An Improved Trigonometric Differential Evolution

Shuzhen Wan, Shengwu Xiong, Jialiang Kou

School of Computer Science and Technology, Wuhan University of Technology,
Wuhan430070, China
wanshuzhen@163.com

Abstract

Differential evolution is an efficient and powerful population-based stochastic technique capable of handling non-differentiable, non-linear and multi-modal objective functions. In order to improve its performance, this paper introduces a best-trigonometric mutation strategy and applies a crossover rate update strategy to the proposed algorithm. The performance of the proposed algorithm is investigated on a set of benchmark functions. The numerical experimental results show that the convergence rate of proposed algorithm is higher and the robustness of proposed algorithm is better than DE and TDE algorithm.

Keywords: Differential Evolution; Trigonometric Mutation; Crossover Update Strategy

1. Introduction

Differential evolution [1] is an efficient and simple algorithm for global optimization in various fields. It has been successfully applied to solve optimization problems in benchmark functions [2, 3] and real-world applications [4, 5]. And it performs better than genetic algorithm (GA)[6], and particle swarm algorithm (PSO)[7] over several benchmarks[8]. There exist many trail vector generation strategies in the DE algorithm which a few can solve some particular problems. There are three control parameters involved in DE, i.e., population size NP, scale factor, crossover rate which can greatly influence the optimization performance of DE. Recently, some researchers have been working hardly on enhancing the performance of DE[9]. In[10], Gamperle studied the Experimental parameter empirical parameters setting of DE. In[11], Zaharie analyzed the relationship between population diversity and control parameters.

Fan and Lampinen proposed a new DE variant, namely trigonometric mutation differential evolution (TDE [12]). They introduced a new local search operation, trigonometric mutation, and embedded them into DE. The trigonometric mutation is a rather greedy operator which can greatly speed up the convergence rate of DE but such an operator usually has the danger of converging prematurely into local optima. For avoiding this phenomenon, they used the parameter (trigonometric mutation probability) to maintain good balance between convergence rate and quality of solution. This new variant of DE performances better on convergence rate over some benchmark functions. However, Fan and Lampinen point out that the trigonometric mutation operation biases the new trial individual strongly in the direction where the best one among three individuals chosen for the mutation is. The donor individuals for trigonometric mutation are randomly chosen within the population. The new trial individuals are prone to the best individual which is random selected from the whole population. If there one donor individual is best one of all individuals, so, some of the new trial individuals are prone to that individual. Thus, the convergence rate will be higher (this are proved in our experiments). In [13], Das noted that the exploitative mutation operator rapidly converged to a minimum of the objective function, but had the danger exist of premature convergence to a suboptimal solution. The aforementioned method of the modified trigonometric mutation operator can seem as an exploitative mutation operator. Therefore, it has the probability of leading the population converge to local optima. To solve these problems, some techniques have been proposed to tune the mutation strategy and parameters such as the population size [14, 15], the scale factor, and the crossover [16, 17].

In our paper, we introduce a best-trigonometric mutation to DE to boost the convergence process. We take the best individual as one of the donor vector and adopt the crossover update strategy to improve the population diversity and avoid trapping in local optima. In the crossover update strategy, the crossover rate of individuals is generated by the normal distribution. Experiments and comparisons
show that the proposed algorithm performs better than TDE and standard DE over 13 well-known numerical benchmark functions.

The rest of this paper is organized as follows: the conventional DE and TDE are reviewed in sections 2 and 3, respectively; section 4 describes the proposed algorithm; in section 5, experimental analysis is presented; in section 6, finally conclusions are drawn.

2. Differential Evolution

There are several variants of DE. In this paper, we shall follow the version of DE/rand/1/bin, which is apparently the most frequently used version, and is often considered as the “basic” version of the DE algorithm. The particular scheme is described as follows:

DE starts with a population of \( NP \) candidate solutions which can be represented as the \( D \)-dimensional parameter, \( X_{i,G} \), \( i = 1, 2, 3, \ldots, NP \), where \( i \) index denotes the \( i \)th individual of the population, \( G \) denotes the generation to which the population belongs. \( NP \) is the number of members in a population. Successive populations are generated by adding the weighted difference of two randomly selected vectors to a third randomly selected vector. This operation is the mutation operator, and then crossover operator is employed to generate new candidate vectors. A selection scheme is employed to determine whether the offspring or the parent survives to the next generation. The process is repeated until a termination criterion is reached. The details are described as follows.

2.1. Mutation

The mutation operation of DE applies the vector differentials between the randomly selected vectors from the population for making a perturbation to the individual subject to the mutation operation. At generation \( G \), for the \( i \)th target individual \( X_{i,G} \), the perturbed individual \( V_{i,G} \) is generated based on the three randomly selected individuals as follows:

\[
V_{i,G} = X_{i,G} + F \cdot (X_{r1,G} - X_{r2,G})
\]  

(1)

Where \( i = 1, \ldots, NP, r_1, r_2, r_3 \in \{1, \ldots, NP\} \) are randomly selected and satisfy: \( r_1 \neq r_2 \neq r_3 \neq i, F \in [0, 1] \), where \( F \) is the scale factor.

2.2. Crossover

The perturbed individual, \( V_{i,G+1} = (v_{i,j,G+1}, \ldots, v_{i,k,G+1}) \) and the target population member, \( X_{i,G} = (x_{i,j,G}, \ldots, x_{i,k,G}) \), are then subject to crossover operation to generate the population of candidate or “trial” vectors, \( U_{i,G+1} = (u_{i,j,G+1}, \ldots, u_{i,k,G+1}) \), as follow:

\[
u_{i,j,G+1} = \begin{cases} 
 v_{i,j,G+1} & \text{if} \ rand_j \leq CR \ or \ j = k \\
 x_{i,j,G} & \text{otherwise}
\end{cases}
\]

(2)

Where \( j = 1, \ldots, n, k \in \{1, \ldots, n\} \) is a random parameter’s index, chosen for each \( i \), and the crossover rate, \( CR \in [0,1] \), the other control parameter is set by user.

2.2. Selection

The selection operation selects, according to fitness value of the population vector and its corresponding trail vector, which vector will become a member of the next generation. Here, if we have the minimization problem, the following selection rule is used:
if \( \text{rand} \leq CR \) or \( j = k \)
\[
J_{i,G+1}^{i} = \begin{cases} 
J_{i,G+1}^{j} & \text{if rand} \leq CR \text{ or } j = k \\
J_{i,G}^{j} & \text{otherwise}
\end{cases}
\]

### 3. TDE

In TDE algorithm[12], the trigonometric mutation operation is embedded into the basic DE to improve the convergence rate. The main difference between DE and TDE is the way the mutation operation is performed. The trigonometric mutation operation is performed according to the following formulation:

\[
V_{i,G+1}^{i} = \left( X_{i,G}^{1} + X_{j,2,G}^{1} + X_{j,3,G}^{1} \right) / 3 + (p_2 - p_1) \left( X_{r_1,G}^{1} - X_{r_2,G}^{1} \right) \\
+ (p_3 - p_1) \left( X_{r_2,G}^{1} - X_{r_3,G}^{1} \right) + (p_1 - p_3) \left( X_{r_3,G}^{1} - X_{r_1,G}^{1} \right)
\]

Where for \( i = 1, 2, 3 \), \( p_i = \sqrt{f(X_{i,G}^{1})} / (f(X_{i,G}^{1}) + f(X_{2,G}^{1}) + f(X_{3,G}^{1})) \)

Thus, the trigonometric mutation operation is a greedy operator that for three given points generates an offspring by exploiting the most promising directions. The performance of this operator can offer an exploitative alternative to DE. The trigonometric mutation biases the offspring strongly in the optimal directions, so the convergence rate will be accelerated. However, such an operator usually leads to the greedy algorithm which is prone to converge prematurely into a local optimum, there need some methods to avoid this problem. In TDE, the mutation probability is used for this purpose. In our algorithm, we modified the mutation strategy with a best-trigonometric mutation to make a higher convergence rate, and we applied a self-adapting control parameter strategy for avoiding premature.

### 4. Improved Trigonometric DE

The trigonometric DE can make good performance on convergence because the trigonometric mutation is a rather greedy operator, which biases the new trial solution in the direction of the best one among the three donor individuals. Inspired from this, we assume that if one of the donor individuals is the best individual of the population, the new trial vector will converge to the best individual, so the convergence rate will be higher. We call this DE variant as best-one-TDE (BTDE). The following experiments are partly proved our idea. However, in some multimodal functions, BTDE performs poorly compared with TDE. After experimental analysis, we believe that though BTDE accelerates the convergence rate, the population may prematurely lose diversity and converge to local optima. Effective measures should be taken to reduce the probability of this happening.

In TDE, the crossover rate \( CR \) is fixed, so it weakens the robustness. There have been some crossover rate update strategies in literatures. In[18], Brest et al. used the random uniform distribution to update the \( CR \). Qin et al. adopted adaptive normal distribution to generate the new \( CR \) [19]. In order to use the simple method to produce good results, we apply the normal distribution to \( CR \) update. It is described as follows:

\[
CR_i = N_i(0.5,0.2)
\]

Where \( N_i(0.5,0.2) \) denotes the normal distribution number with mean 0.5 and standard deviation 0.2. The probability of \( CR \) lies within 0 and 1 is about 99%. This method increases the diversity of population and avoid prematurity. For simplicity, we call our algorithm as modified TDE (MTDE). The outline of our algorithm as follows:

- Step1: Initialize the population, choose \( F \) and \( M_i \), and determine the initial \( CR \).
- Step2: Mutation operation:
2.1) Perform the modified trigonometric mutation with Equation (4) with a probability $M_t$ (in which, one of the donor individuals is the best individual of the population).

2.2) Perform original DE’s mutation with Equation (1) with a probability $(1- M_t)$.

Step3: Crossover operation:
3.1) Update the crossover rate with Equation (5).
3.2) Perform the crossover operation with the updated $CR$.
Step4: Evaluate the population with the object function.
Step5: Selection.
Step6: Repeat step step2 to step5 until the termination criterion is satisfied.

The main difference between our algorithm and TDE is the mutation operation and the crossover rate update strategy. For proposed algorithm (MTDE), one of the three donor individuals is best individual of the population. Extensive experiments in section V verify the promising performance of the MTDE to handle the unimodal and multimodal problems.

5. Experimental Analysis

5.1. Experimental Settings

Experiments were conducted on a set of 13 benchmark functions which are detailed in literatures [18, 20] to evaluate the proposed algorithm and three other DE algorithms. For functions $f_1 - f_{13}$, 30-dimensional (30-D) were tested. The maximum number of function evaluations (FEs) is set to 150 000 for all functions.

The population size NP is set to 100 for all algorithms.

The algorithms in comparison are listed:
- Standard DE (DE/rand/1/bin): $F = 0.5$, $CR = 0.9$;
- TDE: $F = 0.5$, $CR = 0.9$, $M_t = 0.05$; [10]
- BTDE: $F = 0.5, CR = 0.9$; $M_t = 0.05$,
- MTDE: $F = 0.5$, $CR = 0.9$, $M_t = 0.05$.

5.2. Comparison with other DE algorithms

Table 1 is the t-test at $\alpha = 0.05$ by two tailed test which is adopted to compare the significance between two algorithms.

With respect to the quality of the final results of 30 runs on 13 benchmark functions, Table 1 indicates that on average the MTDE algorithm is able to obtain better results than DE and TDE on most of the test functions.

For the unimodal functions $f_1 - f_3$, $f_5$, MTDE is outperformed by BTDE, but performs much better than the other algorithms. It is shown that BTDE has the convergence faster than MTDE on some unimodal functions. It also can be observed from Fig.1 of functions $f_1, f_3$. MTDE makes the best solution on function $f_4$. On the functions $f_6 - f_{13}$, the MTDE algorithm is superior to all other algorithms. In addition, we can observe that proposed MTDE algorithm has the higher convergent rate and better results on almost all functions than DE and TDE. The BTDE performs well convergence over some unimodal test functions but poorly on the multimodal functions. It is clearly that the best-one-trigonometric mutation operation works effectively accompanied with the crossover rate update scheme in the MTDE algorithm.
Table 1. A comparison of experiment results for 30-dimensional problems over 30 independent runs.

<table>
<thead>
<tr>
<th>function</th>
<th>MTDE Mean</th>
<th>MTDE Best</th>
<th>MTDE Std</th>
<th>BTDE Mean</th>
<th>BTDE Best</th>
<th>BTDE Std</th>
<th>TDE Mean</th>
<th>TDE Best</th>
<th>TDE Std</th>
<th>DE Mean</th>
<th>DE Best</th>
<th>DE Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.61E-37</td>
<td>1.12E-36</td>
<td></td>
<td>6.05E-52</td>
<td>1.33E-51</td>
<td></td>
<td>7.08E-35</td>
<td>1.25E-34</td>
<td>1.33E-13</td>
<td>1.05E-13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4.01E-21</td>
<td>5.10E-21</td>
<td></td>
<td>4.51E-25</td>
<td>1.97E-24</td>
<td></td>
<td>1.07E-20</td>
<td>1.09E-20</td>
<td>6.47E-07</td>
<td>2.09E-07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.02E-05</td>
<td>2.67E-05</td>
<td></td>
<td>3.29E-11</td>
<td>8.64E-11</td>
<td></td>
<td>0.001548</td>
<td>0.002085</td>
<td>1.1874</td>
<td>0.849206</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.143756</td>
<td>0.121296</td>
<td></td>
<td>0.38415†</td>
<td>0.439965</td>
<td></td>
<td>5.51484†</td>
<td>2.06131</td>
<td>0.326761†</td>
<td>0.819955</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>11.9272</td>
<td>15.8269</td>
<td></td>
<td>3.40047†</td>
<td>3.5594</td>
<td></td>
<td>25.8918†</td>
<td>13.8743</td>
<td>16.5602†</td>
<td>0.756022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
<td>6.86667†</td>
<td>6.48464</td>
<td></td>
<td>0.333333†</td>
<td>0.606478</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.004320</td>
<td>0.00131†</td>
<td></td>
<td>0.007447†</td>
<td>0.002732</td>
<td></td>
<td>0.004041†</td>
<td>0.001471</td>
<td>0.009164†</td>
<td>0.002645</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-12569.5</td>
<td>2.68E-12</td>
<td></td>
<td>-11453.3†</td>
<td>497.51</td>
<td></td>
<td>-12356†</td>
<td>177.682</td>
<td>177.016†</td>
<td>14.3682</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8.88E-16</td>
<td>1.35E-15</td>
<td></td>
<td>30.8769†</td>
<td>8.75407</td>
<td></td>
<td>49.7699†</td>
<td>52.5618</td>
<td>177.016†</td>
<td>14.3682</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7.05E-15</td>
<td>1.60E-15</td>
<td></td>
<td>2.00175†</td>
<td>0.551663</td>
<td></td>
<td>0.093130†</td>
<td>0.284168</td>
<td>1.16E-07</td>
<td>4.30E-08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0.017829†</td>
<td>0.030366</td>
<td></td>
<td>0.002868†</td>
<td>0.007607</td>
<td>0.000410</td>
<td>0.002249</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1.60E-32</td>
<td>1.31E-33</td>
<td></td>
<td>0.378281†</td>
<td>0.538385</td>
<td></td>
<td>1.64E-32</td>
<td>3.77E-33</td>
<td>1.32E-14†</td>
<td>8.52E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1.35E-32</td>
<td>5.57E-48</td>
<td></td>
<td>0.124208†</td>
<td>0.656037</td>
<td></td>
<td>0.003296†</td>
<td>0.008728</td>
<td>8.52E-14†</td>
<td>4.80E-14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: † indicates our algorithm is significantly better than its competitor with 95% certainty by t-test. ‡ means that the corresponding algorithm is better than our algorithm (MTDE).
6. Conclusions

In this paper, we have proposed two strategies for improving the performance of TDE algorithm. The modified mutation operation strategy can accelerate the convergence rate of the proposed algorithm. And the crossover rate update strategy makes a balance between the fast convergence rate and trapping in local optima. The experimental results show that the proposed algorithm (MTDE) is more efficient than other algorithms not only on the convergence rate but also the robustness over 13 benchmark functions.

7. Acknowledgements

This work was supported by the National Natural Science Foundation of China under grant no. 61170202.

8. References