An Effective Hybrid Evolutionary Algorithm for Solving the Numerical Optimization Problems

Xiaohong Qian¹,², Xumei Wang¹, Yonghong Su¹,² and Liu He¹
¹College of Information Engineering, Wuhan Huaxia University of Technology, Wuhan 430223, China
*Corresponding author: a21424691@qq.com; b461782640@qq.com

Abstract: There are many different algorithms for solving complex optimization problems. Each algorithm has been applied successfully in solving some optimization problems, but not efficiently in other problems. In this paper the Cauchy mutation and the multi-parent hybrid operator are combined to propose a hybrid evolutionary algorithm based on the communication (Mixed Evolutionary Algorithm based on Communication), hereinafter referred to as CMEA. The basic idea of the CMEA algorithm is that the initial population is divided into two subpopulations. Cauchy mutation operators and multiple paternal crossover operators are used to perform two subpopulations parallelly to evolve recursively until the downtime conditions are met. While subpopulation is reorganized, the individual is exchanged together with information. The algorithm flow is given and the performance of the algorithm is compared using a number of standard test functions. Simulation results have shown that this algorithm converges significantly faster than FEP (Fast Evolutionary Programming) algorithm, has good performance in global convergence and stability and is superior to other compared algorithms.

1. Introduction
Evolutionary Programming (EP), Evolution Strategies (ES) and Genetic Algorithm (GA) are three typical evolutionary algorithms, all of which are based on simulating the evolution of a biological population composed of individuals to solve complex optimization problems, especially those that are difficult to solve using other methods or not at all.

Although EP was initially considered an effective solution to artificial intelligence, in recent years, it has been successfully applied to solve numerical optimization problems [1]. In order to solve the problems that the traditional EP algorithm is prone to premature convergence, the Cauchy mutation is introduced to EP, and then FEP (Fast EP) is formed [2]. FEP is more effective than traditional EP in solving multi-peak and multi-local optimization problems. However there is not much improvement for the single peak and multi-peak less local extremum problems and the convergence speed is not fast.

Recently Guo Tao and so on [3] proposed a multi-parent crossover operator for solving function optimization problems. Practice has proved that Guo Tao algorithm has many advantages in solving optimization problems [4]. Its algorithm is simple and computational efficiency is high where you can find more than one optimal solution convergence speed is particularly fast and so on. After experiments, we find that Guo Tao algorithm can converge to the global optimal solution very quickly for the optimization of unimodal functions and multi-peaks with few local extremums.

However it is difficult to find the optimal solution for some multi-peak and multi-local
optimization problems. For example, Generalized Rastrigin's Function has the minimum of zero, but Guo Tao algorithm has an average result of only about 12.

In this paper the Cauchy mutation in FEP and the multi-parent hybrid operator in Guo Tao algorithm are combined to propose a hybrid evolutionary algorithm based on the exchange model (Mixed Evolutionary Algorithm based on Communication), hereinafter referred to as CMEA. In this algorithm, evolution takes place between two parallel populations which uses two different operators separately. Evolution is not only the process of natural selection by environmental individuals, but also the result of exchange among different populations. Subpopulation reorganization leads to the flow of individuals among subgroups and this flow is the manifestation of information exchange between subgroups. Finally 23 test function benchmark are used as an example to verify that the CMEA algorithm has good global convergence and robustness to all kinds of numerical optimization problems.

2. FEP algorithm and introduction of Guo Tao algorithm
This paper discusses the following form of numerical optimization problems: Known n-dimensional real-valued function f: S→R, S∈Rn is a closed set in space Rn, please find a point Xmin∈S which makes f(X) the smallest, that is ∀X∈S, f(Xmin) ≤ f(X).

2.1 Fast Evolution Planning (FEP)
FEP is described as follows:
Step 1: An initial population (Xi,μi) containing (i = 1, 2, ..., μ) is generated. The current evolution algebra k = 1.
Step 2: The function value of all the individuals in the current population is calculated.
Step 3: Each individual generates a descendant according to equation (1).
\[ x'_i(j) = x_i(j) + \eta_i(j) \delta_j \]  
\[ \eta_i(j) = \eta_i(j) \exp(\tau N(0, 1) + \tau N(j, 0, 1)) \]  
(1)
(2)
\[ \delta_j \] means that the average value of each component is 0 and Cauchy random number with standard deviation 1. Generally \( \tau = (\sqrt{2/n}) - 1 \), \( \tau = (\sqrt{2/n}) - 1 \).
Step 4: The function value of the subbody (Xi’ ’η”) (i = 1, 2, ..., μ) is calculated.
Step 5: The tournament is held in the set consisting of the parent (Xi, ηi) and the child (Xi’, η”) (i = 1, 2, ..., μ). For each of the individuals, q individuals are chosen from 2μ randomly to compare with it, and the number of the q individuals inferior to the evaluated individuals is taken as the score of the individual.
Step 6: μ individuals with the highest score selected from the set consisting of (Xi, ηi) and (Xi’, η”) (i = 1, 2, ..., μ) are formed to the next generation population.
Step 7: If the shutdown condition is reached, evolution is discontinued, otherwise evolving algebra k = k+1 and go to step 3.

2.2 Guo Tao algorithm (GT)
Guo Tao algorithm introduced the subspace concept. The subspace occupied by the M points Xj ’= (x1j’, x2j’, ..., xnj’), j = 1, 2, ..., M in S recorded as follows:
\[ V = \{X ∈ D | X = \sum_{j=1}^{M} \alpha_j X_j \} \]  
(3)
Where \( \alpha_j \) satisfies the following conditions: \( \sum_{j=1}^{M} \alpha_j = 1 \), 0.5 ≤ \( \alpha_j \) ≤ 1.5.
The algorithm is described as follows.
Step 1: N initial individuals are randomly generated. P = {X1, X2, ..., Xn}. Xi∈S, the current evolution algebra k = 1.
Step 2: The best individual Xbest in the population and the worst individual Xworst are found.
Step 3: If \( |f(X_{best}) - f(X_{worst})| ≤ \varepsilon \) then go to step 8.
Step 4: M points X1, X2, ···, XM are randomly selected from P form the subspaces V.
Step 5: A point X is chosen from V randomly.
Step 9: If f (X ') < f (Xworst), then Xworst = X'.
Step 7: k = k+ 1, go to step 2.
Step 8: Output Xbest.

3. CMEA algorithm flow
The basic idea of the CMEA algorithm is that the initial population is divided into two subpopulations. The first N individuals are classified as the first population, Cauchy mutation [6] is used to generate N descendants and the last μ - N individuals are classified as the second population using multiple paternal crossover operators to produce μ - N offspring. The random competition method is then used to rank all 2μ individuals, taking the first μ individuals as the next generation population and then continuing to evolve recursively until the downtime conditions are met. The entire algorithm flow is as follows:

Step 1: Parameter initialization. Setting the number of individuals in the population to μ, if the first population contains N individuals, then the second population contains μ - N individuals. The biggest evolutionary algebra is MaxGeneration, the number of randomly competing individuals is q with initial η0, parent number of multi-parent crosses (That is Zhang Cheng subspace size) is M and evolution algebra is k.
Step 2: Population Initialization. An initial population (Xi , ηi), (i = 1, 2; ···,μ) with k = 1 is generated in the feasible solution space of the problem.
Step 3: Evolution is stopped to judge. If k == MaxGeneration is satisfied, evolution aborted and the result is outputted, otherwise go to step 4.
Step 4: Breeding. In the first population N, each individual uses the formula (1) to generate a progeny. The second population uses equation (3) to generate μ - N offsprings. A total of μ offsprings were generated. In order to prevent loss of search ability when the step size is very small, the method in literature [5] is used: if ηi < 10 -4 , then ηi = 10 -4.
Step 5: Population communication. The integration of the parent (Xi, ηi) and the child (Xi', η′i) (i = 1, 2, ··· ,μ) is sorted using the random q competition method. The first μ individuals are the next generation. In this way, the first population and the second population communicate through the flow of individuals.
Step 6: k = k+ 1. Go to step 3.

4. CMEA optimization simulation examples and result comparison
In order to make the experiment representative, this paper chooses 23 different types of standard test functions. It is necessary to choose a large number of standard test functions so that we can analyze the performance of CMEA algorithm for different functions. For example, for some optimization problems it has good performance while for others the effect is not good.

4.1 Standard test function description
In the 23 test functions, f1-f13 is a high-dimensional optimization problem, f1-f5 is a unimodal function, f6, is a single peak discontinuous step length function; f7 is a function of noise; f8- f13 is a multi-peak function and the local extremum points grow geometrically as the dimension increases. These six functions are the most difficult optimization problem type. The function f14-f23 is a multimodal function with low-dimensional and local extrema.

4.2 Experimental Procedures and Results
If the number of individuals in the first population N = 80, the number of individuals is 20 in the second population. The value of parameter M for multi-parental crossover is slightly different for the optimization problem. Table 1 shows the specific parameter settings. For all the experiments in this paper, the remaining parameters are consistent with literature [2], that is q = 10, η0 = 3, μ = 100.
Table 1. The setting of parameter M

| Function | f1-f4 | f5 | f6 | f7 | f8 | f9 | f10 | f11 | f12-f13 | f14-f23 |
|----------|-------|----|----|----|----|----|-----|-----|---------|---------|
| M        | 4     | 6  | 4  | 3  | 5  | 7  | 4   | 5   | 4       | 5       |

In order to reflect the stability of the algorithm, we run the algorithm 50 times independently for each instance and then calculate the average and standard deviation to compare with the FEP algorithm and the SPMEP algorithm[7]. As can be seen from Table 2, the CMEA algorithm is obviously superior to the other two algorithms for the unimodal function f1-f6 and the multi-peak less local exponential function f15-f23. In this case the multi-parent hybrid operator shows a superior performance. With the introduction of multiple patrons in a crossover operation it reduces the possibility that super individuals copy themselves into children, which means that brings a more diverse solution space search results so that the evolutionary algorithm converges to the global optimum. Fig. 1 and Fig. 2 show the evolution curves of the optimal individual function values in the population with evolutionary algebra when f3 and f21 are solved (Fig. 1 shows the ordinate in logarithmic coordinates). As can be seen from the figure, the CMEA algorithm converges significantly faster than the FEP algorithm. The CMEA algorithm is superior both in the early stage of evolution and in the late stage of evolution and CMEA algorithm has a higher rate of evolution, which is the result of combined action between Cauchy mutation operator and multi-parent hybrid operator.

Table 2. CMEA algorithm compared with FEP and SPMEP. "Mean best" indicates the average value found by the algorithm, and "Std dev" represents the standard variance

| Function | Number of Generations | CMEA Mean best | CMEA Std dev | SPME[7] Mean best | SPME[7] Std dev | FEP[2] Mean best | FEP[2] Std dev | CME t-test | CMEA-F t-test |
|----------|-----------------------|----------------|--------------|------------------|-----------------|-----------------|---------------|------------|--------------|
| f1       | 1500                  | 9.5×10^-9     | 5.1×10^-1    | 1.3×10^-3       | 1.0×1           | 5.7×1           | 1.3×10        | -9.19      | -31.00       |
| f2       | 2000                  | 1.27×10^-1    | 2.9×10^-2    | 5.1×10^-3       | 1.5×1           | 8.1×1           | 7.7×10        | -23.0      | -74.26       |
| f3       | 5000                  | 8.0×10^-4     | 3.7×10^-1    | ---             | ---             | 1.6×1           | 1.4×10        | ---        | -7.67        |
| f4       | 5000                  | 1.2×10^-3     | 7.8×10^-1    | 6.5×10^-3       | 2.0×1           | 0.3             | 0.5           | -17.4      | -4.23        |
| f5       | 20000                 | 2.399          | 2.75         | ---             | ---             | 5.06            | 5.87          | ---        | -2.90        |
| f6       | 1500                  | 0.0            | 0.0          | 0.0             | 0.0             | 2.6×10          | 0.0           | -1.11      | 2.43         |
| f7       | 3000                  | 9.3×10^-3     | 4.2×10^-1    | 1.0×10^-2       | 1.5×1           | 7.6×1           | 2.6×10        | -1.11      | 2.43         |
| f8       | 9000                  | -12543.4       | 59.4         | -12569.5        | 9.1×1           | -1255           | 52.6          | 3.10       | 0.98         |
| f9       | 5000                  | 0.15           | 0.4          | 2.9×10^-7       | 6.0×1           | 4.6×1           | 1.2×10        | 2.65       | 1.83         |
| f10      | 1500                  | 1.0×10^-4     | 1.2×10^-2    | 1.9×10^-3       | 4.4×1           | 1.8×1           | 2.1×10        | -27.9      | -60.17       |
| f11      | 2000                  | 6.1×10^-3     | 7.9×10^-2    | 5.6×10^-3       | 1.7×1           | 1.6×1           | 2.2×10        | 0.43       | -2.99        |
| f12      | 1500                  | 6.68×10^-4   | 4.1×10^-1    | 8.5×10^-3      | 9.7×1           | 9.2×1           | 3.6×10        | -13.5      | -17.82       |
| f13      | 1500                  | 9.2×10^-6     | 2.4×10^-1    | 1.4×10^-5       | 9.2×1           | 1.6×1           | 7.3×10        | -1.32      | -13.87       |
| f14      | 100                   | 1.04           | 0.19         | 1.00            | 1.6×1           | 1.22            | 0.56          | 1.48       | -2.15        |
| f15      | 4000                  | 3.0756×10^-3 | 3.65×1       | 4.5×10^-4       | 1.5×1           | 5.0×1           | 3.2×10        | -6.71      | -4.25        |
| f16      | 100                   | -1.03          | 0            | -1.03           | 4.3×1           | -1.03           | 4.9×10        | 0.0        | 0.0          |
| f17      | 100                   | 0.398          | 0            | 0.398           | 5.7×1           | 0.398           | 1.5×10        | 0.0        | 0.0          |
| f18      | 100                   | 3.00           | 0            | 3.00            | 5.6×1           | 3.02            | 0.11          | 0.0        | -1.28        |
| f19      | 100                   | -3.86          | 0            | -3.86           | 1.5×1           | -3.86           | 1.4×10        | 0.0        | 0.0          |
| f20      | 200                   | -3.3           | 4.5×10^-1    | -3.25           | 5.4×1           | -3.27           | 5.9×10        | -5.03      | -2.86        |
| f21      | 100                   | -9.0           | 2.6          | -6.63           | 3.5             | -5.52           | 1.59          | -3.84      | -8.07        |
| f22      | 100                   | -9.97          | 1.4          | -7.4            | 3.00            | -5.52           | 2.12          | -5.49      | -12.38       |
Cauchy's mutation dominates in the multi-peak multi-local exponential function $f_8$-$f_{13}$, parent hybrid operator only plays fine-tuning role. For function $f_9$, CMEA algorithm has poor showing. Figure 3 is an image of a function in two dimensions. As can be seen from the image, this is a complex function with multiple peaks, multiple local maxima and close-spaced peaks. This function grows geometrically with the increase of the number of local extremum points. At this time the Cauchy variation in CMEA algorithm dominates. The domain of function $f_9$ is [-5.12,5.12]. As shown in Figure 4, the Cauchy mutation has more steps in the small multimodal neighborhood with large back and forth in the peak and peak disturbance but don’t have the ability to climb, so it is difficult to find the optimal solution.

| $f_{23}$ | 100 | -10.19 | 1.38 | -6.53 | 4.00 | -6.57 | 3.14 | -6.12 | -7.46 |

Figure 1. The image of a function $f_9$ in two dimensions

Figure 2. The optimal individual curve of function $f_{21}$

Figure 3. The optimal individual curve of function $f_3$

Figure 4. The optimal individual curve of CMEA for function $f_9$

5 Conclusion and the next step of research work

According to the respective advantages of fast evolution programming and Guo Tao algorithm, this paper proposes the hybrid evolutionary algorithm based on AC by combining Cauchy mutation operator and multi-parent hybrid operator. In this algorithm, evolution proceeds between two populations. Communication between the two populations is achieved by the reorganization of the populations. The algorithm flow is given and the performance of the algorithm is compared using a
number of standard test functions. The experimental results show that the CMEA algorithm has good performance in global convergence and stability.

The CMEA algorithm exhibits excellent performance for both unimodal functions and multi-peak less local extremum functions. However there is not much improvement on the multi-peak local extremal function over the FEP algorithm. The next steps of work are as follows: (1) Replacing the Cauchy mutation with Lévy mutation and T mutation to improve the shortcoming that Cauchy mutation step is too large. (2) Improving the parameter M of multi-parent hybrid operator in CMEA algorithm so that has its own adapt to the adjustment function for different types of functions and improving the robustness of the parameters. (3) The CMEA algorithm is used in optimization problems of various linear and nonlinear constraints to make the algorithm more general.

Acknowledgements
This work was partly supported by the Educational Science Planning Project of Hubei Province (No 2017GB088).

References
[1] Guo-sheng HAO, Na GUO, Gai-ge Wang, Zhao-jun ZHANG, De-xuan ZOU. Scheme of Big-Data Supported Interactive Evolutionary Computation, Science and Engineering Research Center, 2017, (1), 26-31.
[2] Yao X, Liu Y, Lin G. Evolutionary programming made faster [J]. IEEE Trans Evolutionary Computation, 1999 (3) : 82-102.
[3] Li Xia, Zhang Jingmin, Li Ruihua, Liu Kunqi, A cultural algorithm based on differential evolution algorithm and Guo Tao algorithm, Computer Engineering, 2012, (13): 178-181.
[4] Deng Changshou, Ren Hongwei, Peng Hu. An improved differential evolution algorithm for mixed integer nonlinear programming problems, Application Research of Computers, 2012, 2: 445-448.
[5] Yousef Alipouri, Javad Poshtan, Yagub Alipouri. A modification to classical evolutionary programming by shifting strategy parameters, Applied Intelligence, 38(2) : 175-192, 2013.
[6] Quan Ji, Wang Xianjia, Evolutionary games in a generalized Moran process with arbitrary selection strength and mutation, Chinese Physics B, pages 25-30, 2011.
[7] M. Ji, H. Tang, J. Guo. A single-point mutation evolutionary programming. Information Processing Letters, 90(2): 293-299, 2004.