

A New P System Based Genetic Algorithm

Caiping Hou*, Xiyu Liu

School of Management Science and Engineering, Shandong Normal University, Jinan, Shandong, China
Corresponding author, e-mail: sdnuhpc@163.com*, sdxyliu@163.com

Abstract

For the “early convergence” or the “genetic drift” of the genetic algorithm, this paper proposes a new genetic algorithm based on P system. Based on the parallel mechanism of P system in membrane computing, we put forward the new P system based genetic algorithm (PBGA). So that we can improve the performance of GA.

Keyword: genetic algorithm, P system, parallel computing

Copyright © 2016 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

In the past few decades, how to optimize the genetic algorithm (GA) have become a hot issue. In the field of mathematical optimization, a genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic (also sometimes called metaheuristic) is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions [1]. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

Thus, a P system is a membrane structure with objects in its membranes, with specified evolution rules for objects, and with given input/output prescriptions. Any object, alone or together with one more object, evolves, can be transformed into other objects, can pass through one membrane, and can dissolve the membrane into which it is placed. All objects evolve at the same time, in parallel; in turn, all membranes are active in parallel. The evolution rules are hierarchized by a priority relation, given in the form of a partial order relation, the rule with the highest priority among the applicable rules is always the one actually applied. If the objects evolve alone, then the system is said to be noncooperative; [Error! Reference source not found.] if there are rules which specify the evolution of several objects at the same time, then the system is cooperative; an intermediate case is that where there are certain objects (we call them catalysts), specified in advance, which do not evolve alone, but appear together with other objects in evolution rules and they are not modified by the use of the rules.

2. The New GA Algorithm Based on P System (PBGA)

2.1. Fitness Function

Genetic algorithm exists such a function that plays decisive role in the evolution direction of genetic algorithm. This function is the fitness function. Fitness may be the objective function and it may also be a function of the objective function related. Moreover, it can also have nothing to do with the objective function. Genetic algorithm evaluates the candidate by the size of the value of the fitness function.

Genetic algorithms can determine the chance of candidate solutions into the next generation of genetic based on the value of the fitness function. It is about whether the quality of candidate solutions may have a better chance of genetic evolution. Also, it is about whether the excellent characteristics of the population can be continued [3]. In this paper, a fitness function

based on exponential transformation is adopted to expand differences in exponential form. The fitness function is described as follows:

$$F(x) = \exp(-C) - \exp[-(f(x) + C)]$$

Where $f(x)$ is the minimized objective function optimization problems. C is a positive number large enough to ensure $(f(x)+C)$ nonnegative. After encoding n individuals of the initial population into quaternary sequence, we sort the fitness values by the fitness function value. Choose the individuals with the smallest (optimal) and the largest (worst) fitness value (2 individuals in total) as elite individuals and put the remaining $(N-2)$ individuals into the P system.

2.2. P Processing

After the operation, the two elite individuals remain in the areas between the basic outer membrane and the surface membrane. The remaining $(N-2)$ individuals are put into m basic membranes equally, which forms a P system. And each individual in each membrane can be viewed as a chromosome. In this section, the membranes will be arranged as a loop topology. The individuals have some evolution rules to evolve each other in the system. In order to make the running time of each membrane relative to the average of the running time, we define the number of the membranes as follows:

$$Z = \lfloor \sqrt{N-2} \rfloor$$

Where N is the total number of individuals of initial population, Z is the square root of $(N-2)$ (round down). When N is large enough, we can get the square root of N directly. Using this way to define the number Z can maintain similar elite individuals between intramembrane and extramembrane, which can reduce the overall convergence time. When adding the membrane structure we define the rules: individuals in Z membranes conduct quantum genetic operations. After several times of iteration, each membrane will generate an elite individual. When the termination criteria are met, the membrane structure will dissolve. Z elite individuals will be released from Z membranes. They will step to next round operation together with the initial 2 elite individuals. Because it involves several elite individuals, P system can effectively solve the problem of getting into the local optimum.

2.3. Genetic Operators

2.3.1. Selection Operator

Selection operator determines the direction of evolution. In this paper, we adopt elitism strategy and choose the fitness function as selection operator. Outside the membrane, we adopt the elitism strategy as well. The elite selection guarantees that the best chromosome in a certain generation would not be lost in the evolutionary process. The basic idea of elite selection is as follows: if a certain individual in the former generation is better than the best individual in the current generation, some individuals in the current generation will be replaced by the better ones in the former generation.

2.3.2. Crossover Operators and Mutation Operator

We will start with a set of rules R :

$$\mathcal{R} = \left\{ \begin{array}{lll} r_1 \equiv [a \rightarrow ab]_e & r_7 \equiv [z_2 \rightarrow z_1]_e & r_{13} \equiv [a \rightarrow \lambda]_s \\ r_2 \equiv [b \rightarrow bc]_e & r_8 \equiv [z_3 \rightarrow z_4]_e & r_{14} \equiv [b \rightarrow \lambda]_s \\ r_3 \equiv [c \rightarrow b^2]_e & r_9 \equiv [z_1]_e \rightarrow b & r_{15} \equiv [b \rightarrow c]_e \\ r_4 \equiv [a \rightarrow bc]_e & r_{10} \equiv [z_2]_e \rightarrow a & r_{16} \equiv [c \rightarrow \lambda]_e \\ r_5 \equiv [z_1 \rightarrow z_2]_e & r_{11} \equiv [z_3]_e \rightarrow c & r_{17} \equiv [z_4 \rightarrow z_1]_e \\ r_6 \equiv [z_2 \rightarrow z_3]_e & r_{12} \equiv [z_4]_e \rightarrow a & r_{18} \equiv [z_4]_e \rightarrow b \end{array} \right\}$$

Our aim is to use genetic algorithms in order to find a P system which computes the square of number 4, from an initial set of P systems [Error! Reference source not found.]. The genetic evolution will only correspond to changes in the set of rules. The genetic operations in order to develop a genetic algorithm on the P systems are the following:

Crossover: Given two P systems π_1 and π_2 and their sets of rules R1 and R2, let $P_1^1 P_1^2$ and $P_2^1 P_2^2$ two partitions of R₁ and R₂ respectively. Then, we obtain two offsprings π_1' and π_2' by considering the set of rules $R_1' = P_1^1 \cup P_2^1$ and $R_2' = P_1^2 \cup P_2^2$.

Mutation: Given an evolution rule $[u \rightarrow v]_h$ with $u \in \Gamma$ and $v \in \Gamma^*$, the mutation operator changes the object u by one from $\Gamma - \{u\}$ or the object w in the multiset v by one object from $\Gamma - \{w\}$ or by λ . For a dissolution rule $[u]_h \rightarrow w$, the mutation operator changes the object u or w by a different one from Γ .

Only for practical reasons, in this case study we will impose an extra condition. All the P systems considered as individuals in our genetic algorithm must be deterministic. This is checked by ensuring that, for each P system and each membrane, there are no two rules triggered by the same object [Error! Reference source not found.].

2.4. Procedure of the PBGA

The procedure of PBGA can be summarized as follows:

1. Initialize a population with N individuals.
2. Calculate the fitness value of each individual and sort.
3. Choose the two individuals with the best and worst value as elite individual. The remaining (N-2) individuals are put into the P system. Set maximum evolution generation inside the membrane as T1.
4. Conduct genetic operations in each membrane.
5. Repeat step (4) until the termination criteria inside the membrane is met. m elites are released outside and step to next round operation together with the initial 2 elite individuals. And maximum evolution generation outside the membranes is set as T2.
6. Conduct genetic operations on elite individuals.
7. Repeat step (6) until the termination criteria outside the membranes is met and finally optimal solution is generated. The PBGA algorithm ends.

3. Experimental Simulation

In order to investigate the performance of the PDNA-GA algorithm, we choose three real-life data sets provided in UCI [Error! Reference source not found.](including the Iris, Breast Cancer, Wine) and compared the running time with the classical GA algorithm.

In the experiments, the parameters of PBGA are set as follows. The population size is 300 (N = 300). The parameters of the mutation operator is set as $g_0 = G_{max}/2$, $a_1 = 0.02$, $b_1 = 0.2$, $aa = 20/G_{max}$, $pm3 = 0.05$. The number of membranes is 17 ($m = 17$). The maximum evolution general inside T1 and outside T2 is 500. For each test problem, we run 50 times for the two algorithms of PBGA and GA. The results are shown in Table 1.

Table 1. The results obtained by the algorithms for 50 runs on the three data sets

Data sets	PBGA	GA
Iris	97.67 ±0.2466	99.83 ±5.5239
Breast Cancer	3000.16 ±20.246	3249.26 ±229.734
Wine	16296.07 ±0.9416	16298.42 ±2.1523

Table 1 gives the comparison results of the two algorithms on the three data sets, respectively. From the results, we can see that the PBGA provides the optimum average value and smallest standard deviation in compare to GA algorithms. For instance, the optimum value is 97.67 which is obtained in most of runs of PBGA algorithm, however, the GA fails to attain the value even once within 50 runs. And the results on the Wine also show that the PBGA algorithm provides the optimum value of 16296.07 while the GA obtains 16298.42. Experimental results show that PBGA is an effective improvement to GA algorithm.

4. Conclusion

In this paper, a new GA algorithm based on P system (PBGA) is proposed by combining GA with P system for the first time. After repeated verifications, the PBGA algorithm performs pretty well when the population size is large enough. The future work is to improve the algorithm efficiency when the population size is small.

Acknowledgements

Project supported by National Natural Science Foundation of China (61170038, 61472231, 61502283), Jinan City independent innovation plan project in Colleges and Universities, China (201401202), Ministry of education of Humanities and social science research project, China (12YJA630152), Social Science Fund Project of Shandong Province, China (11CGLJ22)

References

- [1] Whitley D. A genetic algorithm tutorial. *Statistics and computing*. 1994; 4(2): 65-85.
- [2] Păun G. Computing with membranes. *Journal of Computer and System Sciences*. 2000; 61(1): 108-143.
- [3] Wang K, Wang N. A protein inspired RNA genetic algorithm for parameter estimation in hydrocracking of heavy oil. *Chem Eng J*. 2011. 167(1): 228-239.
- [4] Escuela G, Gutiérrez Naranjo MÁ. *An application of genetic algorithms to membrane computing*. Proceedings of the Eighth Brainstorming Week on Membrane Computing. Sevilla, ETS de Ingeniería Informática. 2010: 101-108.
- [5] Zhao S, Liu X. Research on a New DNA-GA Algorithm Based on P System. *Frontier and Future Development of Information Technology in Medicine and Education*. Springer, Netherlands. 2014: 1691-1698.
- [6] <http://archive.ics.uci.edu/ml/>