

Genetic Algorithm Theory and Its Application

Shaojun Yi^{1, *a}, Shengliang Zou^{2, b}

¹Nanchang University of Gongqing College, Jiujiang City, China;

²Nanchang University of Gongqing College, Jiujiang City, China.

*Corresponding author Email: ^a22756050@qq.com, ^b731944618@qq.com

Keywords: Genetic algorithm; Theory; Application

Abstract. Genetic Algorithm (GA) is a very effective optimization solution method. It can provide faster and better results than some traditional optimization methods for the control structures and control functions in the control field. It can be said that the intelligent connection of GA and CAD software packages for high-precision system modeling and controller automation design and parameter optimization will be a new and effective method. The current evolution of GA learning in human beings is still formal. It has not yet been able to portray the evolutionary process of human beings. It has failed to portray the true learning process of neuron thinking.

1. Introduction

At present, genetic algorithms have begun to be applied in various fields. At this stage, there is still no perfect theoretical analysis and algorithm analysis of genetic algorithms. It is a problem that needs to be solved in the future development process [1]. The related problems such as small-scale algorithm, genetic algorithm coding, selection of genetic algorithm control parameters, non-directional genetic operators, poor local search ability and premature convergence, etc. Based on this, we should further study the above issues and promote the application and development of genetic algorithms [2].

2. The principle and steps of genetic algorithm

The genetic algorithm is a probabilistic search algorithm with a global and self-adaptive ability to simulate the evolutionary and genetic processes that naturally inoculate the fittest and survive the fittest. In the process of natural evolution, organisms adapt to the external environment through inheritance and mutation, and survive and evolve from one generation to the next. The genetic algorithm simulates the above evolutionary phenomena. It maps the search space to a genetic space, that is, encodes every possible solution in the solution space into a vector called a chromosome. Each element of the vector is called a gene. All chromosomes form a group and each chromosome is evaluated according to a predetermined objective function, and a fitness value is given based on the result. At the beginning of the algorithm, some chromosomes are randomly generated, their fitness is calculated, and genetic manipulations such as selective copying, crossover, and mutation are performed on each chromosome according to the fitness degree [3], chromosomes with low fitness are removed, chromosomes with high fitness are left, and new ones are obtained. Groups. Since the members of the new group are the outstanding ones of the previous generation, they inherited the excellent performance of the previous generation and thus were clearly superior to the previous generation. The genetic algorithm repeats literately in this way, evolving towards a more optimal solution until it meets some predetermined optimization criterion [4]. The main steps of the genetic algorithm are as follows:

- (1) The solution of the problem is expressed as a code string (biological term is called a chromosome), and each code string represents a feasible solution to the problem;
- (2) Generate an initial population consisting of chromosomes of a certain length randomly;
- (3) Decode the chromosomes into optimization parameters, calculate the corresponding objective function and transform them into fitness values;

- (4) According to the level of chromosome individual fitness, perform application replication, crossover and mutation operations to generate the next generation of groups;
- (5) Return to step until the stop criteria is met.

In this way, step (3) through step (5) are repeatedly performed, so that the dyeing population is continuously evolved from generation to generation, and finally the individual most suitable for the problem environment is searched for, and an optimal solution to the problem is obtained.

3. The theoretical study of genetic algorithms

3.1 Code representation

Coding is the primary problem to be solved when applying genetic algorithms, and it is also a key step in the design of genetic algorithms. Holland's binary coding is the most commonly used coding method in genetic algorithms [5]. For the insufficiency of binary coding, many improvements have been proposed. The following are typical: (1) Gray code coding. It is a binary coded variant. Gray code not only has some advantages of binary coding, but also can improve the local search ability of genetic algorithms. (2) Real number coding. The method is suitable for representing a larger number of genetic algorithms, so that the genetic algorithm is closer to the problem space, and the encoding and decoding process is avoided. (3) Decimal encoding. The method uses decimal-coded control parameters to mitigate the premature convergence problem of "combined explosion" and genetic algorithms. (4) Non-numerical coding. The gene values in a chromosome encoding string take a set of symbols that have only code meaning and no numerical meaning. These symbols can be either numbers or characters. The advantage of non-numeric coding is that the expertise and related algorithms of the required problem can be used in genetic algorithms.

3.2 Adaptive functions

The concept of fitness is used in genetic algorithms to measure how well each individual in the population can achieve or contribute to finding the optimal solution in the optimization calculation. Individuals with higher fitness are more likely to inherit to the next generation, while those with lower fitness are less likely to inherit to the next generation. The function that measures individual fitness is called the fitness function. In the research process of genetic algorithms, He considered the function value and its rate of change of the function at the search point, and added this information to the fitness function so that the chromosome selected by probability not only has a smaller function value, but also Has a large change in function. The experimental results show that the convergence rate of this method is obviously higher than the standard genetic algorithm.

3.3 Genetic manipulation

The operation of the genetic algorithm is to simulate the operation of the biological gene. Its task is to impose a certain operation on the individual's fitness to achieve the evolution of the survival of the fittest. From the perspective of optimizing search, genetic operations can gradually optimize the solution of the problem and approach the optimal solution. Genetic operations include the following three basic genetic operators: selection, crossover, and mutation. The selection and intersection basically completed most of the search functions of the genetic algorithm, and the mutation increased the ability of the genetic algorithm to find a near-optimal solution.

(1) Select operation. The selection operation is a process of selecting a vitality individual in a group to create a new group. The genetic algorithm uses a selection operator to perform operations on individuals in a group. Potts et al. outlined 23 options. The convergence model of the selection operator was introduced by Goldberg for the first time; then it was expanded by Goldberg and Deb, and the concept of substitution time was proposed, and the selection pressure between various selection strategies could be compared.

(2) Crossover operation. Fork refers to the exchange of some genes between two chromosomes that cross each other in a certain way, thus forming two new individuals. It is the main method for generating new individuals, determines the global search ability of genetic algorithms, and plays a key role in genetic algorithms. Potts et al. summarized 17 crossover methods. The relationship between the mating operator and its exploration subspace is studied, the guiding principles for designing good operators are proposed, and a heuristic mating operator is constructed.

(3) mutation operation. Variation refers to the replacement of gene values at certain loci in an individual's chromosomal coding string with other alleles at that locus to form a new individual. It is an auxiliary method for generating new individuals and determines the local search ability of genetic algorithms. The mutation operator and the crossover operator cooperate with each other to complete the global search and local search of the search space, so that the genetic algorithm can complete the optimization process of the optimization problem with a good search performance, but the traditional mutation method cannot be effectively maintained. Allelic diversity at the same locus. Therefore, in view of the fact that the general mutation operation is likely to cause the program to fall into the local optimum and reach the global optimal solution, Wu Qiuling et al. proposed a new mutation operator that can maintain population diversity and has an adaptive mutation probability. The probability of mutation at each gene's position on the chromosome of an operator is adapt determined by the gene's frequency at its locus. Shi Jiying et al proposed the genetic algorithm of catastrophic uniform distribution, improved the mutation operator, kept the genes with high weight in the individual, and only made the genes with low weights mutated, achieving the goal of accelerating the convergence. Liu Huayong proposed a fast convergence genetic algorithm with adaptive search capability by introducing an adaptive Gaussian mutation operator, and obtained a finite element fairing model based on point-based curves and surfaces.

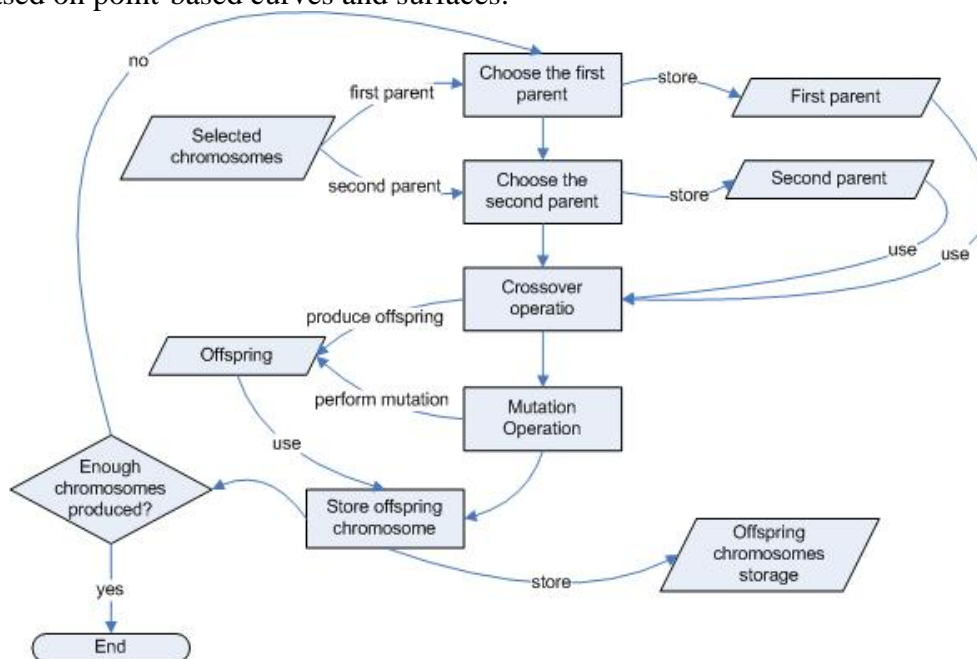


Fig. 1 Genetic Genetic Algorithm Demonstration Program

4. The improvement of genetic algorithm

4.1 Mufti-population genetic algorithm

Multi-population genetic algorithm is to divide the large population reasonably into many small populations and independently optimize all the small populations. Then the populations are counted. According to the above methods, certain effects can be obtained. Based on the mutation operator multi-population genetic algorithm, which is mainly the structure of subgroups and populations, it can prevent the problems existing in the traditional methods. After appropriate addition of multiple groups, different parameters can be used to control different populations to achieve different goals. Populations can be contacted based on immigration operators, and artificial methods can be used to ensure that each generation has the optimal population of individuals to further improve the genetic algorithm.

4.2 Optimization of control parameters

The genetic algorithm mainly includes the following control parameters: crossover probability , population number , and mutation probability P_m . The combination of different parameters will affect the actual performance of the genetic algorithm in different degrees. The first proposed

performance genetic algorithm performance parameters are to test five functions, and then choose a reasonable range of parameters, $P_m = 0.001$, $P_c = 0.6$, $N_{pop} = 50$, and has been widely used, since then also proposed to use orthogonal experiments to control and optimize parameters. The method is mainly based on the above method to balance the derisiveness, reduce the number of experiments, reach the form of most search parameter space combinations, clarify the influence of parameters on the actual results, and propose a reasonable and accurate search method to ensure effective solution to the problem.

4.3 Combined with intelligent algorithms

The genetic algorithm has a relatively strong global search capability and can quickly determine the optimal global point. However, there is a weak local search capability. When accurate solutions take a long time, organic intelligence algorithms and strong local search capabilities complement each other. Combined with particle swarm optimization and genetic algorithms, chaotic sequences are used to form nonlinear selections, initial populations, adaptive changes in variation scales, and multi-access offspring competitions to achieve genetic changes. According to the improved genetic algorithm, grain group optimization and the retention of elite individuals, a common new population is obtained, so as to avoid the problems of local convergence, premature convergence, and slow convergence rate in conventional algorithms. Combined with simulated annealing algorithm and genetic algorithm, combined with simplex method and genetic algorithm, set neural network and genetic algorithm, not only can improve performance, but also can expand the application field.

4.4 Combined with other intelligent algorithms

The global search ability of the genetic algorithm is strong, and the global optimum can be determined quickly, but the local search ability is weak, and further accurate solution takes a long time. Therefore, the combination of algorithms with strong local search ability and genetic algorithms can complement each other. Hangman et al. proposed the combination of genetic algorithm and abut search. Wei Ming et al. combined genetic algorithm with chaos optimization. In the course of genetic evolution, according to the relative diversity of populations, chaotic fields were introduced into each individual to search for effective genes. The effective combination of genetic algorithm is good at global optimization and strong chaotic local search ability, which significantly improves the computational efficiency and has great practical value. Ren Ziwu et al. combined genetic algorithm with particle swarm optimization, and used chaotic sequences to generate initial populations, non-linear sorting selections, multiple cross generations, competitive selection and adaptive mutations, and other improved genetic operations. Group optimization and improved genetic algorithm (IGA) combine to produce new individuals in order to overcome the shortcomings of slow convergence, premature convergence and local convergence in conventional algorithms. These hybrid strategies not only improve the performance of the algorithm, but also extend the application area of the algorithm.

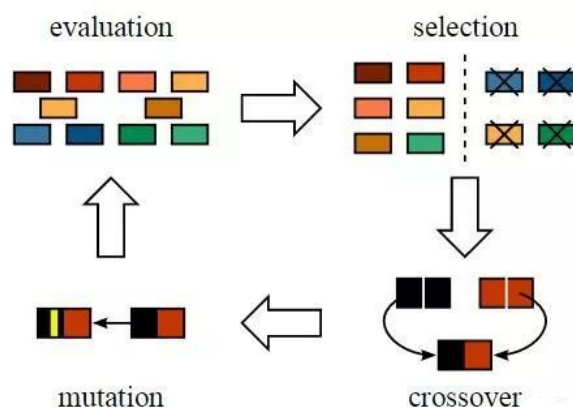


Fig. 2 Genetic algorithm demonstration analysis

5. Genetic algorithm in the field of control research and application

5.1 Automatic control area

In the field of automatic control, the genetic algorithm has played an enormous role, it can not only solve the problem of system parameter identification, improve the machine's automatic recognition ability, but also optimize the controller parameters. Kara and other programming adaptive fuzzy logic controllers apply genetic algorithm computations; Es-posit is based on Carr's research, applies genetic algorithms when researching RBF networks, optimizes RBF neural networks, and applies genetic algorithms such as Vein The structure and weights of the RBF neural network are improved, and the complete optimization of the RBF is basically achieved. In another research, foreign scientist Ionesco applied genetic algorithm to design the mangle train controller and successfully improved the function of the controller; Chinese scientist Yan Wenjun and others also proposed another design optimization robust control based on the genetic algorithm. Device.

5.2 In terms recombination optimization problems

The combination (best) optimization problem is a class of optimization problems. It studies continuous variables and discrete variables. Its goal is to find the optimal solution from the feasible solutions of the combination problem. combination optimization seems to be simple, but in reality it is not so. Solving combination optimization problems not only requires a lot of run time, but also requires a huge storage space. Existing computers cannot yet meet the criteria for combination optimization, the so-called "combination explosion". Genetic algorithms are widely used in combination optimization.

5.3 In terms of mufti-objective function optimization problems

The mufti-objective optimization problem was first proposed by the economist V. Pareto when studying the economic equilibrium, and the Pareto optimal solution was introduced and promoted. The mufti-objective optimization problem is abbreviated as "MOP" and its objective function has 2 or more. The effective solution of mufti-objective programming problem is called "Pareto optimal solution". The previous mufti-objective optimization algorithm takes a long time and has a large amount of calculation. It can no longer meet the social and economic development. With the development of computer science and life information science, the intelligent optimization algorithm can meet the actual needs when dealing with mufti-objective optimization problems. Genetic algorithm has great advantages in solving mufti-objective function optimization problems. The main points are as follows: 1 The genetic algorithm can process multiple targets at the same time, reducing the steps of the target sorting; 2 The genetic algorithm can search in many ways, not traditional The limitation of the algorithm is a major breakthrough of the mathematical programming method. 3 The genetic algorithm can solve the random and uncertain discrete search space problem. 4 The genetic algorithm can transform the mufti-objective optimization problem into a single target. Genetic algorithms can make up for the shortcomings of traditional mathematical programming methods. Therefore, it is a research hot spot to apply genetic algorithms to solve mufti-objective function optimization problems.

5.4 System identification and model price reduction

System identification is the basis of control system design. There are many effective methods. However, most of these technologies deal with the linear model of parameters and are based on the assumption that the search space is continuous and different. Current online identification methods are recursive implementations of off-line methods. These recursive methods are essentially local search methods using gradient techniques. When the search space is not divisible or the parameters are non-linear, these methods are not easy to find the global optimal solution. The search of the genetic algorithm does not depend on the gradient information and does not require The solution function is subtle, requiring only the solution of a feasible solution under self-constraint conditions, and the genetic algorithm has the characteristics of global search. GA provides a simple and effective method for the identification of nonlinear systems.

5.5 Social and Economic Applications

In the early course of economics research, genetic algorithms were mainly used to analyze and solve mathematical formula, and obtained corresponding results. However, compared with machine learning systems, there are still great differences and gaps. For example, Let tau was established in 1997 according to the above methods. A simple subject model; Bauer fully understands and analyzes the application of genetic algorithms in the fields of investment and economy. With the progress and development of society, genetic algorithms have been applied in the financial and commercial fields. At this stage, a large number of genetic algorithms have been applied at the time of stock investment and financial system analysis.

6. Conclusion

In recent years, genetic algorithms have developed very rapidly, both in theoretical research and in practical applications. With the passage of time, there has been a major breakthrough in the theoretical research of genetic algorithms. However, the theoretical system of genetic algorithms is still not perfect, which will hinder the development of genetic algorithms. Therefore, the theoretical research on genetic algorithms needs to be more in-depth and detailed. Scientists have successfully used genetic algorithms to solve problems in life. It is believed that this will be a key research area of genetic algorithms in the near future.

7. References

- [1]. Wu Mei, Lu Jingui, et al. Review of research progress in genetic algorithms. Machine Tools and Hydraulics. Vol. 42 (2008) No. 11, p. 176-179.
- [2]. Shi Mingxia, et al. Multi-population collaborative evolutionary genetic algorithm. Journal of Shangqiu Teachers College. Vol. 23 (2006) No. 5, p. 72-74.
- [3]. Ma Yuming, He Ailing, Li Aimin, et al. A review of theoretical research on genetic algorithms . Journal of Shandong Institute of Light Industry. Vol. 17 (2004) No. 6, p. 77-80.
- [4]. Xuan Guangnan, Cheng Runchuan, et al. Genetic algorithm and engineering design. Beijing: Science Press. Vol. 27 (2000) No.3, p.89-92.
- [5]. He Xingui, Liang Jiuyu, et al. Genetic algorithm using gradient of objective function. Journal of Software. Vol. 33 (2001) No. 12, p. 985-989.