An Improved Differential Evolution Algorithm for Solving High Dimensional Optimization Problem

Chunfeng Song and Yuanbin Hou

The Institute of Electric and Control Engineering, Xi’an University of Science and Technology, Xi’an 710054 China

Abstract

In order to improve the weak situation of the global search ability, the stability and time consuming of optimization of differential evolution (DE) algorithm in solving high dimensional optimization problem, an improved differential evolution algorithm with multi-population and multi-strategy (MPMSIDE) is proposed to solve high dimensional optimization problem. Firstly, the different DE mutation strategies are studied. Then the MPMSIDE algorithm divides the population into several sub-populations, which evolve independently and communicate with each other at regular intervals by using different DE strategies, in order to save the computation time. And the improved mutation strategy and local optimization strategy are introduced to raise and balance the global searching ability and local searching ability, and improve the optimization efficiency. The self-adaptive update strategy is used to adjust the scaling factor and crossover factor for making the parameter sensitivity of DE algorithm and improving the stability and robustness. Finally, the proposed MPMSIDE algorithm is applied to standard test function optimization for verifying the effectiveness. The experimental results show that the proposed MPMSIDE algorithm has a relatively better optimization performance for solving complex optimization problem, and takes on remarkable optimizing ability, higher searching accuracy and faster convergence speed.

Keywords: differential evolution, multi-population, multi-strategy, control parameter, complex optimization problem

1. Introduction

The optimization method is divided into the traditional optimization methods and heuristic optimization methods. The traditional optimization methods mainly realize the order of single feasible solution and deterministic search based on the objective function gradient (or derivative) information. And the heuristic optimization methods are a kind of bionic algorithm, which realizes the parallel and stochastic optimization of multi solutions by using the heuristic strategy. The heuristic search algorithms do not require the continuous and differentiable information of the objective function, and take on better global search ability [1, 2]. So they has become a hot topic in the optimization area.

In these heuristic optimization methods, differential evolution (DE) algorithm is a heuristic random search algorithm based on population differences. The DE algorithm was proposed by Storn in 1995, to solve Chebyshev polynomial problem, and complex optimization problems subsequently [3]. The DE algorithm has a very special connection with evolutionary algorithms. The DE algorithm and particle swarm optimization (PSO) algorithm [4] are optimization algorithms based on the swarm intelligence theory. They use the generated swarm intelligence of the competition and cooperation between individuals within the population to guide optimization search. But compared to the evolutionary algorithm, the DE algorithm retains the global search strategy based on the population, uses the real-coding, simple differential mutation and competitive survival strategy to reduce the complexity of the genetic operation. At the same time, the unique
remembering ability of DE algorithm can dynamically track the current search, in order to adjust its search strategy. The DE algorithm has a strong global convergence and robustness, and does not require the character information of solving problem. It is suitable to solve the optimization problems in the complex environment, which can not be solved by using conventional mathematical programming methods. So the DE algorithm is an efficient parallel search algorithm, theory and an application research has important academic significance and engineering value.


These improved DE algorithms overcome the premature convergence and falling into local optimum problems. But the local search ability, convergence speed and optimization accuracy of the algorithm still require to be further strengthened. So an improved differential evolution algorithm with multi- population and multi-
strategy (MPMSIDE) is proposed to solve high dimensional optimization problem is proposed in this paper.

2. Differential Evolution

The DE algorithm is a computational intelligent method by simulating biological evolution of natural population. The main idea is: the cooperation, competition and the generational evolution and reproduction of individuals in a population are used to improve the adaptation degree of individuals to the external environment, so as to approximate the optimal solution of the problem. In essence, the DE algorithm is a greedy genetic algorithm with high quality thought based on real coding. The definition of optimization problem and mathematical describing of the DE algorithm are given:

For optimization problem:

\[
\min f(x_1, x_2, \ldots, x_d) \quad \text{s.t. } \quad x^L_j \leq x_j \leq x^U_j
\]

where \( j = 1, 2, 3, \ldots, D \), \( D \) is the dimension of the solution space. \( x^L_j \) and \( x^U_j \) respectively represent the upper and lower bounds of the \( j^{th} \) component \( x_j \).

The flow of the DE algorithm is shown in Figure 1.

![Figure 1. The Flow of DE Algorithm](image)

2.1. Initialization

The key parameters of DE algorithm are initialized. These parameters include the population size (\( N \)), the mutation factor (\( F \)), the crossover rate (\( CR \)) and the stopping criterion (\( T \)). The initial generation counter is set as \( t = 1 \). Each individual is encoded as a vector of floating-point numbers. And the prescribed upper and lower bounds of each decision variable with random generated values are initialized according to the uniform probability distribution in the N-dimensional problem.
space. The following equation is used to initialize the initial population of individuals:

\[ x_j(0) = x_j^l + rand(0,1) \times (x_j^u - x_j^l) \quad (j = 0,1,2,\cdots,D) \quad (2) \]

### 2.2. Mutation

The DE algorithm executes the individual mutation for target vector by using differential strategy. The mutation component is the different vector of the parent. Each vector consists of two different individuals \((x'_k, x'_j)\). According to the different generation method of mutation individual, there proposed some different DE algorithms.

1. DE/rand/1/bin
   \[ x_k = x^l_k + F \times (x^u_k - x^l_k) \quad (3) \]
2. DE/rand/2/bin
   \[ x_k = x^l_k + F \times [(x^u_k - x^l_k) + (x^u_j - x^l_j)] \quad (4) \]
3. DE/best/1/bin
   \[ x_k = x^g_{best} + F \times (x^u_k - x^l_k) \quad (5) \]
4. DE/rand/2/bin
   \[ x_k = x^g_{best} + F \times [(x^u_k - x^l_k) + (x^u_j - x^l_j)] \quad (6) \]
5. DE/current-to-best/1
   \[ x_k = x^l_k + F \times [(x^u_k - x^l_k) + (x^u_{best} - x^l_{best})] \quad (7) \]

### 2.3. Crossover

The crossover operation is used to construct an offspring by mixing current components. In order to enhance the potential diversity of the population, crossover operation plays a key role. The essence of crossover operation is to execute the uniform crossover between the generated individual \((x_k)\) in the mutation and the \(i^{th}\) individual \((x_j')\) in the population in order to compensate mutation search in the previous step to generate test vector \((x_G)\). It includes the binomial cross method and index cross method. The binomial cross method is used. The specific crossover operator equation is given:

\[
x_G = \begin{cases} x_{ij} & \text{rand}(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{ij}' & \text{other} \end{cases} \quad (j = 1,2,\cdots,D) \quad (8)
\]

where \(j_{rand} \in \{1,2,\cdots,D\}\) is a random integer, which is used to ensure that at least one component in target individuals \(x'_j\) makes the crossover operation. \(x_{ij}'\) stands for the \(i^{th}\) individual of \(j^{th}\) real-valued vector, \(x_{ij}\) stands for the \(i^{th}\) individual of \(j^{th}\) real-valued vector of a mutant vector, \(\text{rand}(0,1)\) is the \(j^{th}\) evaluation of a uniform random number generation with\([0,1]\), \(CR\) (crossover rate) is \([0,1]\).

### 2.4 Selection

The selection operator is used to construct a population by selecting the trial vectors and their predecessors according to the better fitness value or the optimal value. The DE algorithm generates the offspring by using the greedy selection strategy. After the crossover operation is executed, the test individual will be
competed between \( x_G \) and \( x_i^t \). The individual with the better fitness value will be selected as the offspring. If the objective function is to be minimized, the selected operation equation is given:

\[
    x_{i}^{t+1} = \begin{cases} 
    x_G & f(x_G) < f(x_i^t) \\
    x_i^t & f(x_i^t) \geq f(x_G)
    \end{cases}
\]  

(9)

3. An Improved DE Algorithm with Multi-population and Multi-Strategy (MPMSIDE)

3.1. The Idea of MPMSIDE Algorithm

The DE algorithm a evolutionary algorithm based on real coding, the mutation operation, crossover operation and selection operation of the individual are used to constantly evolve the population until the algorithm terminates. Due to the weak situation of the global search ability, the stability and time consuming of optimization of differential evolution (DE) algorithm in solving high dimensional optimization problem, this paper proposes an improved differential evolution algorithm with multi-population and multi-strategy (MPMSIDE) for solving high dimensional optimization problem. In the proposed MPMSIDE algorithm, according the fitness value, standard deviation of fitness and distance of each individual, the population is divided into three different subpopulations, which are best population with the better fitness of individuals, worst population the poor fitness of individuals and general population with the rest individuals. The best population is responsible for local search and improves the convergence speed and precision. The worst population is responsible for global search, jumps out the local optimum and avoids premature convergence. The general population is responsible for balancing the global search ability and local search ability. The mutation is the key operation of DE algorithm; the selected mutation strategy determines population direction in the process of evolution. The improved mutation strategy is introduced to enhance the global optimization ability of the greedy algorithm, avoid possible stagnation in a local minimum value for dealing with complex functions with high dimension multimodal optimization problems, and the premature loss of population diversity. The local optimization strategy is used to avoid the local extreme point and improve the local hill-climbing ability in the local search. The self-adaptive update strategy determines the similarity between the best individual and the general individual according to the individual similarity coefficient for reducing the adverse effects of the linear adjusting scaling factors and making the parameter sensitivity of DE algorithm and improving the stability and robustness.

The improved mutation strategy DE/best/1 is given:

\[
    x_i = x_{g\text{best}}^t + F_2 \times (x_{g1}^t - x_{g2}^t) + \alpha \times F_2 \times (x_{g3}^t - x_{g4}^t)
\]  

(10)

where \( \alpha \) is correlation parameter of vector. The added \( \alpha \) can give more choice space in this mutation. The improved mutation strategy can greatly enhance the searching space and the local search ability of the algorithm. The MPMSIDE algorithms increases a number of control parameters and the population, and effectively enhance the searching space the local search ability and provides a more flexible choice than standard DE algorithm.

3.2. The Flow of MPMSIDE Algorithm
On the basis of the above idea, the flow of an improved differential evolution algorithm with multi-population and multi-strategy (MPMSIDE) in solving high-dimensional optimization problem is shown in Figure 2.

4. The Results of Testing and Analyzing MPMSIDE Algorithm

The famous Benchmarks functions are selected to test the performance of the MPMSIDE algorithm in this paper. For example, Rosenbrock function, Noisy function, Schwefel1.2 function, Step function, Schwefel2.21 function and so on. In order to make the MPMSIDE algorithm with comparative results, the MPMSIDE algorithm is compared with the DE algorithm and CDE algorithm. The experiment works on Intel(R) Core i5-4200U, 2.40GHz, 2G RAM, Windows 8 and Matlab 2012. The experimental parameters are given: population size $N = 66$, crossover probability factor $CR = 0.9$, scaling factor $F = 0.5$, the functional dimension is 30, the maximum evolution generation $T_{\text{max}} = 1000$. The specific formula and variables’ range of all functions are shown in Table 1.

Table 1. Benchmarks Testing Functions
The MPMSIDE algorithm, DE algorithm and CDE algorithm run independently with 30 times for five functions. The maximum value, minimum value and mean value and standard deviation are used to evaluate three algorithms. The results are shown in Table 2.

Table 2. The Results for the DE, CDE and MPMSIDE Algorithms

<table>
<thead>
<tr>
<th>Function</th>
<th>Algorithm</th>
<th>The optimal value</th>
<th>Maximum value</th>
<th>Minimum value</th>
<th>Mean best value</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>f₁</td>
<td>DE</td>
<td>3.143 ±0.00</td>
<td>6.433 ±0.02</td>
<td>4.357 ±0.01</td>
<td>1.931 ±0.00</td>
<td></td>
</tr>
<tr>
<td>f₂</td>
<td>CDE</td>
<td>0.942 ±0.04</td>
<td>1.364 ±0.07</td>
<td>3.422 ±0.08</td>
<td>4.430 ±0.08</td>
<td></td>
</tr>
<tr>
<td>f₂</td>
<td>MPMSIDE</td>
<td>7.042 ±1.12</td>
<td>5.032 ±1.14</td>
<td>8.322 ±1.13</td>
<td>6.669 ±1.13</td>
<td></td>
</tr>
<tr>
<td>f₃</td>
<td>DE</td>
<td>5.332 ±5.33</td>
<td>7.511 ±5.33</td>
<td>1.328 ±0.01</td>
<td>6.730 ±0.01</td>
<td></td>
</tr>
<tr>
<td>f₃</td>
<td>CDE</td>
<td>4.341 ±4.04</td>
<td>3.983 ±4.04</td>
<td>7.034 ±4.03</td>
<td>6.322 ±4.03</td>
<td></td>
</tr>
<tr>
<td>f₃</td>
<td>MPMSIDE</td>
<td>2.668 ±0.03</td>
<td>1.468 ±0.04</td>
<td>5.105 ±0.04</td>
<td>3.471 ±0.04</td>
<td></td>
</tr>
<tr>
<td>f₄</td>
<td>DE</td>
<td>3.167 ±2.20</td>
<td>5.287 ±2.24</td>
<td>3.145 ±2.21</td>
<td>4.203 ±2.22</td>
<td></td>
</tr>
<tr>
<td>f₄</td>
<td>MPMSIDE</td>
<td>6.846 ±2.32</td>
<td>3.690 ±2.32</td>
<td>4.480 ±2.30</td>
<td>4.364 ±2.24</td>
<td></td>
</tr>
<tr>
<td>f₅</td>
<td>CDE</td>
<td>0.000 ±0.00</td>
<td>0.000 ±0.00</td>
<td>0.000 ±0.00</td>
<td>0.000 ±0.00</td>
<td></td>
</tr>
<tr>
<td>f₅</td>
<td>MPMSIDE</td>
<td>5.036 ±4.02</td>
<td>7.299 ±4.03</td>
<td>9.036 ±4.02</td>
<td>1.693 ±4.03</td>
<td></td>
</tr>
</tbody>
</table>

As can be seen from the Table 2, for the five famous Benchmarks functions with the MPMSIDE algorithm, the MPMSIDE algorithm is better optimization performance than DE algorithm for solving f₁, f₂, f₃, f₄ and f₅ functions. The MPMSIDE algorithm is better optimization performance than CDE algorithm for solving f₁, f₃, f₄ and f₅ functions. For f₄ function, The MPMSIDE algorithm obtains the optimal value (zero). So the proposed MPMSIDE algorithm takes on the better global convergence ability and searching precision in solving high dimensional optimization problems.

5. Conclusion

DE algorithm is an efficient and powerful algorithm for solving complex optimization problems. It depends on the mutation strategy and crossover strategy, and the values of the associated control parameters. And for one complex optimization problem, the different mutation strategy and crossover strategy may be more effective than single mutation strategy and crossover strategy. So an improved differential evolution algorithm with multi-population and multi-strategy (MPMSIDE) is proposed to solve high dimensional optimization problem in this paper. In the MPMSIDE algorithm, the population is divided into best population, worst population and general population according the fitness value, standard deviation of fitness and distance of each individual. The improved mutation strategy is introduced to enhance the global optimization ability.
of the greedy algorithm, avoid possible stagnation in a local minimum value for dealing with complex functions with high dimension multimodal optimization problems. The local optimization strategy is used to avoid the local extreme point and improve the local hill-climbing ability in the local search. The self-adaptive method is used to automatically adjust the scaling factor and crossover factor during the running time. The performance of MPMSIDE algorithm is evaluated on benchmark problems and is favorably compared with DE, CDE algorithm in the literature. The results demonstrate that the proposed MPMSIDE algorithm is overall more effective and takes on better searching precision, convergence speed, and global convergence ability.

ACKNOWLEDGEMENTS

This research was supported by the Youth Science Foundation (51405381) and the Special Foundation for Scientists of Xi’an university of science and technology (201314).

References


Authors

**Chunfeng Song.** He was born in Nanyang city, the province of Henan in July, 1977, who graduated from the University of Science and Technology and obtained the Master of Engineering degree in 2005. He is majoring in the research about the control theory and control engineering.

**Yuanbin Hou.** She was born in November, 1953, graduated from Xi’an Jiaotong University and obtained the Doctor of Engineering degree in 1997. She is majoring in the research about the control theory and control engineering.