Improved Genetic Algorithm based on Binary Coding

Chao Yang[a], Qin Wu

Department of Computer, Tianjin University of Technology and Education, Tianjin 300222, China.

[a]yangchao_id@163.com

Abstract

Aiming at the problem of slow convergence and low accuracy of genetic algorithm, an improved genetic algorithm based on binary coding is proposed. The algorithm abandons the traditional genetic algorithm roulette strategy; the cross strategy uses multiple individuals to make the best genes inherited; the mutation strategy uses the double gene mutation method; the elite feedback strategy is added to improve the algorithm's convergence speed and solution accuracy; According to the algorithm in different periods, use adaptive crossover, mutation probability, dynamic adjustment algorithm. The algorithm simulates multiple test functions to verify the effectiveness of the algorithm, which greatly improves the algorithm's convergence speed and solution accuracy.

Keywords

Genetic algorithm; Function optimization; Binary coding.

1. Introduction

All manuscripts must be in English, also the table and figure texts, otherwise we cannot publish your paper. Please keep a second copy of your manuscript in your office. When receiving the paper, we assume that the corresponding authors grant us the copyright to use the paper for the book or journal in question. Should authors use tables or figures from other Publications, they must ask the corresponding publishers to grant them the right to publish this material in their paper. Use *italic* for emphasizing a word or phrase. Do not use boldface typing or capital letters except for section headings (cf. remarks on section headings, below).

Genetic algorithm simulates the principle of "natural selection, survival of the fittest" in the process of natural evolution. Its main features are the search method between groups and the exchange of individual information in the group[1]. The genetic algorithm has the characteristics of strong robustness, independent of the mathematical characteristics of the problem, and strong spatial search ability of the solution. But genetic algorithms also have many problems, such as the most common "premature" phenomenon, poor local search capabilities, and slow convergence.

Since the genetic algorithm (GA) was proposed by Professor J. Holland in 1975, genetic algorithm has been researched and applied. Researchers study how genetic algorithms can be improved. The application fields of genetic algorithms are wide, such as function optimization, combination optimization, production scheduling problems, image processing, automatic control, machine detection and diagnosis optimization, flight scheduling problems, etc. He[2]et al. used genetic algorithms to optimize the design of the flight path in the terminal area to solve the problems of flight conflicts and evasion of restricted areas in the flight path. Li[3] et al. proposed a solution algorithm based on Monte Carlo similarity genetic algorithm, which realized the solution function of balanced transportation problem. Zhang[4] et al. proposed a tracking registration method based on image matching, which combined differential evolution strategy, taboo search strategy and genetic algorithm to meet the real-time requirements of tracking registration. Yao[5] et al. used genetic algorithms to study the parameter optimization of the NES, reduce the amplitude of the vibration system, and obtain a better multi-mode vibration suppression effect. Wu[6] et al. puts forward a method of predicting and analyzing BP neural network based on genetic algorithm optimization, which can more accurately predict the error of the falling point of the outer trajectory at a constant wind speed. Zhu[7] et al. combines genetic algorithm and simulated annealing algorithm to reduce the time of goods in and out of the warehouse, the distance of goods in the same group and the center of gravity of the shelves. Li[8] et al. proposes a real-time shared scheduling model for aviation materials based on improved adaptive genetic algorithm for aeronautical materials scheduling planning for different maintenance tasks of civil aircraft. Sun[9] et al. improves and optimizes the basic genetic algorithm and uses simulated annealing algorithm for population selection to make it highly efficient in AGV path planning. Wang[10] et al. proposed a new construction mechanism of mutation operator to improve the global optimization ability of genetic algorithm. Wen[11] et al. combines Apriori algorithm and genetic algorithm to avoid multiple scans of the database while reducing redundancy and improving the number and efficiency of mining frequent patterns. He[12] et al. made improvements to the standard genetic algorithm to solve the job shop scheduling problem with the optimization goal of minimizing the maximum completion time.

This paper proposes an improved genetic algorithm based on binary coding. The algorithm in this paper is an improvement made in the standard genetic algorithm, discarding the original roulette strategy; the cross strategy selects three individuals to maximize the inheritance of excellent genes; the mutation strategy uses the double gene mutation method; adding elite feedback strategy, Improve the convergence speed of the algorithm; use adaptive crossover, mutation probability, and dynamic adjustment algorithm. The algorithm simulates multiple test functions to verify the effectiveness of the algorithm, which greatly improves the algorithm's convergence speed and solution accuracy.

2. Genetic algorithm improvement

2.1 Encoding and Decoding

This article uses binary encoding. In the algorithm, the binary code makes up the individual's chromosomes. The length of the chromosome determines the accuracy of the solution. The longer the chromosome, the more accurate the solution. The method of binary encoding is relatively simple, but there are many ways of decoding. The decoding method is different, and the obtained solutions are also different. Let an individual binary code be (b0, b1, ... , bn). Decoding formula:

$$x = \left(\sum_{i=0}^{len} b_i \times 2^i\right)_{10} \frac{MAX-MIN}{2^{len}-1} + MIN \tag{1}$$

Where, x is a real number in the domain. len is the chromosome length. MAX is the maximum value of the domain. MIN is the minimum value of the domain.

2.2 Cross operation

Genetic algorithm simulates the evolution of biology in nature, excellent genes are inherited, so that the population evolves. In order to allow more excellent genes to be retained, this article selects three individuals as parents. The random number $r1 \in (0,1)$. When r1 is less than the preset threshold Pc, it will do cross-operation with other individuals as a parent; otherwise, it will not participate. Let P1 be the parent individual, P2 and P3 the random parent individual, locus the randomly selected gene, $r2 \in (0,1)$. When $r2 > 0.5$, randomly select the genes at the genetic loci of the parent individuals P1, P2, and P3 to inherit; otherwise, select the genes that appear at high frequencies in the three individual loci to inherit the high-frequency genes to the next generation.

2.3 Mutation operation

Genetic variation is to make changes in the genes of an organism. The genotype changes so that the phenotype may change. Genetic mutation is an important factor in biological evolution, and the mutation operation of genetic algorithms is derived from it. The mutation operation is also an effective method for the algorithm to jump out of the local extremum. This article takes two chromosomes as an example and adopts the method of double gene mutation to perform genetic manipulation. Random number $r2 \in (0,1)$, when r2 is less than the preset threshold Pm, then randomly select the gene locus of one chromosome to mutate, and the other chromosome to mutate in situ; otherwise, do not mutate.

2.4 Elite feedback operation

The purpose of the elite feedback strategy is to accelerate the convergence of the algorithm. This strategy uses known information to control the direction of population evolution, make individuals approach the current optimal individual, improve individual adaptability, and thus accelerate the pace of population evolution, while also improving the accuracy of the solution. In the traditional algorithm, the population evolves and selects the current best individual through each generation of the algorithm. The current best individual information is often ignored, but each generation of the current best individual contains more or less valuable information. The elite feedback strategy Then use this information to act on the population and make the individual react.

2.5 Adaptive operation

Because the algorithm needs different search capabilities in the early and late stages of the search. In the early days of algorithmic search, there were many individuals in the population. At this time, more individuals were required to participate in the crossover strategy to generate new individuals, that is, a higher crossover probability Pc was needed. To the late stage of the algorithm, due to the lack of individual diversity in the population. The genotypes of individuals are mostly similar, so a higher mutation probability is required to increase population diversity.

The formula is:

Crossover probability:

$$\mathrm{Pc}' = \begin{cases} Pc - \frac{25}{MAXGEN}\left(\frac{1}{2} - r_1\right)\left(1 - (2r_1)^{\frac{1}{21}}\right) & r_1 \le 0.5 \\ Pc - \frac{25}{MAXGEN}\left(r_1 - \frac{1}{2}\right)\left(1 - (2 - 2r_1)^{\frac{1}{21}}\right) & r_1 > 0.5 \end{cases} \tag{2}$$

Mutation probability:

$$\mathrm{Pm}' = \begin{cases} Pm + \frac{20}{MAXGEN} r_2\left(1 - r_2^{\frac{1}{21}}\right) & r_2 \le 0.5 \\ Pm + \frac{20}{MAXGEN}(1 - r_2)\left((1 - r_2)^{\frac{1}{21}} - 1\right) & r_2 > 0.5 \end{cases} \tag{3}$$

Where, Pc' and Pm' are contemporary crossover probability and mutation probability respectively. Pc' and Pm are the previous generation crossover probability and mutation probability respectively. MAXGEN is the maximum number of iterations of the algorithm. $r_1 \in (0,1)$,$r_2 \in (0,1)$.

It is not difficult to find from the above formula that as the algebra increases, the crossover probability gradually decreases and the mutation probability gradually increases. In this paper, the minimum crossover probability is reduced to 0.4, and the maximum variation probability is increased to 0.3. Through the above formula, the algorithm's differentiated needs for search capabilities in the early and late search are well satisfied, making the direction of the algorithm's evolution process more clear and the control of the algorithm more precise.

2.6 Improved algorithm flow

Step 1: Initialize the population and related parameters;

Step 2: Evaluate the population and find the current best individual Best;

Step 3: Implement crossover, mutation, and elite feedback strategies;

Step 4: Evaluate the population and find the best contemporary individual CBest. If the best contemporary individual CBest is better than the current best individual Best, then CBest→Best; otherwise Best→Best;

Step 5: Implement adaptive strategy.

Step 6: Determine whether the termination condition is met, if not, then go to step 3; otherwise, go to step 7;

Step 7: Output the current best individual Best, and the algorithm terminates.

3. Numerieal ExamPle

Comparing the improved genetic algorithm (I_GA) with the traditional genetic algorithm (GA), the parameters of both algorithms are the same. The population size is 100. The maximum number of iterations is 100. The crossover probability is 0.8. The mutation probability is 0.1.

Table 1 shows the test function and the definition domain. The optimal values of f1-f5 are all obtained at (0,0), and the optimal value is 0. The optimal value of f6 is obtained at (420.9687,420.9687), and the optimal value is 0. Table 2 counts that each algorithm is executed 30 times. This experiment was conducted in the following environment, operating system: Windows 7 Ultimate, processor: Intel(R)Core(TM) i5-3337U CPU @1.80GHz 1.80GHz, installation memory: 8.00GB, software: Microsoft Visual Studio 2017 , Language: C.

Table 1 Test function

Test function	Search space
$f_1=\sum_{i=1}^{n} x_i^2$	$[-100,100]^n$
$f_2=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]^n$
$f_3=0.5+\frac{sin^2(x_1^2-x_2^2)-0.5}{[1+0.001(x_1^2+x_2^2)]^2}$	$[-100,100]^2$
$f_4=10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]$	$[-5.12,5.12]^n$
$f_5=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+\exp(1)$	$[-32.768,32.768]^n$
$f_6=418.9829n-\sum_{i=1}^{n}x_i\sin(\sqrt{\lvert x_i\rvert})$	$[-500,500]^n$

Figure 1-2 are function images of test functions f1 and f2, respectively. It can be seen from the figure that f1 and f2 are simple single-peak functions, and there is only one minimum value globally. The search space length of f1 is 200. The search space length of f2 is 20. In terms of the search space, two single-peak functions are in contrast. The single-peak function is a real-valued function with only one strictly local maximum (peak) in the interval considered, but the single-peak function is not sufficient to represent the global search capability of the function, so the multi-peak functions f3, f4, f5 and f6. The multimodal function is a function that contains multiple local optimal solutions or global optimal solutions. The multimodal function can evaluate the advantages and disadvantages of the algorithm more comprehensively, such as global search ability, local search ability, convergence speed, and jump out of local optimal Ability etc.

Figures 3-6 are function images of the test functions f3, f4, f5, and f6, respectively. It can be observed from the figure that f3, f4, f5 and f6 are multi-peak functions, and there are many widely distributed local extremums in the search space. Figure 3 shows that f3 has a highly oscillating function image around the global optimal value. Figures 4-6 have many extreme points in the global. For the multimodal function, if the traditional algorithm is used, it is easy to fall into the local extreme value, especially the risk of the mountain climbing algorithm falling into many local extreme values, and the difficulty of solving is greatly increased.

In Table 2, the comparison between the algorithm and the genetic algorithm is obvious. Under the conditions of populations of the same size and population evolution algebra, the algorithm in this paper is far superior to genetic algorithms. Both f1 and f2 are single-peak functions. Through horizontal comparison, the accuracy of the solution obtained by the algorithm in this paper is more accurate when the two algorithms are in the same search space and the time used by the algorithm is almost the same. In longitudinal comparison, the search space of f1 is 10 times that of f2, and the accuracy of the solutions obtained by both algorithms in f2 is higher than that of f1. Variance is used to measure the degree of deviation between a random variable and its mathematical expectation. In the algorithm in this paper, the accuracy of the variance obtained by f1 is 10^{-64}, and the variance of f2 is 0, indicating that the 30-time algorithm can obtain similar results each time, and the algorithm is relatively stable.

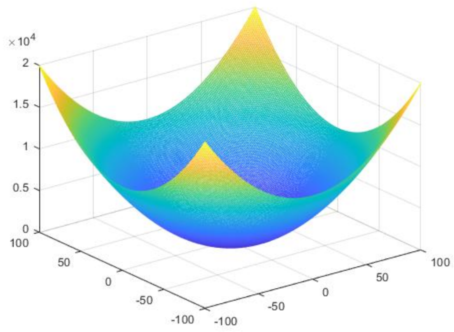

Fig. 1 f1 function

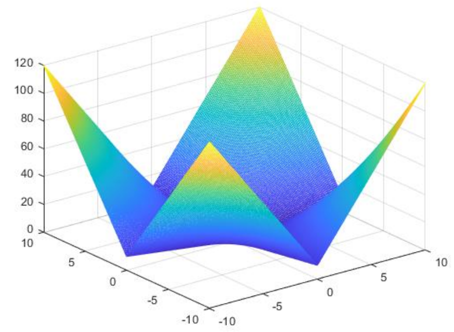

Fig. 2 f2 function

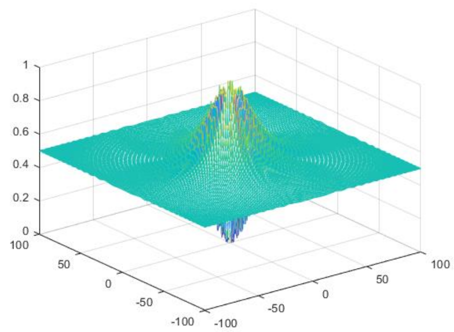

Fig. 3 f3 function

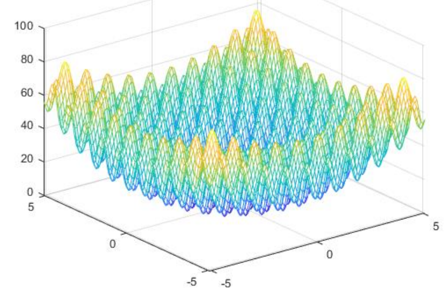

Fig. 4 f4 function

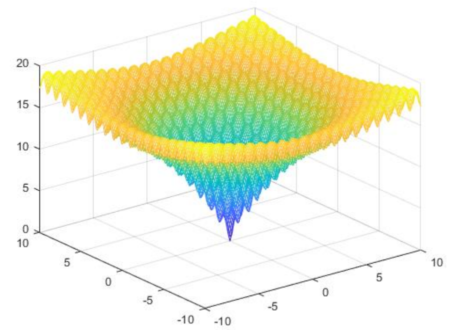

Fig. 5 f5 function

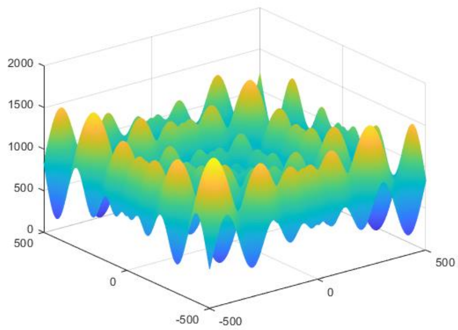

Fig. 6 f6 function

f3, f4, f5 and f6 are all multi-peak functions, which can more comprehensively judge the performance of the algorithm. The f3 function is highly oscillating around the global optimal value (0,0). The data shows that both algorithms can find the global optimal. But the accuracy of the solution of the algorithm in this paper is 10^{-11}, while the accuracy of the solution of the comparison algorithm is only 10^{-4}, which shows that the algorithm of this paper is better. The three functions f4, f5 and f6 have many extreme points globally. As can be seen from the data in Table 2, the best results of the two algorithms are near the global optimum. The variance of this algorithm is 0 when calculating f4 and f5 functions, and the algorithm is more stable. When the comparison algorithm calculates f4 and f5, the worst values are 2.2 and 5.7, respectively. It can be seen that the algorithm has not found the optimal solution, and at the same time it has fallen into the local extreme value. When the algorithm in this paper calculates f6, it can be seen from the best value, the worst value and the variance that the algorithm has fluctuations, but it is smaller than the genetic algorithm. The difference between the algorithm and the genetic algorithm in this paper is very small.

Table 2 Comparison of function optimization results

f_x	Algorithm	Best result	Worst result	Variance	CPU/s
f_1	GA	2.2×10^{-1}	$4.7 \times 10^{+1}$	$5.3 \times 10^{+2}$	0.0643
	I_GA	1.8×10^{-8}	1.8×10^{-8}	1.8×10^{-46}	0.0579
f_2	GA	3.4×10^{-1}	8.9×10^{-1}	5.5×10^{-2}	0.0541
	I_GA	1.9×10^{-5}	1.9×10^{-5}	0	0.0584
f_3	GA	6.2×10^{-3}	9.8×10^{-3}	3.2×10^{-6}	0.0628
	I_GA	1.8×10^{-11}	9.1×10^{-11}	1.2×10^{-21}	0.0559
f_4	GA	3.1×10^{-1}	2.2×10^{-0}	8.5×10^{-1}	0.0608
	I_GA	9.5×10^{-9}	9.5×10^{-9}	0	0.0574
f_5	GA	7.4×10^{-1}	5.7×10^{-0}	5.9×10^{-0}	0.0537
	I_GA	1.3×10^{-4}	1.3×10^{-4}	0	0.0596
f_6	GA	8.1×10^{-0}	$2.0 \times 10^{+2}$	$1.1 \times 10^{+3}$	0.0534
	I_GA	2.6×10^{-5}	1.0×10^{-1}	3.5×10^{-4}	0.0573

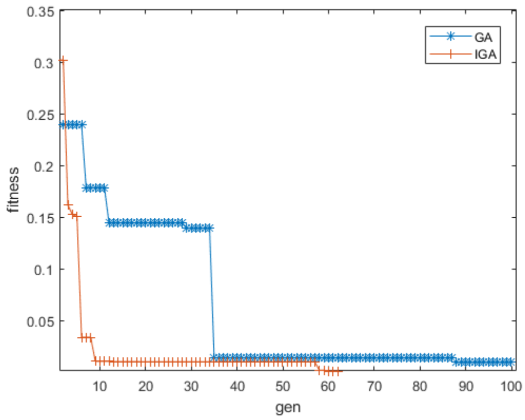

Fig. 7 comparison of f3

Due to space limitations, this article takes the test function f3 as an example. Figure 7 is a comparison between the results of the algorithm and genetic algorithm for calculating the f3 function. It can be seen from the figure that in the first generation, the algorithm obtained in this paper is larger than the result obtained by the genetic algorithm. But the algorithm in this paper has searched for the optimal value near the 10th generation. The comparison algorithm has a drop in the 5th and 10th generation, indicating that the algorithm jumped out of the local extremum and searched for a better value. The comparison algorithm has remained level in the 10-35th generation, indicating that the algorithm is at a local extremum and cannot be jumped out. The comparison algorithm searches around the global optimal value around the 35th generation and keeps it until the 88th generation. The comparison algorithm converges in the 88-100th generation algorithm, and the accuracy of the solution is 10^{-3}. In this paper, the algorithm is in the local search state from the 9th to 58th generation, and the global optimal value is searched in the 62nd generation algorithm. It can be seen intuitively from Figure 7 that the algorithm in this paper converges faster than the genetic algorithm, and the accuracy of the solution is higher. Figure 7 can illustrate that the global search ability and local search ability of the algorithm in this paper are better than the genetic algorithm.

4. Conclusion

This paper proposes an improved genetic algorithm based on binary coding. The algorithm in this paper discards the original roulette strategy; and the cross strategy selects multiple individuals to maximize the inheritance of excellent genes to the next generation; the mutation strategy uses a double gene mutation method to enhance the algorithm's ability to jump out of local extreme values; add elite Feedback strategy to improve the convergence speed and solution accuracy of the algorithm; use adaptive crossover, mutation probability, and dynamic adjustment algorithm. The algorithm of this paper simulates and tests 6 test functions. Compared with the standard genetic algorithm, it can be intuitively seen from the data and the legend that the algorithm of this paper has faster convergence speed and higher solution accuracy.

References

[1] Li Yan, Yuan Hongyu, Yu Jiaqiao, Zhang Gengwei, Liu Keping. A review of the application of genetic algorithms in optimization problems [J]. Shandong Industrial Technology, 2019(12): 242-243+180 (In Chinese).

[2] He Miao, Zhang Chuan. Genetic algorithm-based terminal area trajectory optimization method [J/OL]. Flight mechanics: 1-5 [2020-07-01]. https://doi.org/10.13645/j .cnki.fd20200602.001(In Chinese).

[3] Li Yuanfeng, Li Zhangwei, Qin Zihao, Hu Jun, Zhang Guijun. Research on transportation problems based on Monte Carlo similarity genetic algorithm [J/OL]. Computer Science: 1-11[2020-07-01].http:// kns.cnki.net/ kcms/ detail/ 50.1075. TP. 20200513. 1621. 036. Html (In Chinese).

[4] Zhang Haopeng, Guo Yu, Tang Pengzhou, Huang Shaohua, Liu Daoyuan, Li Han. Tracking registration method of augmented reality assembly system based on image matching [J/OL]. Computer integrated manufacturing system: 1-18[2020-07-01]. http://kns.cnki.net/ kcms/ detail/ 11.5946.TP.20200113.0938.004.html(In Chinese).

[5] Yao Hongliang, Zhang Qin, Yang Peiran, Wen Bangchun. Parameter optimization method of piecewise linear stiffness nonlinear energy trap [J]. Journal of Northeastern University (Natural Science Edition), 2019, 40(12): 1732-1738(In Chinese).

[6] Wu Chaofeng, Yang Zhen, Cao Wenhui, Guo Donghai. Error prediction of outer trajectory landing point based on GA-BP algorithm[J]. Journal of Ordnance Equipment Engineering, 2019, 40(12): 67-71(In Chinese).

[7] Zhu Jie, Zhang Wenyi, Xue Fei. Storage optimization of three-dimensional warehouse based on genetic simulated annealing algorithm [J]. Computer Applications, 2020, 40(01): 284-291(In Chinese).

[8] Li Yaohua, Liang Chen, Wei Dongdong. Research on aeronautical material dynamic scheduling planning based on adaptive genetic algorithm (English) [J]. Machine Tool and Hydraulics, 2020, 48(06): 200-208(In Chinese).

[9] Sun Bo, Jiang Ping, Zhou Genrong, Dong Dianyong. AGV path planning based on improved genetic algorithm [J]. Computer Engineering and Design, 2020, 41(02): 550-556(In Chinese).

[10] Wang Chunyang, Zhao Yuqing, Xie Jinxing, Su Bentang. Improvement of genetic algorithm mutation operator [J]. Journal of Shandong Agricultural University (Natural Science Edition), 2019, 50(05): 898-901(In Chinese).

[11] Wen Wu, Guo Youqing. Improvement of Apriori algorithm combined with genetic algorithm [J]. Computer Engineering and Design, 2019, 40(07): 1922-1926(In Chinese).

[12] He Bin, Zhang Jiexin, Zhang Fuqiang. An improved genetic algorithm for solving job shop scheduling problems [J]. Manufacturing Automation, 2018, 40(08): 113-117(In Chinese).