

# Research on the Multi-population Differential Evolution Algorithm and the Performance

Renquan Huang, Jing Tian, Juanjuan Wang and Junmin Kang  
 School of Economy and Finance, Xi'an International Studies University, Xi'an, China

**Abstract**—The original differential evolution algorithm (DE) is a single-population differential evolution algorithm (SPDE). DE converges very quickly, and takes the advantage of robustness. The improved DE has a better performance, but there are premature problems in optimizing complex problems. The multi-population differential evolution algorithm (MPDE) is proposed to overcome premature problems in this paper. The optimal substitution strategy (OSS) and the elite immigration strategy (EIS) are studied to maintain the diversity of populations. The simulation concludes that MPDE converges faster than SPDE in optimizing the ultra-high dimensional problems, and the EIS is superior to the OSS. However, the efficiency of DE is more effective than that of MPDE when the algorithms converge. Research shows that multi-population strategy is a feasible and effective way to the premature problems of DE.

**Keywords**—multi-population differential evolution algorithm; optimization algorithm; optimal substitution strategy; elite immigration strategy

## I. INTRODUCTION

In 1995, the differential evolution algorithm (DE) was proposed by Rainer Storn and Kenneth Price in [1] [2]. DE has the advantages of faster convergence, less parameters to control and stronger robustness than other optimization algorithms. DE which is real-coded, surpasses the current popular random algorithms in terms of convergence speed and robustness. Especially, the optimization performance of DE is improved when the adaptive strategy and local enhanced operator are used in [3]. However, the single-population differential evolution (SPDE) algorithm is effective in optimizing the simple problems. It has some limitations when the problems are complex. The premature problem is the most serious of them [4][6]. The main reasons of the premature problem are as follows:

- *Differences of individual fitness value.* When the fitness value of a certain individual is much better than others, the population will be controlled by the individual, and the algorithm will fail to find the global optimal solution.
- *The effect of the population size  $n$ .* When the population size  $n$  is too small, the diversity of the population will be too simple. So, the individual differences of the population are not obvious, and there is almost no competition between individuals, which will lead to the evolution is slow or even terminated.

When the population size  $n$  is too large, it will lead to a sharp increase in computation.

- *Terminal condition.* The maximum evolutionary algebra is one of the common termination conditions. If the maximum evolutionary algebra is set too small, it will lead to premature problems.

The premature problem is a serious problem that cannot be ignored in DE. The appropriate measures which effectively prevent premature phenomenon in the process of evolution, should be taken to maintain the diversity of the population. In this regard, we propose the MPDE in this paper. The random initial population is divided into several sub-populations in MPDE, and each sub-population (group) runs DE. To maintain the diversity of the population, the optimal substitution strategy (OSS) and the elite immigration strategy (EIS) are introduced to MPDE, which overcome the premature problem effectively.

## II. THE IMPROVED DIFFERENTIAL EVOLUTION ALGORITHM

DE has the advantage of fast convergence. To improve the optimization ability of DE, the crossover probability and crossover factor are changed as adaptive, and the local enhanced operator is used in the algorithm.

### A. Differential Evolution Algorithm

Differential evolutionary algorithm (DE), which inherits the idea of survival of the fittest, is a kind of evolution algorithm [6]. For each individual in the population, 3 points are randomly selected from the population. One point is taken as the basis, and the other 2 points are taken as the reference to make a disturbance. New points are generated after crossing, and the better one is retained by natural selection to achieve population evolution. Suppose the problem to be optimized is  $\min_{x \in R^n} f(x)$ ,

the main steps of the algorithm are as follows:

**Step 1:** Initialization. Set the population size  $N$ , the number of variables  $m$ , cross probability  $P_c$ , and cross factor  $P_m$ . When the evolutionary generation  $t = 0$ , initialize the low / up bound of the variables  $lb / ub$  and the initial population  $\bar{X}(0) = \{\bar{X}_1(0), \bar{X}_2(0), \dots, \bar{X}_N(0)\}$ , where the vector  $\bar{X}_i(0) = \{x_1^i(0), x_2^i(0), \dots, x_m^i(0)\}$ .

**Step 2:** Evaluation. Calculate the fitness value  $f(\bar{X}_i(t))$  for each individual  $\bar{X}_i(t)$ .

**Step 3: Mutation.** To the individual vector  $\bar{X}_i(t)$  in the population, three indices  $r_1, r_2, r_3 \in \{1, 2, \dots, N\}$  and a integer  $j_r \in \{1, 2, \dots, m\}$  are chosen by a random way.

$$x_j^{(i)}(t) = \begin{cases} x_j^{(r_1)}(t) + P_m \cdot (x_j^{(r_2)}(t) - x_j^{(r_3)}(t)), & \text{if } (rand < P_c \text{ or } j = j_r) \\ x_j^{(i)}(t) & , \text{otherwise} \end{cases}$$

**Step 4: Selection.**

$$\bar{X}_i(t+1) = \begin{cases} \bar{X}_i'(t), & \text{if } (f(\bar{X}_i'(t)) < f(\bar{X}_i(t))) \\ \bar{X}_i(t), & \text{otherwise} \end{cases}$$

**Step 5: Terminal condition.** If the individual vector  $\bar{X}_i(t+1)$  satisfies the termination condition, then  $\bar{X}_i(t+1)$  is the optimal solution, otherwise turn to Step 2.

### B. Adaptive Improvement

In DE, the crossover probability  $P_c$  and the crossover factor  $P_m$  are constant values. When the optimization problems are complex, the optimizing efficiency is not efficient enough. In the adaptive improvement,  $P_c$  and  $P_m$  are adapted according to the individual fitness values. When the population has a tendency to converge the local optimal solution, it increases both of the  $P_c$  and  $P_m$  values accordingly. When the population has a tendency to diverge, it reduces both of the  $P_c$  and  $P_m$  values. The  $P_c$  and  $P_m$  are adapted according to (1) and (2) [9]:

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{max} - f_{avg}} & f' \geq f_{avg} \\ P_{c1} & f' < f_{avg} \end{cases} \quad (1)$$

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f_{max} - f)}{f_{max} - f_{avg}} & f \geq f_{avg} \\ P_{m1} & f < f_{avg} \end{cases} \quad (2)$$

The parameters are described as follows [7]:  $P_{c1}$ : the higher crossover probability is 0.7~0.9;  $P_{c2}$ : the lower crossover probability is 0.4~0.6;  $P_{m1}$ : the higher crossover factor is 0.08~0.1;  $P_{m2}$ : the lower crossover factor is 0.01~0.05;  $f_{max}$ : the maximum fitness value in the population;  $f_{avg}$ : the average fitness value in the population;  $f'$ : the higher fitness value of  $\bar{X}_{r_2}(t)$  and  $\bar{X}_{r_3}(t)$ ;  $f$ : fitness value of  $\bar{X}_{r_1}(t)$ .

According to (1) and (2),  $P_c$  and  $P_m$  can be adaptively adjusted, which improves the optimization performance of the algorithm.

### C. Local Enhancement Improvement

Because DE generates a new intermediate individual through random deviation perturbation, the local optimal performance is not very well. While approximating the optimal solution, it still needs to iterates several generations to get the optimal value, which affects the convergence speed of the algorithm. Therefore, the local enhancement operator  $MP$  is introduced to DE. After obtaining a new population, some individuals in the new population (excluding the current optimal individual) are reassigned with probability  $MP$ . In this way, the individuals are distributed by the optimal individual of the current population. Local enhancement operator can enhance the greediness of the individuals, and makes the DE converges faster.

$$\bar{X}_{i,t+1} = \bar{X}_{best,t+1} + P_l \cdot (\bar{X}_{r_1,t+1} - \bar{X}_{r_2,t+1}) \quad (3)$$

Where,  $\bar{X}_{i,t+1}$  is the enhanced new individual,  $\bar{X}_{r_1,t+1}$  and  $\bar{X}_{r_2,t+1}$  are the original individuals,  $\bar{X}_{best,t+1}$  is the best individual of the current population,  $P_l$  is the perturbation factor. The indices  $r_1$  and  $r_2$  are mutually exclusive integers, which meet  $r_1 \neq r_2 \neq i$ .

It is the essence of local enhancement to DE, that is to make some individuals seek the solution by the optimal vector of the current population. While keeping the diversity of the population, the greed of the good individuals is increased to ensure that the algorithm finds the global optimal solution quickly. The local searching ability of the algorithm is improved by the perturbation factor  $P_l$ , which accelerates the convergence speed, especially when approximating the global optimal solution.

The adaptive local enhanced differential evolution algorithm (ADLEDE) is generated by the adaptive and local enhanced improvement based on DE. The flow chart of ADLEDE is shown in Fig. 1.

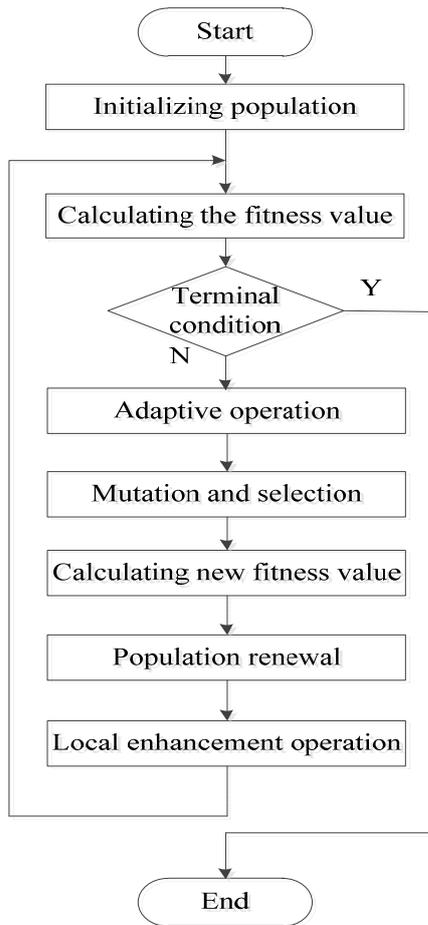


FIGURE I. LOW CHART OF ADLEDE

### III. MULTI-POPULATION DIFFERENTIAL EVOLUTION ALGORITHM

In this paper, we propose the MPDE is developed based on ADLEDE, which is abbreviated as MPADLEDE. The different populations optimize parallelly in MPADLEDE. It divides the randomly generated initial population into several sub-populations [4][5], and each sub-population (group) runs DE independently. In the process of evolution, the excellent individuals in the group will be selected to maintain the diversity of each group, and it improves the convergence ability of the algorithm. So, how to select the excellent individuals among the groups is a key problem in MPDE. Optimal substitution strategy (OSS) and elite immigration strategy (EIS) are studied in this paper.

#### A. Optimal Substitution Strategy

The OSS is a global selection and substitution strategy. To improve the original population, it selects the best individual among the groups to replace the worst individuals in the original groups. The main steps of OSS could be described as follows:

**Step 1:** Randomly generate  $N$  populations (groups), and  $n$  is the population size of each group. The parameters to each group are initialized as ADLEDE.

**Step 2:**  $N$  groups are co-evolved according to ADLEDE, and each group calculates the fitness value. They adaptively adjust the crossover probability  $P_c$  and crossover factor  $P_m$ , and complete the evaluation, mutation, selection, local enhancement and other operations.

**Step 3:** At the end of each generation, choose the best individual among the  $N$  groups, and replace the worst individual of each group.

**Step 4:** If the terminal condition is satisfied, end the program; If not, turn to step 2.

The OSS is described in Fig. 2.

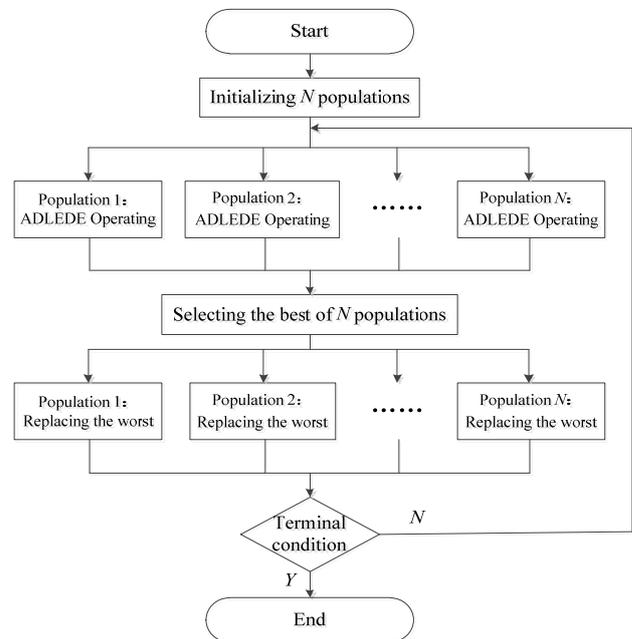


FIGURE II. FLOW CHART OF OSS

#### B. Elite Immigration Strategy

The EIS is a local selection and immigration strategy. To improve the convergence of original population, EIS selects the best individual of each population to replace the worst individual of the target group. The main differences of OSS and EIS is in step 3. For EIS, step 3 should be changed as: at the end of each generation, choose the best individual of each group, and replace the worst individual of the target group. The EIS is described in Fig. 3.

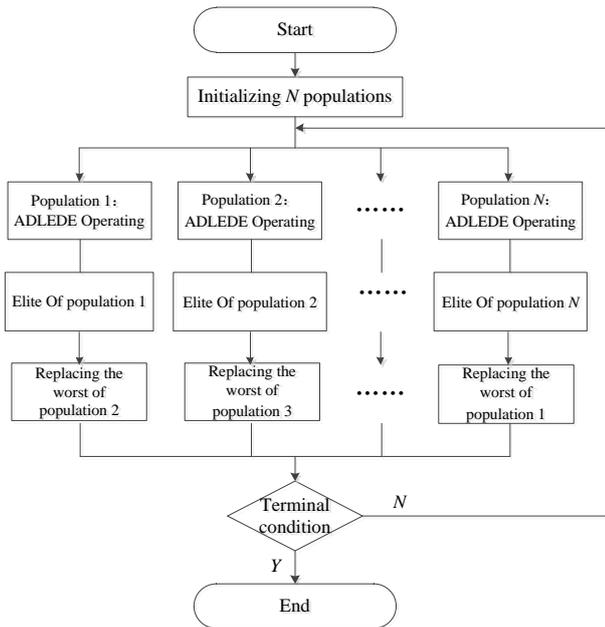


FIGURE III. FLOW CHAT OF EIS

#### IV. SIMULATION AND OPTIMIZATION PERFORMANCE ANALYSIS

In this simulation part, the OSS and the EIS are compared, and the performances of MPADLEDE and ADLEDE are studied.

##### A. Test Functions Selection and Parameters Setting

Usually, the Rosenbrock function, Sphere function, Griewank function and Rastrigin function are selected as the test functions according to [7]. Researches show that ADLEDE has good convergence and stability when the dimension is less than 30. In this part, the Sphere function is selected to do some researches on OSS and EIS. Both low (30-D) and ultra- high dimensions (300-D) conditions are considered in the simulation

to study MPADLEDE and ADLEDE. The parameters of the test function are set as Tab. 1, and the parameters of MEADLEDE and ADLEDE are set as Tab. 2. In the simulation, the operating system is Windows 10, the processor is Intel i7-8550U clocked at 1.8GHz, and RAM is 8GB. The maximum evolutionary algebra is set at 1500 for the algorithms.

TABLE I. THE TEST FUNCTIONS

| Test function | Expression           | Low dimensions | Ultra- high dimensions | Domain         | Optimum value |
|---------------|----------------------|----------------|------------------------|----------------|---------------|
| Sphere        | $\sum_{i=1}^n x_i^2$ | 30             | 300                    | $[-100,100]^n$ | 0             |

TABLE II. THE PARAMETERS OF THE ALGORITHMS

| Parameters | Refer to                         | Values             |
|------------|----------------------------------|--------------------|
| N          | number of multi-population       | 4                  |
| nofv       | number of variables (dimensions) | 30/300             |
| popsiz     | population size                  | 90/150             |
| lb/ub      | low / up bound of the variables  | domain             |
| mp         | local enhancement factor         | 0.01               |
| pc1, pc2   | crossover probability            | pc1=0.8, pc2=0.5   |
| pm1, pm2   | crossover factor                 | pm1=0.09, pm2=0.03 |
| precision  | accuracy to the optimum value    | 0.01               |

##### B. Analysis of the Convergence Performance

Each of the algorithm is simulated 20 times, and the results show in Tab. 3.

According to the simulation, the followings can be concluded:

- The performance of EIS is better than that of OSS. The simulation results show that all the indexes of EIS are better than the indicators of OSS both in low and ultra-high dimensions. At the average time, it is reduced by 11.8% (ultra-high dimensions) and 13.3% (low dimensions).

TABLE III. SIMULATION RESULTS

| Condition             | Algorithm | Convergence rate | Minimum generation | Maximum generation | Average generation | Average time(s) |
|-----------------------|-----------|------------------|--------------------|--------------------|--------------------|-----------------|
| ultra-high dimensions | OSS       | 100%             | 412                | 445                | 429.15             | 78.8969         |
|                       | EIS       | 100%             | 390                | 441                | 410.05             | 69.5927         |
|                       | ADLEDE    | 30%              | 1384               | 1486               | 1463.4             | 4.4839          |
| Low dimensions        | OSS       | 100%             | 98                 | 105                | 101.3              | 17.3674         |
|                       | EIS       | 100%             | 93                 | 102                | 98.4               | 15.0620         |
|                       | ADLEDE    | 100%             | 119                | 230                | 172.1              | 0.96567         |

- OSS converges faster in the early stage, and EIS has a better convergence performance in later stage. The convergence performances of EIS and OSS are shown in Fig. 4 in the case of low dimensions. OSS replaces the worst individual of each group with the best one in all sub-populations. It has the advantage of global optimization, so it can quickly approach the optimal

solution in the early stage. However, OSS also has the disadvantage of unicity at the same time, which reduces the diversity of the populations. Because of that, it converges slowly in the later stage. EIS selects the best individual of each population to replace the worst individual of the target group, and the diversity of the populations keeps well. Even if the performance

of EIS is worse than that of OSS in the early stage, EIS performs much better than OSS in the later stage. Finally, EIS is superior to OSS at the performance of convergence for the whole stage.

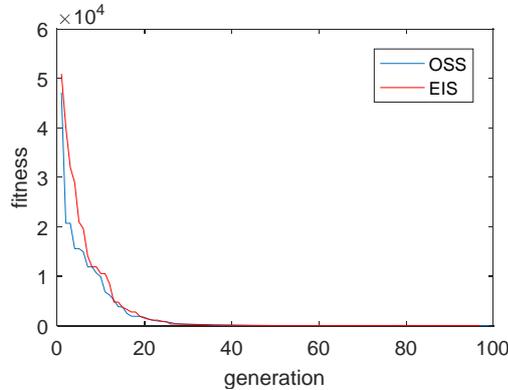


FIGURE IV. LOW-DIMENSIONAL PERFORMANCE OF OSS AND EIS

- MPADLEDE has more obvious convergence advantages than ADLEDE in the simulation of ultra-high dimensions. Whether MPADLEDE takes the OSS or the EIS, it is convergent 100%. However, ADLEDE converges 6 times in 20 simulations, accounting for only 30%. Therefore, the SPDE has premature defects in optimizing the ultra-high dimensional problems, and it shows instability characteristics in convergence performance.
- ADLEDE is more efficient than MPADLEDE if both of them converge. In the ultra-high dimensional simulation, the average convergence time of ADLEDE is only 4.4839s, while EIS takes 69.5927s. In the low dimensional simulation, ADLEDE takes 0.96567s, while EIS takes 15.0620s, which is 15 times more than ADLEDE. The main reason is that MPDE runs several DEs parallelly in various groups, it greatly increases the load of computers.
- MPADLEDE converges faster than ADLEDE at the same evolutionary generation. The convergent performances of ADLEDE and EIS are showed in Fig. 5 in the condition of low dimensions. It takes more average convergence time of MPADLEDE than that of ADLEDE. However, MPADLEDE adopts the strategy of “time-for-space”, it has a better convergence performance than ADLEDE at the same evolutionary generation. The indexes including minimum generation, maximum generation and average generation of MPADLEDE are superior to that of ADLEDE. Especially in the ultra-high dimensional simulation, ADLEDE does not converge all the time.

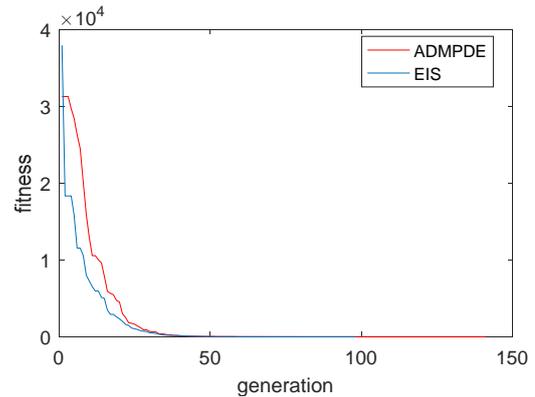


FIGURE V. LOW-DIMENSIONAL PERFORMANCE OF ADMPDE AND EIS

## V. CONCLUSION

DE shows good convergence and stability in optimizing simple problems, but it has premature risk when the problems are complex. In this paper, the MPDE is proposed, which overcomes the premature problems of SPDE. OSS and EIS are used in MPADLEDE to maintain the diversity of the populations. According to the simulation, it concludes that multi-population is a feasible and effective way to solve the premature problems. However, there are many other problems to be studied. Multi-population strategy can solve the premature problems of DE, but it takes much more time to converge. So how to reduce the converging time is a key problem to be studied next. OSS and EIS of MPADLEDE are studied in this paper. Other multi-population strategies can be used to improve the performance of the algorithm.

## ACKNOWLEDGMENT

We thank professor Weimin Li from Air Force Engineering University for his useful comments which have greatly improved the manuscript. We also like to thank anonymous reviewers who gave constructive suggestions that has helped to improve the quality of the paper. The work described in this paper is supported by a grant from the National Natural Science Foundation of China (No. 61402364), and the Xi'an International Studies University Research Foundation (No. 19XWB06). We also would like to thank them for providing necessary financial assistance.

## REFERENCES

- [1] Storn R. Price K., “Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces,” ICSI, USA, Tech. Rep. TR-95-012, 1995 [Online]. Available: <http://icsi.berkeley.edu/~storn/litera.html>.
- [2] Storn R., Price K. “Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces”. *Journal Global Optimization*, vol. 11, no. 4, 1997, pp. 361-349.
- [3] Storn R., “Differential evolution research: Trends and open questions,” in *Advances in Differential Evolution*, U. K. Chakraborty, Ed. Berlin, Germany: Springer, 2008, pp. 1–32.

- [4] Su H. J., Yang H. P., Wang Y. J., "Research on differential evolution algorithm: a survey," *Systems Engineering and Electronics*, vol. 30, no. 9, 2008, pp. 1793-1797.
- [5] Tasgetiren M. F., Suganthan P. N., "A multi-populated differential evolution algorithm for solving constrained optimization problem," in *Proc. IEEE Int. Conf. Evol. Comput.*, 2006, pp. 33-40.
- [6] Huang R. Q., Li W. M., Zhou X. G. et al. "Research on firepower distribution model of surface to air missile based on differential evolution algorithm," *Journal of Air Force Engineering University(Natural Science Edition)*, vol. 10, no. 5, 2009, pp. 41-44.
- [7] Huang R. Q., Jin C., He X. J. et al. "A modified differential evolution algorithm with adaptive and local enhanced operator," *Journal of Air Force Engineering University(Natural Science Edition)*, vol. 12, no. 3, 2011, pp. 84-89.
- [8] Huang R. Q., Li W. M., Lei Z. Y. et al. "Research on the deployment optimization to the multi-static infrared sensors in early warning of ballistic missile," *Transducer and Microsystem Technologies*, vol.29, no. 5, 2010, pp. 31-33.
- [9] Fan. H. Y., Lampinen J. "A trigonometric mutation operation to differential evolution," *Journal of Global Optimization*, vol. 27, 2003, pp. 105-129.
- [10] Wang C. H., "Research of adaptive differential evolution fuzzy clustering segmentation algorithm," master's thesis, Xi'an University of Technology, China, 2017, pp.25-41.
- [11] Xu Y.L. "Research on optimization problem of wireless sensor networks using differential evolution algorithms," doctoral dissertation, Donghua University, China, 2015, pp. 61-68.