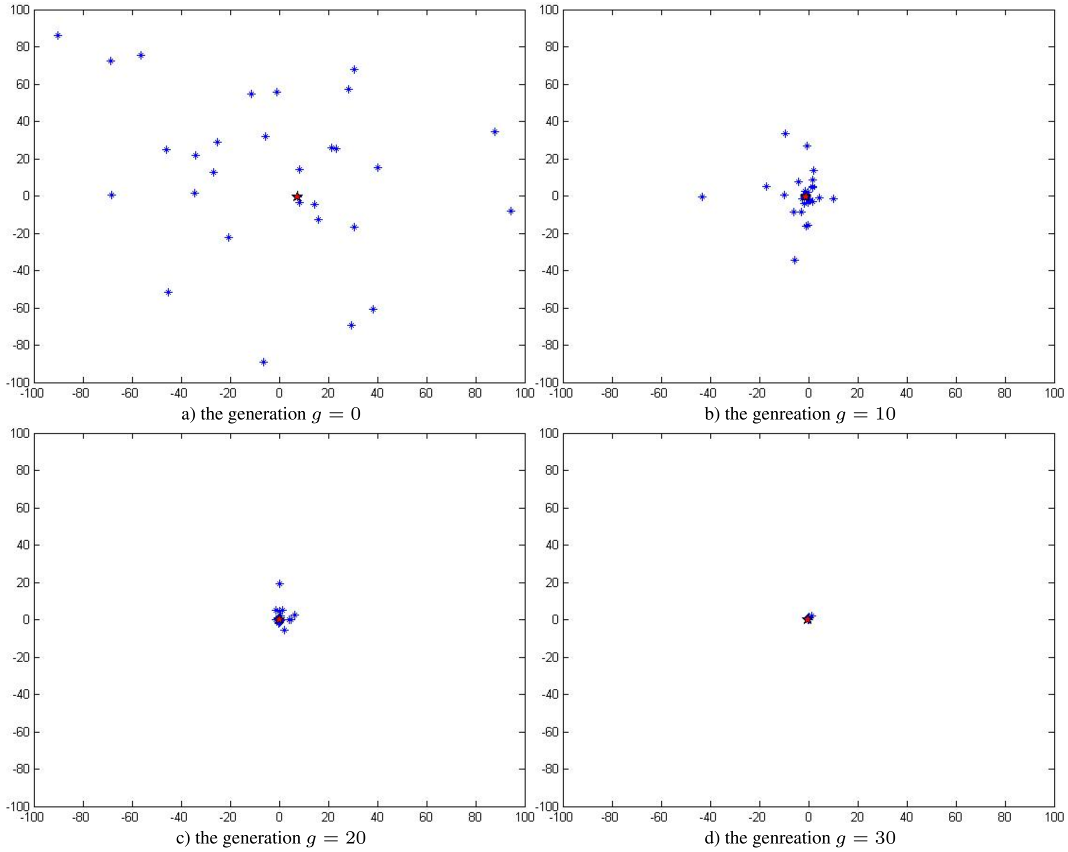


Fig. 3 Search behaviour of TGBDE for 2-D Sphere function, where '$*$' in blue represents the individuals in the current population at generation g and '•' in red indicates the best individual in the current population.

Algorithm 1 Triangular Gaussian Bare-Bones DE

```
Randomly initialize the population at generation g = 0.
Set μ_CR.
while the stopping criterion is not satisfied do
    for i := 1 to N_p do
        CR_i = randn(μ_CR, 0.1)
        Generate the mutant vector V_i^g according to (11)
        Generate the trial vector U_i^g according to (7)
        Evaluate the trial vector U_i^g
        if f(U_i^g) ≤ f(X_i^g) then
            X_i^{g+1} = U_i^g
        else
            X_i^{g+1} = X_i^g
        end if
    end for
    g ← g + 1
end while
```

TGBDE is still likely to get trapped into local minima, although it adopts a triangular structure in its mutation strategy to reduce the likelihood. This happens particularly when the three vectors $\overrightarrow{X}_{best}^{g}$, $\overrightarrow{X}_{i}^{g}$ and $\overrightarrow{X}_{j}^{g}$ roughly lie in a single line, which renders the triangular structure of limited use. To further improve the exploration ability of TGBDE, we suggest to combine TGBDE with the "DE/rand/1" strategy, which leads to the improved TGBDE or ITGBDE for short. The "DE/rand/1" strategy selects all vectors for mutation randomly from the current population, so it can guarantee there is sufficient exploration during the search for ITGBDE. In every generation, each vector is assigned one strategy (either TGBDE or "DE/rand/1") with possibility τ, and τ is adaptively adjusted based on each strategy's recent performance. Thus, ITGBDE's mutation strategy can be expressed as follows:

$$M_i = \begin{cases} TGBDE \quad rand(0,1) \leq \tau \\ DE/rand/1 \quad otherwise \end{cases} \tag{15}$$

Like SaDE (Qin AK et al, 2009), the mutation strategy is selected according to the probability τ learned from a certain number of previous generations. At the beginning of evolution, τ is initialized as 0.5, i.e., the two strategies have equal possibility to be chosen. In every generation g, n_k^g records the times that the strategy k ($k = 1, 2$) is selected to produce mutant vectors, and ns_k^g records the number of trial vectors generated by strategy k that can survive to next generation. After a fixed number of previous generation (LP), called the learning period, we can calculate the success ratio (sr_k) for strategy k, the formula is as follows,

$$sr_k = \frac{\sum_{g=t}^{t+LP-1} ns_k^g}{\sum_{g=t}^{t+LP-1} n_k^g} \tag{16}$$

Algorithm 2 Improved Triangular Gaussian Bare-Bones DE

Randomly initialize the population at generation $g = 0$.
Set μ_{CR}, μ_F and $\tau = 0.5$
while the stopping criterion is not satisfied **do**
  for $i := 1$ **to** N_p **do**
    CR_i=$randn(\mu_{CR}, 0.1)$
    F_i=$randn(\mu_F, 0.1)$
    if $rand(0, 1) < \tau$ **then**
      Generate the mutant vector $\overrightarrow{V}_i^g$ according to (11)
      $n_1^g \leftarrow n_1^g + 1$
    else
      Generate the mutant vector $\overrightarrow{V}_i^g$ according to (2)
      $n_2^g \leftarrow n_2^g + 1$
    end if
    Generate the trial vector $\overrightarrow{U}_i^g$ according to (7)
    Evaluate the trial vector $\overrightarrow{U}_i^g$
    if $f(\overrightarrow{U}_i^g) \leq f(\overrightarrow{X}_i^g)$ **then**
      $\overrightarrow{X}_i^{g+1}$=$\overrightarrow{U}_i^g$
    else
      $\overrightarrow{X}_i^{g+1}$=$\overrightarrow{X}_i^g$
    end if
    if $f(\overrightarrow{U}_i^g) \leq f(\overrightarrow{X}_i^g)$ **then**
      if $strategy_k$ is selected **then**
        $ns_k^g \leftarrow ns_k^g + 1$
      end if
    end if
  end for
  if $g\%LP == 0$ **then**
    update τ according to(17)
    Set all $n_k^g, ns_k^g = 0$
  end if
  $g \leftarrow g + 1$
end while

According to the sr_k value, the τ is computed by

$$\tau = LBP + (1 - 2 \cdot LBP)\frac{sr_1}{sr_1 + sr_2} \quad (17)$$

where LBP is the lower bound possibility of τ and ensures each mutation strategy has at least LBP to participate mutation.

The pseudocode of ITGBDE is described in Algorithm 2. In Algorithm 2, the scaling factor F for the "DE/rand/1" strategy is generated by Gaussian distribution with mean value μ_F and standard deviation 0.1. The crossover possibility CR remains the same as Eq.(14) in TGBDE. The computational cost of ITGBDE is $O(T \cdot NP \cdot D)$, which is determined by the iteration number T, the population size NP and the dimension of search space D.

To better demonstrate the difference between TGBDE and improved TGBDE(ITGBDE), the 2-D Shekel's Foxholes function (as shown in Figure 4) is adopted as a case study. The Shekel's Foxholes function is a typical multimodal function with 24 distinct local minima and the global minimum $f(-32, -32) = 0.998004$ in the domain $[-65.536, 65.536]^2$. Both TGBDE and ITGBDE with $NP = 30$, $\mu_F = 0.5$, $\mu_{CR} = 0.5$, $LP = 50$ and $LBP = 0.3$ are tested on this problem. The population distribution of TGBDE and

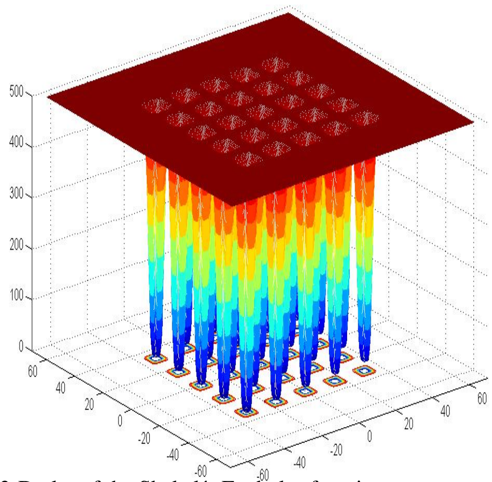


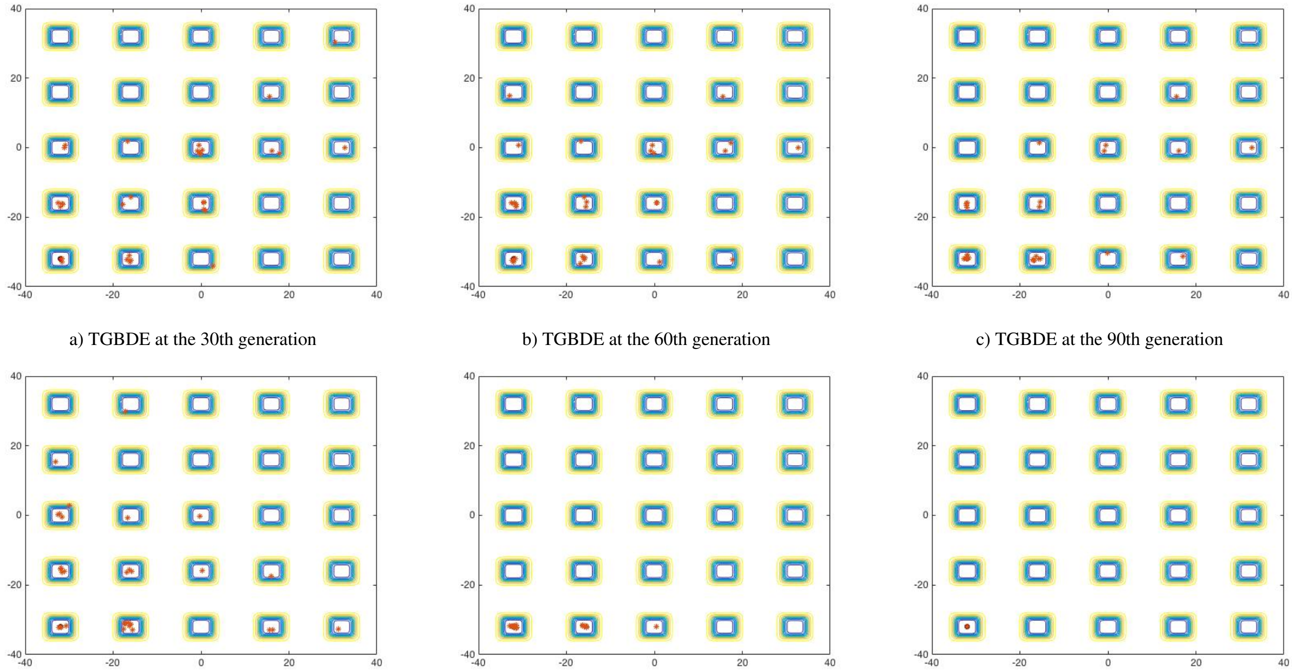
Fig. 4 3-D plot of the Shekel´s Foxholes function.

ITGBDE at the 30th, 60th and 90th generation is plotted in Figure 5. It is observable from the figure that both population individuals of TGBDE and ITGBDE are scattered in most local minima at the 30th generation. As the number of generations increases, the population individuals try to jump out of local optima and move towards the global optimum. At the 60th generation, the individuals of ITGBDE gathers in three local optima near the global optimum whereas the population of TGBDE is still distributed in 13 local optima areas. At the 90th generation, all the individuals of ITGBDE reaches the global minimum point $(-32, -32)$ but most individuals of TGBDE still get trapped in 10 local optima. It implies that ITGBDE maintains a good balance between exploration and exploitation due to the proposed strategies.

5 Experimental Study

5.1 Test Functions

In order to test the performance of the proposed approach, we choose a set of 20 benchmark test functions in the following experiments. The detailed description of these benchmark functions is shown in Table 1. The first 13 functions are well-known benchmarks, which have been widely used in the literature (Brest J et al, 2006; Brest J and Maucec MS, 2009; Wang H et al, 2013; Yao X et al, 1999; Zhang J and Sanderson AC, 2009). Here, F_{01} -F_{04} are continuous unimodal functions. F_{05} has a narrow valley from the perceived local optimum to the global optimum, F_{06} is a discontinuous step function with one minimum, F_{07} is a noisy quartic function, and F_{08}-F_{13} are multimodal functions with many local minima. F_{14}-F_{20} are shifted functions used in (Wang H et al, 2009).



a) TGBDE at the 30th generation b) TGBDE at the 60th generation c) TGBDE at the 90th generation

d) ITGBDE at the 30th generation e) ITGBDE at the 60th generation f) ITGBDE at the 90th generation

Fig. 5 Search behaviour of TGBDE for 2-D Shekel$\acute{}$s Foxholes function, where '$*$' represents population distribution at generation g.

Table 1 Benchmark functions

Type	functions	Name	$f(\overrightarrow{X}^*)$	Search range
Unimodal Functions	F_{01}	Sphere	0	[-100,100]
	F_{02}	Schwefel 2.22	0	[-10,10]
	F_{03}	Schwefel 1.2	0	[-100,100]
	F_{04}	Schwefel 2.21	0	[-100,100]
	F_{05}	Rosenbrock	0	[-100,100]
	F_{06}	Step	0	[-100,100]
	F_{07}	Quartic with Noise	0	[-1.28,1.28]
Multimodal Functions	F_{08}	Schwefel 2.26	-1259.5	[-500,500]
	F_{09}	Rastrigin	0	[-5.12,5.12]
	F_{10}	Ackley	0	[-32,32]
	F_{11}	Griewank	0	[-600,600]
	F_{12}	Penalized1	0	[-50,50]
	F_{13}	Penalized2	0	[-50,50]
Shift Functions	F_{14}	Shift Sphere	0	[-100,100]
	F_{15}	Shift Schwefel 1.2	0	[-100,100]
	F_{16}	Shift Schwefel 1.2 with Noise	0	[-100,100]
	F_{17}	Shift Griewank	0	[-600,600]
	F_{18}	Shift Ackley	0	[-32,32]
	F_{19}	Shift Penalized1	0	[-50,50]
	F_{20}	Shift Penalized2	0	[-50,50]

5.2 Influence of Parameter Settings

In ITGBDE, two control parameters μ_F and μ_{CR} are introduced and the values of these parameters affect its performance. In order to make a better choice of there control parameters, we investigate the influence of different μ_F and μ_{CR} values. We make four common combinations, that is μ_F=0.5/μ_{CR}=0.2, μ_F=0.5/μ_{CR}=0.5, μ_F=0.9/μ_{CR}=0.2 and μ_F=0.9/μ_{CR}=0.5. The dimensions of testing functions were all set to 30 (D=30), the maximal number of fitness function evaluations was $10000 \times D$. The population size NP was 30 (NP=30) and lower bound possibility was set to 0.3 ($LBP = 0.3$). ITGBDE with each combination of μ_F and μ_{CR} was run 30 times on each test function, the mean error and standard deviation values are presented in Table 2. The best mean error for each function across the different parameter combination is shown in boldface.

To select the best value of μ_F and μ_{CR} at a statistical level, the Friedman test(García S et al, 2009, 2010) is conducted to obtain an average ranking. Table 3 shows the average ranking of TGBDE with different μ_{CR} and μ_F values. From Table 3, μ_F=0.5/μ_{CR}=0.5 achieves the best average ranking. Therefore the combination of μ_F=0.5/μ_{CR}=0.5 is adopted in the following experiments.

5.3 Comparison with Other DE Variants

A number of DE variants are employed for empirical study. The algorithms, together with their parameter settings, are listed as follows:

1. OXDE (Wang Y et al, 2012) with $NP = D$, $F = 0.95$, $CR = 0.9$;
2. SADE (Zhang J and Sanderson AC, 2009) with an external archive, $NP = 30$, $F = 0.5$, $CR = 0.3$;
3. MGBDE (Wang H et al, 2013) with $NP = 100$, $F = 0.9$, $CR = 0.9$;
4. TGBDE (proposed in 4.2.2) with $NP = 30$, $\mu_{CR} = 0.5$;
5. ITGBDE (proposed in 4.2.3) with $NP = 30$, $\mu_{CR} = 0.5$, $\mu_F = 0.5$, LBP=0.3, and LP=50 (suggested in SaDE (Zhang J and Sanderson AC, 2009)).

To make a fair comparison, the parameters of OXDE, SADE and MGBDE were set the same as those used in their original literature. The maximum number of function evaluations FEs was set to $10000 \times D$. Each algorithm was executed independently 30 times, the mean solution error and standard deviation are reported in Tables 4 and 5 for $D = 30$ and $D = 50$, respectively. The wilcoxon's rank-sum test at the 0.05 significance level is employed to judge the significant difference between ITGBDE and the other algorithms, $'+'$, $'-'$ or $' ='$ represent our proposed algorithm ITGBDE is better than, worse than, or similar to the compared one, respectively, in the Wilcoxon's rank-sum test. The comparison results are summarized as $'w'$, $'l'$ and $'t'$ in the last three rows of the table, which mean that ITGBDE wins in w functions, loses in l functions and ties in t functions.

Observing the results in Table 4 for 30-D test functions, we can see that ITGBDE performs generally better than the other algorithms over most of the test functions. To be specific, ITGBDE obtains competitive and in most case the best results in all three groups of test problems (i.e., unimodal, multimodal, and shifted functions). On the first seven unimodal problems, ITGBDE provides very good function values, showing it has very high ability of exploitation. It is defeated mainly by TGBDE on f_{01} and f_{02}. This is understandable because ITGBDE allocates some computational resources to DE/rand/1 for exploration in addition to using TGBDE for exploitation, whereas TGBDE focuses only on exploitation, which is favorable when solving these two unimodal functions. For the second multimodal group (f_{08} to f_{13}), ITGBDE performs comparatively with MGBDE and SADE, indicating ITGBDE is able to solve some multimodal problems. TGBDE is better than OXDE but cannot compete with MGBDE, SADE, and ITGBDE. The difference between TGBDE and ITGBDE implies the hybrid mutation strategy can increase exploration so as to jump out of local minima when solving multimodal problems. For the last seven shifted functions, ITGBDE performs best. Particularly, the outperformance of ITGBDE over the other algorithms on the last four shifted multimodal functions clearly illustrates that ITGBDE is capable of handling multimodality, due to great exploration ability rendered by the proposed hybrid mutation strategy. It is noteworthy that MGBDE also employs a Gaussian bare-bones model, but it is better than ITGBDE only on f_{11}, while ITGBDE outperforms MGBDE on 11 functions, as evidenced by statistical testing. This comparison indirectly verifies the effectiveness of the proposed triangular structure and the hybrid mutation strategy.

Similar observations can be made from Table 5 for the case of $D = 50$. Again, ITGBDE provides better results than the other algorithms for most of the test functions, including both unimodal and multimodal ones. The high performance on unimodal problems shows that ITGBDE has a nice exploitation ability so that it can converge to the minimum rapidly. The high performance on multimodal problems indicates that ITGBDE can not only explore the search space widely so as to avoid being trapped in local minima, but also have the ability to balance exploitation and exploration in order to find the global minimum with a fast convergence speed.

In order to determine whether there is a statistically significant difference between the results obtained by the compared algorithms on the test suite, we conducted the Friedman test (García S et al, 2009, 2010). Table 6 shows the average ranking of the five DE variants on the test suite for both D=30 and D=50, and the best value is highlighted in boldface. We can rank the five algorithms, according to their average rankings indicated in the table, from best to worst as follows: ITGBDE, MGBDE, SADE, TGBDE and OXDE. Therefore, ITGBDE can significantly outperform the other algorithms on the test problems used in this paper.

Figure 6 shows the convergence curves of OXDE, MGBDE, SADE, TGBDE and ITGBDE on some 30-D example functions, and similar observations can be obtained from the other functions and are not included here. It can be observed from the figure that TGBDE and ITGBDE have a faster convergence speed than the others on all the functions excluding f_{04} during the search. While most the algorithms seem premature, ITGBDE is still able to further lower the function value, as indicated on f_{04}. Overall, ITGBDE has better performance than TGBDE, showing the proposed hybrid mutation strategy indeed improves DE.

Table 2 The results obtained by TGBDE with different control parameters

	μ_F=0.5/μ_{CR}=0.2	μ_F=0.5/μ_{CR}=0.5	μ_F=0.9/μ_{CR}=0.2	μ_F=0.9/μ_{CR}=0.5
	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
F_{01}	3.33E-156 ± 6.34E-156	2.02E-174 ± 0.00E+00	1.31E-124 ± 1.84E-124	**8.03E-182 ± 0.00E+00**
F_{02}	1.27E-102 ± 6.97E-103	8.52E-121 ± 7.60E-121	5.13E-88 ± 3.05E-88	**6.20E-124 ± 1.31E-123**
F_{03}	2.05E+00 ± 1.76E+00	6.70E-08 ± 7.52E-08	1.03E+01 ± 6.57E+00	**2.35E-18 ± 3.42E-18**
F_{04}	**4.33E-29 ± 7.32E-29**	1.41E-18 ± 7.08E-18	1.39E-28 ± 1.85E-28	3.00E-07 ± 1.30E-06
F_{06}	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F_{07}	6.33E-03 ± 6.56E-03	**1.21E-03 ± 4.08E-04**	1.97E-03 ± 5.49E-04	1.59E-03 ± 5.86E-04
F_{08}	**-1.26E+04 ± 1.85E-12**	-1.25E+04 ± 7.41E+01	**-1.26E+04 ± 1.85E-12**	-1.20E+04 ± 3.14E+00
F_{09}	**0.00E+00 ± 0.00E+00**	4.05E+00 ± 2.07E+00	**0.00E+00 ± 0.00E+00**	1.65E+01 ± 4.97E+00
F_{10}	**3.55E-15 ± 0.00E+00**	**3.55E-15 ± 0.00E+00**	5.21E-15 ± 1.80E-15	3.93E-01 ± 6.01E-01
F_{11}	**0.00E+00 ± 0.00E+00**	1.40E-03 ± 3.22E-03	**0.00E+00 ± 0.00E+00**	6.15E-03 ± 9.78E-03
F_{12}	**1.57E-32 ± 5.57E-48**	**1.57E-32 ± 5.57E-48**	**1.57E-32 ± 5.57E-48**	4.49E-02 ± 1.35E-01
F_{13}	**1.35E-32 ± 5.57E-48**	**1.35E-32 ± 5.57E-48**	**1.35E-32 ± 5.57E-48**	5.54E-02 ± 2.91E-01
F_{14}	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	3.53E-31 ± 1.05E-30
F_{15}	2.46E+00 ± 2.13E+00	5.97E-08 ± 7.33E-08	8.51E+00 ± 6.60E+00	**3.71E-18 ± 6.08E-18**
F_{16}	6.12E+01 3.43E+01	**3.73E-06 ±5.19E-06**	9.61E+01 ± 4.00E+01	1.09E+01 ± 5.94E+01
F_{17}	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	3.94E-03 ± 5.12E-03
F_{18}	3.67E-15 ± 6.49E-16	**3.55E-15 ± 0.00E+00**	5.68E-15 ± 1.77E-15	1.83E-01 ± 4.78E-01
F_{19}	**1.57E-32 ± 5.57E-48**	**1.57E-32 ± 5.57E-48**	**1.57E-32 ± 5.57E-48**	3.81E-02 ± 1.72E-01
F_{20}	**1.35E-32 ± 5.57E-48**	**1.35E-32 ± 5.57E-48**	3.66E-04 ± 2.01E-03	5.62E-02 ± 2.93E-01

Table 3 The average rankings of ITGBDE with different control parameters

Control Parameters	Ranking
μ_F=0.5/μ_{CR}=0.2	2.30
μ_F=0.5/μ_{CR}=0.5	**2.03**
μ_F=0.9/μ_{CR}=0.2	2.70
μ_F=0.9/μ_{CR}=0.5	2.98

6 Conclusions

Mutation operation plays a fundamental role in DE and has been increasingly studied over the years. In this paper, a triangular Gaussian bare-bones mutation strategy is proposed, which uses the physical position and the fitness differences between three vectors (the current vector, the global best vector and the randomly selected vector) in a triangular topology. DE with this strategy (called TGBDE) is less likely to get trapped into local minima while maintaining a high level of convergence speed. TGBDE is further improved by cooperating with the popular "DE/rand/1" strategy to balance the exploration and exploitation behaviour. The key parameters in the improved version of TGBDE (ITGBDE) have been deeply investigated on a wide range of test problems, deriving the best parameter values. TGBDE and ITGBDE have been also compared with on a number of state-of-the-art DEs, and empirical studies have demonstrated that the proposed mutation strategies are promising for improving the performance of DE.

Like many DE variants, the proposed strategies are not free from parameters, which may limit their wide application. Encouraged by promising performance, we would like to focus on reducing parameters from ITGBDE in our future work.

Acknowledgment

This work was supported by National Natural Science Foundation of China (61501198) Wuhan Youth Science and Technology Chenguang program2014072704011248 Natural Science Foundation of Hubei Province (2014CFB461).

Jinglei Guo declares that she has no conflict of interest. Yong Wu declares that he has no conflict of interest. Wei Xie declares that he has no conflict of interest. ShouYong Jiang declares that he has no conflict of interest.

This article does not contain any studies with human participants or animals performed by any of the authors.

References

Awad NH, Ali MZ, Suganthan PN, Reynolds RG(2016), An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In Proceedings of IEEE Congress on Evolutionary Computation, pp. 2958–2965.

Brest J, Greiner S, Bošković 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

Brest J, Maučec M S(2009), Population size reduction for the differential evolution algorithm. Applied Intelligence 29(3):228–247

Darwin C(2009), On the Origin of the Species By Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life. Penguin Classics, London

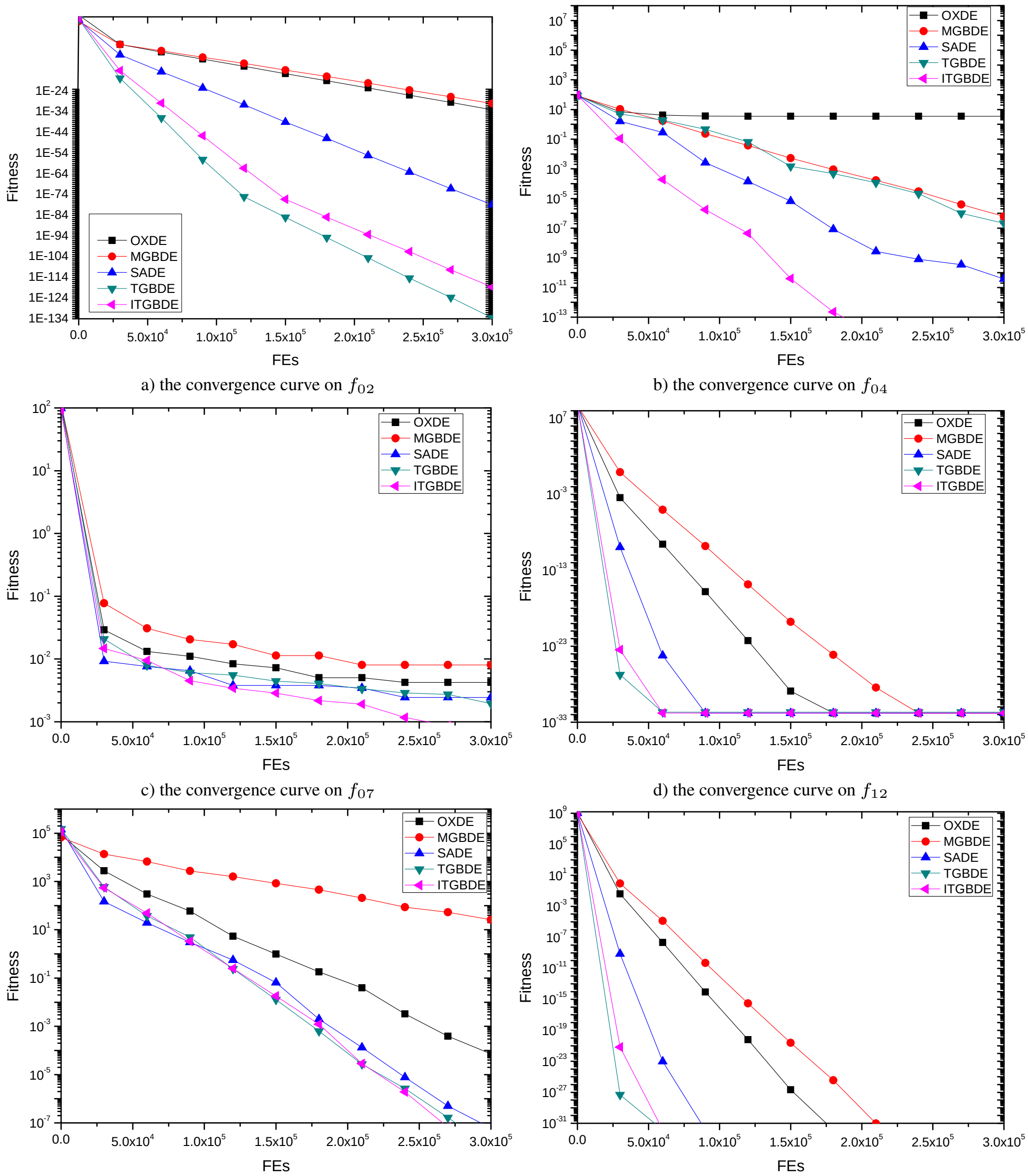
Das S, Abraham A, Chakraborty U K, Konar A(2009) Differential evolution using a neighborhood-based mutation operator. IEEE Trans. Evol. Comput. 13(3):526–553

Table 4 The results obtained by OXDE, MGBDE, SADE, TGBDE and ITGBDE for D=30

	OXDE	MGBDE	SADE	TGBDE	ITGBDE
	Mean± Std	Mean± Std	Mean±Std	Mean± Std	Mean ± Std
F_{01}	1.68E-59 ± 4.09E-59 +	2.49E-75 ± 1.36E-74 +	2.51E-131± 6.33E-131 +	**1.38E-185 ± 0.00E+00** -	2.02E-174 ±0.00E+00
F_{02}	2.39E-33 ± 1.97E-33 +	2.08E-43 ± 9.81E-43 +	1.06E-79 ± 1.15E-79 +	**3.78E-134 ± 8.84E-134** -	8.52E-121 ± 7.60E-121
F_{03}	3.09E-05 ± 4.70E-05 +	7.10E+00 ± 9.10E+00 +	8.92E-07 ± 2.15E-06 +	9.60E-08 ± 1.38E-07 =	**6.70E-08 ± 7.52E-08**
F_{04}	7.17E+00 ± 3.49E+00 +	1.43E-12 ± 3.43E-12 +	1.21E-11 ± 5.14E-11 +	5.94E-08 ± 2.19E-07 +	**1.41E-18 ± 7.08E-18**
F_{05}	**9.30E-01 ± 1.71E+00** -	1.55E+01 ± 1.14E+01 +	2.72E+01 ± 1.98E+01 +	3.02E+01 ± 2.69E+01 +	8.82E+00± 6.04E+00
F_{06}	**0.00E+00 ± 0.00E+00** =	**0.00E+00 ± 0.00E+00** =	**0.00E+00 ± 0.00E+00** =	2.33E-01 ± 7.74E-01 +	**0.00E+00 ± 0.00E+00**
F_{07}	4.26E-03 ± 1.69E-03 +	2.50E-03 ± 7.04E-04 +	2.27E-03 ± 1.05E-03 +	1.33E-03 ± 5.16E-04 =	**1.21E-03 ± 4.08E-04**
F_{08}	**-1.26E+04 ± 5.14E+01** -	-1.22E+04 ± 3.14E+02 +	**-1.26E+04 ± 2.16E+01** +	-1.20E+04 ± 2.83E+02 +	-1.25E+04 ± 7.41E+01
F_{09}	1.07E+01 ± 3.41E+00 +	4.44E+00 ± 1.82E+00 =	**3.32E-02 ± 1.82E-01** -	5.07E+00 ± 5.16E-04 =	4.05E+00 ± 2.07E+00
F_{10}	**3.55E-15** ± 0.00E+00 =	6.51E-15 ± 3.11E-15 +	**3.55E-15 ± 0.00E+00** =	6.95E-02 ± 2.66E-01 +	**3.55E-15 ± 0.00E+00**
F_{11}	2.14E-03 ± 4.42E-03 =	**0.00E+00 ± 0.00E+00** -	8.51E-03 ± 1.50E-02 -	3.45E-03 ± 7.58E-03 =	1.40E-03 ± 3.22E-03
F_{12}	2.07E-02 ± 7.89E-02 =	**1.57E-32 ± 5.89E-35** =	3.46E-03 ± 1.89E-02 =	3.46E-03 ± 1.89E-02 =	**1.57E-32± 5.57E-48**
F_{13}	3.66E-04 ± 2.01E-03 =	**1.35E-32 ± 5.57E-48** =	7.32E-04 ± 2.79E-03 =	1.61E-32 ± 1.44E-32 =	**1.35E-32 ± 5.57E-48**
F_{14}	6.57E-33 ± 3.60E-32 =	**0.00E+00 ± 0.00E+00** =	1.64E-33 ± 9.00E-33 =	1.64E-31 ± 9.00E-31 =	**0.00E+00 ± 0.00E+00**
F_{15}	1.31E-05± 1.42E-05 +	7.86E+00 ± 5.57E-48 +	6.69E-06 ± 3.25E-05 +	8.56E-08 ± 1.05E-07 =	**5.97E-08 ± 7.33E-08**
F_{16}	1.12E-04 ± 1.25E-04 +	8.99E+00 ± 1.38E+01 +	3.31E-01 ± 1.22E+00 +	7.63E+00 ± 2.18E+01 +	**3.73E-06 ± 5.19E-06**
F_{17}	3.53E-03 ± 6.86E-03 +	**0.00E+00 ± 0.00E+00** =	8.52E-03± 1.22E-02 +	1.89E-03 ± 3.97E-03 +	**0.00E+00 ± 0.00E+00**
F_{18}	3.10E-02 ± 1.70E-01 =	7.11E-15 ± 1.62E-15 +	6.21E-02 ± 2.36E-01 =	1.24E-14 ± 3.58E-15 +	**3.55E-15 ± 0.00E+00**
F_{19}	3.46E-03 ± 1.89E-02 =	1.59E-32 ± 5.57E-48 =	1.38E-02 ± 4.50E-02 =	6.91E-03 ± 2.63E-02 +	**1.57E-32 ± 5.57E-48**
F_{20}	3.66E-04 ± 3.66E-04 =	**1.35E-32 ± 5.57E-48** =	7.32E-04 ± 2.79E-03 =	2.93E-03 ± 8.62E-03 =	**1.35E-32 ± 5.57E-48**
w	9	11	10	9	–
l	2	1	2	2	–
t	9	8	8	9	–

Table 5 The results obtained by OXDE, MGBDE, SADE, TGBDE and ITGBDE for D=50

	OXDE	MGBDE	SADE	TGBDE	ITGBDE
	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
F_{01}	1.75E-27 ± 2.06E-27 +	4.11E-68 ± 2.24E-67 +	3.71E-119 ± 9.29E-119 +	8.84E-162 ± 3.54E-161 +	**3.36E-164** ± **0.00E+00**
F_{02}	9.84E-16 ± 5.06E-16 +	8.26E-42 ± 4.39E-41 +	6.00E-81 ± 1.25E-80 +	**6.32E-127** ± **9.88E-127** -	3.75E-117 ± 3.93E-117
F_{03}	7.40E+01 ± 3.72E+01 +	1.60E+03 ± 1.10E+03 +	**3.30E-03** ± **3.98E-03** -	8.85E-03 ± 7.94E-03 -	2.87E-01 ± 2.00E-01
F_{04}	9.60E+00 ± 2.49E+00 +	1.03E-05 ± 7.52E-06 +	6.20E-03 ± 2.50E-02 +	1.38E-06 ± 2.23E-06 +	**1.64E-08** ± **6.97E-08**
F_{05}	**1.08E+01** ± **1.21E+01** -	3.86E+01 ± 2.54E+01 +	6.47E+01 ± 3.61E+01 +	4.64E+01± 3.05E+01 +	2.53E+01 ± 1.76E+01
F_{06}	**0.00E+00** ± 0.00E+00 =	**0.00E+00** ± **0.00E+00** =	**0.00E+00** ± **0.00E+00** =	2.67E+00 ± 5.55E+00 +	**0.00E+00** ± **0.00E+00**
F_{07}	9.11E-03 ± 2.54E-03 +	4.66E-03 ± 1.24E-03 +	9.08E-03 ±2.25E-03 +	2.46E-03 ± 9.44E-04 =	**2.16E-03** ± **6.69E-04**
F_{08}	-2.09E+04 ± 3.70E-12 =	-1.98E+04 ± 5.57E+02 +	-2.09E+04 ± 3.70E-12 =	-1.91E+04 ± 4.59E+02 +	**-2.08E+04** ± **1.91E+02**
F_{09}	1.59E+01 ± 3.38E+00 +	2.22E+01 ± 6.68E+00 +	**6.96E-01** ± **9.11E-01** -	1.78E+01 ± 6.19E+00 +	1.24E+01 ± 4.05E+00
F_{10}	1.10E-14 ± 4.97E-15 +	1.02E-14 ± 3.58E-15 +	1.18E+00 ± 4.69E-01 +	2.66E-01 ± 5.24E-01 +	**7.11E-15** ± **0.00E+00**
F_{11}	2.37E-03 ± 8.03E-03 =	**7.39E-04** ± **2.83E-03** =	3.85E-03 ± 9.47E-03 =	6.29E-03 ± 1.46E-02 +	8.22E-04 ± 2.53E-03
F_{12}	6.22E-03 ± 2.50E-02 +	**9.42E-33** ± **9.49E-33** =	1.25E-02 ± 5.76E-02 +	6.29E-03 ± 2.70E-02 +	**9.42E-33** ± **1.39E-48**
F_{13}	3.66E-04 ± 2.01E-03 +	1.45E-32 ± 3.75E-33 =	3.66E-04 ± 2.01E-03 +	5.70E-02 ± 2.28E-01 +	**1.35E-32** ± **5.57E-48**
F_{14}	1.35E-27 ± 1.68E-27 +	4.60E-32 ± 1.53E-31 =	**2.30E-32** ± **4.24E-32** =	1.40E-30 ± 2.54E-30 +	4.40E-31 ± 2.30E-30
F_{15}	7.44E+01 ± 3.46E+01 +	1.55E+03 ± 8.62E+02 +	**4.24E-32** ± **3.55E-03** -	1.29E-02 ± 2.14E-02 -	2.16E-01 ± 1.42E-01
F_{16}	1.24E+02 ± 5.88E+01 +	1.81E+03 ± 9.98E+02 +	1.33E+02 ± 1.73E+02 +	2.37E+02 ± 1.81E+02 +	**3.17E+00** ± **2.83E+00**
F_{17}	4.11E-04 ± 2.25E-03 =	8.22E-04 ± 2.53E-03 =	8.56E-03 ± 1.94E-02 +	8.98E-03 ± 2.58E-02 +	**0.00E+00** ± **0.00E+00**
F_{18}	1.02E-14 ± 5.26E-15 +	1.09E-14 ± 4.47E-15 +	1.26E+00 ± 4.67E-01 +	2.90E-01 ± 5.10E-01 +	**7.11E-15** ± **0.00E+00**
F_{19}	1.66E-02 ± 5.88E-02 +	9.52E-33 ± 2.43E-34 +	1.45E-02 ± 4.81E-02 +	1.04E-02 ± 2.87E-02 +	**9.42E-33** ± **1.39E-48**
F_{20}	1.52E-25 ± 6.32E-25 +	3.66E-04 ± 2.01E-03 +	5.43E-02 ± 2.91E-01 +	6.67E-02 ± 2.92E-01 +	**1.35E-32** ± **5.57E-48**
w	15	14	13	16	-
l	1	0	3	3	–
t	4	6	4	1	–



a) the convergence curve on f_{02} b) the convergence curve on f_{04}

c) the convergence curve on f_{07} d) the convergence curve on f_{12}

e) the convergence curve on f_{15} f) the convergence curve on f_{20}

Fig. 6 The convergence curves of OXDE, MGBDE, SADE, TGBDE, and ITGBDE on some selected functions.

Table 6 The average rankings of OXDE, MGBDE, SADE, TGBDE, and ITGBDE

Algorithms	Ranking
OXDE	3.56
MGBDE	3.03
SADE	3.30
TGBDE	3.46
ITGBDE	**1.65**

Das S, Suganthan P N(2011) Differential evolution: A survey of the state-of-the-art. IEEE Trans. Evol. Comput. 15(1):4–31

Das S, Mullick SS, Suganthan PN(2016), Recent advances in differential evolution – An updated survey. Swarm and Evolutionary Computation 27(1):1–30

Epitropakis M G, Tasoulis D K, Pavlidis N G, Vrahatis M N(2011) Enhancing differential evolution utilizing proximity-based mutation operators. IEEE Trans. Evol. Comput. 15(1):99–119

Fan HY, Lampinen J(2003), A trigonometric mutation operation to differential evolution. Journal of global optimization 27(1):105–129

Gamperle R, Muller SD, Koumoutsakos P(2002), A parameter study for differential evolution. in: Procedings of Internationaal Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, pp. 293–298

García S, Molina D, Lozano M, Herrera F(2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. Journal of Heuristics 15(6):617–644

García S, Fernández A, Luengo J, and Herrera F(2010), Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inform. Sci. 180(10):2044–2064

Ghosh A, Das S, Chowdhury A, and Giri R(2011), An improved differential evolution algorithm with fitness-based adaptation of the control parameters. Inform. Sci. 181(18):3749–3765

Gong W, Z. Cai(2013), Differential evolution with ranking-based mutation operators. IEEE Trans. Cybern. 43(6):2066–2081

Guo SM, Yang CC, Hsu PH, Tsai JC(2014), Improving differential evolution with successful-parent-selecting framework. IEEE Trans. Evol. Comput. 19(5):717–730

Kennedy J(2003), Bare bones particle swarms. In Proceedings of IEEE Swarm Intelligence Symposium , pp. 80–87

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

Price K, Storn RM, Lampinen JA(2006), Differential evolution: a practical approach to global optimization. Springer Science & Business Media

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

Rahnamayan S, Tizhoosh HR, Salama M, Opposition-based differential evolution. IEEE Trans. Evol. Comput. 12(1):64–79

R. Storn and K. Price(1995), Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces, vol. 3, ICSI Berkeley

Tanabe R and Fukunaga AS(2013), Success-history based parameter adaptation for differential evolution. In Proceedings of 2013 IEEE Congr. Evol. Comput., pp.71–78.

Tanabe R and Fukunaga AS(2014), Improving the search performance of shade using linear population size reduction. In Proceedings of 2014 IEEE Congr. Evol. Comput., pp.1658–1665.

Wang H, Wu ZJ, Liu Y, Jiang DZ, Chen LL(2009), Space transformation search: a new evolutionary technique. In Proceedings of 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 537–544.

Wang H, Rahnamayan S, Sun H, Omran MG(2013), Gaussian bare-bones differential evolution. IEEE Trans. Cybern., 43(2):634–647

Wang Y, Cai Z, Zhang Q(2011), Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans. Evol. Comput. 15(1):55–66

Wang Y, Cai ZX, Zhang Q(2012), Enhancing the search ability of differential evolution through orthogonal crossover. Inform. Sci. 185(1):153–177

Wang Y, Liu ZZ, Li J, Li HX, Yen GG(2016), Utilizing cumulative population distribution information in differential evolution. Applied Soft Computing, 48(1):329–346

Yao X, Liu Y, Lin G(1999), Evolutionary programming made faster. IEEE Trans. Evol. Comput. 3(2):82–102

Yildiz AR(2013), A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations. Applied Soft Computing. 13(3):1561–1566

Yu WJ, Shen M, Chen WN, Zhan ZH, Gong YJ, Lin Y, Liu O, Zhang J(2014), Differential evolution with two-level parameter adaptation. IEEE Trans. Cybern. 44(7):1080–1099

Zhang J and Sanderson AC(2009), JADE: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput., 13(5):945–958