

## An improved Differential Evolution optimization based on P system

Yanhua Zhong, Shuzhi Nie

Depart. of Electronics and Information Technology  
Jiangmen Polytechnic  
Jiangmen, China  
e-mail: zhflowers@163.com

**Abstract**—Membrane computing (P system) is a new computing model; it comes from the research of the basic function and structure of creature's cell membrane. It has complex structure and multi-level. It converges quickly and has high quality of the optimal results. In this article, discussed the basic theory of the membrane computing and the steps of the algorithm. Apply Membrane computing principle to the differential evolution algorithm, constructed a hybrid differential evolution algorithm on the basis of the membrane structure. Finally, utilized the Objective Functions to test the new algorithm performance, compared with related algorithms to analyze the advantages and disadvantages of the new algorithm.

**Keywords**- *Differential Evolution; Cell Membrane Structure; P System; Function Optimization*

### I. INTRODUCTION

The differential evolution algorithm (DE) [1, 2] is a new evolutionary computation technique. By Storn et al in 1995, was originally envisaged to solve the Chebyshev polynomial, but later found that the DE effective technology solve the complex optimization problems. DE and artificial life, in particular, the evolutionary algorithm has a very special connection. DE and particle swarm optimization (PSO) [3], are optimization algorithm based on swarm intelligence theory. Optimize the search for intelligent guidance through cooperation and competition between individuals within populations groups, but compared to other evolutionary algorithms, DE reserves the population-based global search strategy, using real-coded, based on a simple variation of the differential operation and one-on-one competitive strategy for survival, reducing the genetic complexity of the operation. At the same time, DE unique capacity of memory it can dynamically track the current search, to adjust their search strategy, with strong global convergence ability and robustness, and without the aid of the characteristics of the problem, and it is suitable to solve some of the complex optimization problem. Currently, DE has in many areas of application, such as artificial neural networks, chemical, electrical, mechanical design, robotics, signal processing, bioinformatics, economics, modern agriculture, food safety, environmental protection and operational research, etc. However DE get a wide range of research, compared with other evolutionary algorithms, the results of their research is quite fragmented, and lack of systematic, in particular, no major breakthrough in the theory.

Membrane Computing [4] is a theory, which studies formal models of distributed computations with basic notions inspired and highly abstracted from biology. In a nutshell, membrane computing considers computational models having such basic components as a membrane (separator of an internal region from external one), membrane structure (nested one or a network), multiset of abstract objects (presenting the content of a region) and evolution/communication rules over multiset of objects. In other words, models from membrane computing (called also P systems) are computational devices, intuitively presented by a totality of membranes organized in hierarchical way (a membrane can enclose other membranes) or as networks (membranes are connected to each other by means of communication channels and can form in this way any topology). Each membrane has a collection of some objects, which can be presented in multiple copies, i.e., as multisets. Objects can be considered to have an atomic structure, hence, presented as letters, or to have complex structure, for instance, to be presented as strings. Each membrane can have a collection of evolution/communication rules, which can be of one of the two following types: multiset-rewriting rules or symport/antiport rules. Multiset-rewriting rules are an abstraction of chemical reaction rules consuming certain amount of certain chemicals and producing other chemicals. Symport/antiport communication rules are an abstraction of symport/antiport transporting rules of chemicals through cellular membranes. Evolution/communication rules being applied usually in a maximally parallel nondeterministic way define behavior of P systems, called evolution or computation. Rules describe how content of membranes changes. A number of universality results is known for various types of P systems, i.e., a P system can be viewed as an abstract universal (i.e., programmable) device. There were reported several applications of P systems in biology, linguistics, computer science, management, etc. Nowadays research in membrane computing goes into three main directions: exploring computational power and universality of various classes of P systems; attacking NP-complete problems and using P systems for modelling purposes; attempting to implement P systems in hardware and bioware.

In this paper, the main method is to combine membrane computing ideas with standard Differential Evolution (DE). The idea to form a membrane community solution space, the Individual in the community will be quick convergence solution space effects. The best Individual, as Representative, complete the optimal Individual of the higher level. This can

effectively prevent the fall into local minimum, premature convergence or stagnation.

## II. DIFFERENTIAL EVOLUTION MEMBRANE OPTIMIZATION ALGORITHMS

### A. Mode of membrane computing

Studied DNA computing for many years, inspired by biological cells, Gheorghe Păun proposed the membrane computing in 2000 [3], through dealing with compounds from layered structure of living cells to abstract the computing model. The model is called membrane systems or P system, A P system of n dimensional can be expressed as the following multiple tuples:

$$\Pi = (V, T, C, \mu, w_1, \dots, w_m, (R_1, \rho_1), \dots, (R_m, \rho_m)) \quad (1)$$

In formulary (1),  $V$  is alphabet, its elements are called objects;  $T \subseteq V$  is output alphabet;  $C \subseteq V - T$  is catalyst, its elements don't change in the evolutionary process, don't create new characters too, however, perform some evolution rules required its;  $\mu$  is a membrane structure that contain m membranes, each membrane and its enclosed area show with a label set of  $H$ ,  $H = \{1, 2, \dots, m\}$ , then m is  $\Pi$ 's

dimensional;  $w_i \in V^*$  ( $1 \leq i \leq m$ ) show multiple sets that contain objects in the region i of membrane structure m,  $V^*$  is the collection of arbitrary character string composed of  $V$ 's character; Evolution rules is a binary set of  $(u, v)$ , Usually written as  $u \rightarrow v$ ,  $u$  is the string of  $V^*$ ,  $v = v'$  or  $v = v' \delta$ ,  $v'$  is the character string of collection  $\{a_{here}, a_{out}, a_{in_j} | a \in V, 1 \leq j \leq m\}$ ,  $\delta$  is the special characters don't belong to  $V$ , When a rule contains  $\delta$ , membrane is dissolved while the implementation of the rules, regard the length of  $u$  as the rules of  $u \rightarrow v$  radius,  $R_i$  ( $1 \leq i \leq m$ ) is the finite set of evolution rules, each  $R_i$  is associated with the region of  $i$  in the membrane structure of

$\mu$ ,  $\rho_i$  is the second order in  $R_i$ , which is the preferential relations, Which is the preferential relation that implementation the rules of n.

In short, P system consists of three parts: hierarchical structure of membrane; the multiple set of objects; evolution rules.

### B. The differential evolution algorithm (DE)

The differential evolution algorithm (DE) is a group-based evolutionary algorithm, with the memory of the individual optimal solution and the features of information sharing within populations, to achieve optimization problems through cooperation and competition between individuals within populations, its essence is the greed genetic algorithm based on the real number coding.

Assuming the solution  $\min_{x \in R^n} f(x)$ , the DE algorithm description is:

(1) Initialization, input evolutionary parameters: population size  $N$ , crossover probability  $P_c$ , cross-factor  $F \in (0, 1)$ , evolution algebra  $t = 0$ , randomly generated initial population  $\bar{X}(0) = \{X_1(0), \dots, X_N(0)\}$ , including  $X_i(0) = (x_1^i(0), \dots, x_n^i(0))$ .

(2) Individual evaluation, calculate for each individual target values  $f(X_i(t))$  for  $X_i(t)$ .

(3) Reproduction, to the population of each individual  $X_i(t)$  randomly generated three distinct random integers  $r_1, r_2, r_3 \in \{1, 2, \dots, N\}$  and random integer  $j_{rand} \in \{1, 2, \dots, n\}$ ,

$$X_j^{(i)}(t) = \begin{cases} x_j^{(r_1)}(t) + F(x_j^{(r_2)}(t) - x_j^{(r_3)}(t)), & \text{if } rand[0, 1] < P_c \text{ or } j = j_{rand} \\ x_j^{(i)}(t), & \text{else} \end{cases} \quad (2)$$

(4) Select the following expression:

$$X_i(t+1) = \begin{cases} x_j^{(i)}(t), & \text{if } f(x_j^{(i)}(t)) < f(x_j^{(i)}(t)), \\ x_j^{(i)}(t) & \text{else,} \end{cases} \quad (3)$$

(5) Termination of test, if populations of  $X_i(t+1)$  satisfy the termination criterion, output Individuals  $X_i(t+1)$  with the smallest target as the optimal solution. Otherwise, go to (2)

The search performance of DE algorithm depends on the balance of the algorithm for global exploration and local development capacity, which is largely dependent on the algorithm control parameters, including population size, scaling factor and crossover probability. Compared to other evolutionary algorithms, DE required adjusting the parameters less, reasonable parameter selection guidance see [1, 5].

### C. Differential evolution based on P system

The Differential evolution based on P system process is as follows:

Step 1: the establishment of the structure contains  $m$  membranes, each membrane and the surrounding areas use label set  $H = \{1, 2, \dots, j, \dots, m\}$ .

Step 2: The Individuals were randomly assigned to the  $m$ -membranes.

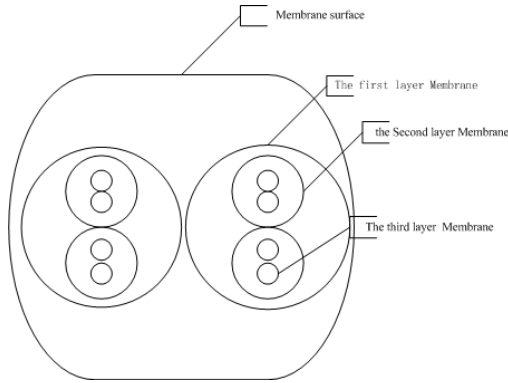


Figure 1. Four-layer structure of membrane

Step3: initialize the membrane objects, and object set size, were taken randomly the Decimal vector, and set operating parameters, such as maximum operating generation, the DE parameters F and CR. And to ensure that each membrane at least one individual

Step4: randomly generated basic membrane DE evolution generation.

Step 5: In each membrane, respectively, using the DE optimization in which the DE procedure refer to 2.2.

Step6: transit operations, exchange of information between the membrane and the membrane. The population evolution of each membrane is completed; the local optimal solution is left on the observation of population, the film dissolved, the local optimal solution into the next layer of membrane.

Step7: All population (the Local optimal solution) joined to the next-layer membrane, updated population through differential evolution to complete the evolutionary search, until all membrane of the surface membrane dissolved and disappeared, otherwise, Go back to step (4) to continue, finally, the remaining population in the surface membrane was the optimal solution, and ended the algorithm.

### III. ALGORITHM SIMULATIONS AND TESTING

To test the optimum performance of hybrid Differential Evolution (C), used typical test functions of as follows:

(1) Griewank function:

$$f(\bar{x}) = \frac{1}{4000} \sum_{i=1}^{50} x_i^2 - \prod_{i=1}^{50} \left( \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) + 1; x_i \in [-600, 600] \quad (4)$$

In formula (4), the optimal value of Griewank function is  $f(\bar{0}) = 0$ , which have numerous broad local minimum, which is the most difficult function to converge to the global minimum, the local picture of two-dimensional Griewank function is shown in Figure 2. We can know it has many local minima; they are differing little, so the algorithm is easy to fall into local minima.

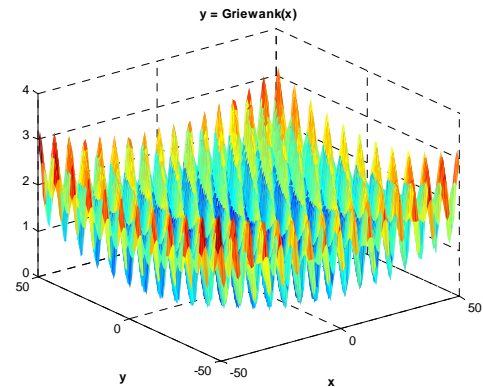


Figure 2. Function Graphs of Griewank function

(2) Rosenbrock function

$$f_2(\bar{x}) = \sum_{i=1}^l (100x_{i+1} - x_i^2)^2 + (1 - x_i)^2; x_i \in [-5.12, 5.12] \quad (5)$$

This is the famous Banana function. The global optimal value of Banana function is  $f(\bar{1}) = 0$ , hidden in a narrow parabolic shaped flat valley being. Because of its independent variables associated with each other, this valley is very difficult to find. This is a very hard to find excellent test functions.

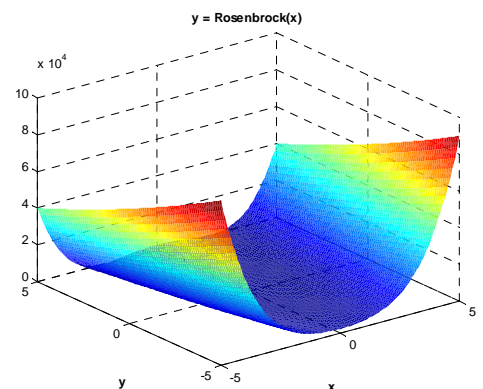


Figure 3. Function Graphs of and Rosenbrock function

Compared the properties of DEMC with other optimization algorithms, respectively adopted genetic algorithm (GA), membrane computing (MC), Stom etc. proposed differential evolution algorithm (DE) [1], J.Liu and J.Lampinen proposed fuzzy adaptive differential evolution algorithm (FADE) based-on DE [8]. FADE has a strong global search capability, can effectively avoid the premature convergence. Algorithms mentioned above respectively evaluated fifty-dimensional variable Griewank function and Rosenbrock function with the same number, explained the operation generation of the test function, the testing results are shown in Table 1. Some data come from the literature [8].

TABLE I. THE TESTING RESULTS OF DIFFERENT ALGORITHMS

algorithm	evolution generations(G)	operation results of Griewank function	operation results of Rosenbrock function
GA	5000	153.98	497.79
DE	5000	312.78	2378
FADE	5000	0.5741	41.59
MC	50	269.7	1145.68
MC	5000	2.68e-7	43.94
DEMC	50	184.6	45.632
DEMC	5000	8.43e-4	3.53

Test results from Table 1 can be seen the superiority of DEMC. Especially for the Rosenbrock function tests, other algorithms have not found the optimal solution. Experimental tests of DEMC closest optimal solution, the higher accuracy of solution value can be obtained.

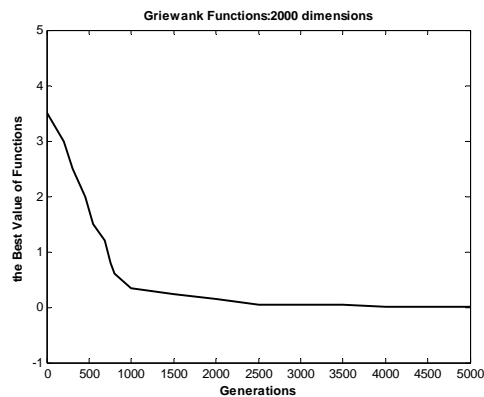


Figure 4. The convergence curve of DEMC testing Griewank function

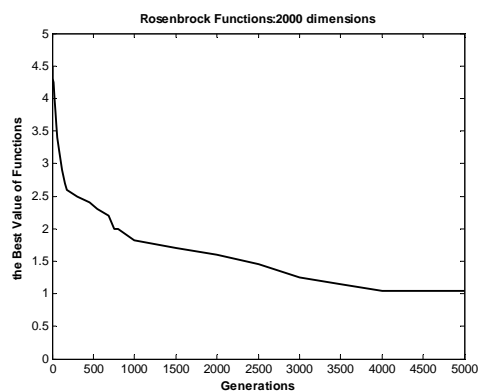


Figure 5. The convergence curve of DEMC testing Rosenbrock function

About fifty-dimensional variable Griewank test function, After 5,000 generations of evolution, FADE convergence value can converge to 0.5741, which is far superior to the convergence value of DE, it is 312.78, but more less than the

optimization value of DEMC, DEMC only needed average 36.3 generation can get the function value is less than 0.5741. Obviously, for the optimization method of DEMC, which convergence speed faster than FADE. When the number of variables increased, complexity of the problem would increase exponentially. Used DEMC to solve 2000 independent variables of Griewank function and Rosenbrock function are shown in Figure 4,5, after 1000 generations, the result of Griewank function is reduced to 0.405, and the result of Rosenbrock function is reduced to 3.53, which is superior to FADE solving Griewank function and Rosenbrock function of 5000-dimensional variables to obtain the optimal function value.

#### IV. CONCLUSIONS

The simulation results above showed that the DEMC is an efficient optimal method, the results can be infinitely close to the optimal solution. Membrane computing was still in the stage of theoretical research, in this paper, the performance of proposed DEMC and its other applications need further study. Finally, The DEMC algorithm has the following advantages:

- 1) The algorithm is generic and does not depend on the problem of information;
- 2) The method is simple, easy to implement;
- 3) Groups, search, with the capacity of the memory of individual optimal solution;
- 4) Cooperative search, with the ability of individual local information and groups global information guidance algorithm to search further;
- 5) Easily mixed with other algorithms, construction algorithms with better performance.

#### REFERENCES

- [1] Storn R, Price K. Differential evolution —A simple and efficient adaptive scheme for global optimization over continuous spaces. Berkeley: University of California, 2006.
- [2] Lampinen J. A bibliography of differential evolution algorithm [EB/OL]. (2002210214). <http://www.lut.fi/~jlampine/debiblio.htm>.
- [3] B. Liu, L. Wang, Y.H. Jin, et al. Advances in particle swarm optimization algorithm. Control and Instrument s in Chemical Industry, 2005, 32 (3): 126.
- [4] Gh. Paun. Computing with membranes. Journal of Computer and System Sciences, 61, 1(2000), 108-143.
- [5] X.F. Xie, W.J. Zhang, G.R. Zhang, et al. Empirical study of differential evolution. Control and Decision, 2004, 19 (1): 49252, 56.
- [6] B. Liu, L. Wang, Y.H. Jin. Advances in differential evolution. Control and Decision.2008, 22(7), 721-729.
- [7] S.J. Yuan, X.S. Gu. A Differential Evolution Based Membrane Computing Algorithm. Shanghai Institute of Chemistry and Chemical Engineering 2010 Annual Conference Proceedings (automated thematic), 2010.
- [8] J. Liu, J. lampinen. A fuzzy adaptive differential evolution algorithm. Soft Computing.2005, 9(6):448-462.
- [9] L. Huang, N. Wang. An Optimization Algorithm Inspired by Membrane Computing. Lecture Notes in Science.2006:49-52.