Journal of computer science and software applications
https://www.mfacademia.org/
ISSN:2377-0430
Vol. 2, No. 3, 2022

Integration of K-Means Clustering in Genetic Algorithms to Optimize Convergence and Maintain Diversity

Isabella Hernandez

Xavier University

Isabella9@gmail.com

Abstract:

To mitigate premature convergence in genetic algorithms and to maintain population diversity as iterations increase, a genetic algorithm incorporating K-means clustering is proposed. The integration of the K-means clustering method within the genetic algorithm involves an initial division of the population into distinct subpopulations. Selection, crossover, and mutation operations are then conducted independently within each subpopulation, facilitating local optimization. The optimal value from each subpopulation is preserved, ranked, and the overall optimal value is retained to prevent entrapment in local optima. The effectiveness of this enhanced algorithm is subsequently evaluated using standard test functions.

Keywords:

K-means,genetic algorithm,Algorithm optimization.

1. Introduction

Genetic algorithm was proposed by Professor Holland in 1975. Later, Dejong, Goldberg, and others summarized the algorithm for simulating biological evolution. Genetic algorithm (GA) comes from Darwin's theory of evolution, Weizmann's theory of species selection and Mendel's theory of population genetics. GA is a type of self-organizing and adaptive artificial intelligence technology that simulates the evolution process and mechanism of natural organisms to solve extreme value problems [1].

The basic idea of genetic algorithm is an algorithm formed by simulating the genetic mechanism of nature and the theory of biological evolution. It follows the principle of "natural selection, survival of the fittest", which is the same as the survival law of natural organisms, so it has a solid biological foundation [2]. Genetic algorithm is an optimization algorithm based on global probability search, and there is no hard requirement on the nature of the optimization function. Therefore, genetic algorithms have been widely applied in function optimization, image processing, path planning, neural network training, workshop management, and other fields [3]. The earliest genetic algorithm studied was the Simple Genetic Algorithm (SGA), which was summarized by Goldberg. It has the following drawbacks: slow convergence speed and cannot guarantee convergence to the global optimal solution; In the subsequent improvement process, Bian Runqiang et al. proposed an improved genetic algorithm [4], in which the crossover operation adopts a close relative crossover avoidance strategy and adopts optimal mutation to accelerate the convergence speed. However, the cross avoidance strategy of close relatives is not conducive to generating diverse populations.

2. Algorithm Introduction

2.1. K-means clustering

Clustering refers to the process of dividing individuals in the entire population into several clusters, each with similar individuals. It is also an effective method to maintain and increase the diversity and distribution of population evolution. The k-means clustering algorithm was proposed by MacQueen JB [5] in 1967 and is widely used in many clustering algorithms in science and industry, with a significant impact. When using the k-means algorithm, the first step is to determine the number of K initial cluster centers. Then, using the principle of nearest distance, each object is assigned to the nearest cluster center to obtain a class, and the current center of each cluster is calculated. As a new aggregation point, using the iterative update method, each object is reassigned to the nearest aggregation point, and the geometric center of each class will be used as the reference point for the next iteration. The iteration makes the selected reference point closer to the true geometric center of the class, with the maximum similarity between inter class objects and the minimum similarity between inter class objects [6]. When the termination condition is met, the algorithm ends and replaces the original initial cluster with a new cluster.

2.2. Genetic Algorithm

2.2.1. Selection Operation

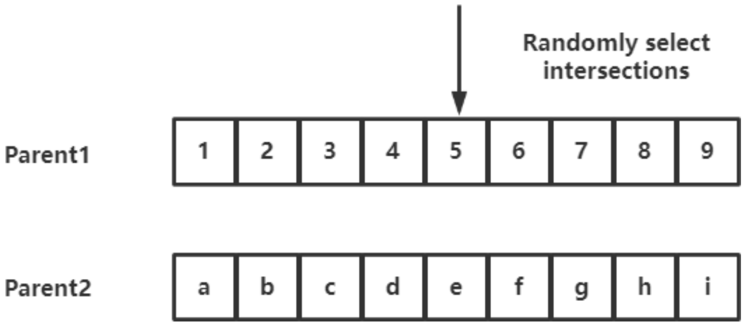
Selection refers to the operation of selecting excellent individuals from the entire population to eliminate inferior individuals. The selection in genetic algorithm is based on natural selection in the theory of biological evolution. This selection is based on the fitness value of individuals in the population as an indicator. Through selection, excellent individuals can be inherited to the next generation. There are many methods for selecting operations, such as elite retention strategy, roulette selection, random selection, and so on. This paper adopts roulette wheel selection, which is also called proportional selection. As the name says, the probability of an individual being selected in a population is positively related to the size of its fitness value. The selected individual thus enters the next step of genetic algorithm.

The probability of being selected is:

$$P_i = f_i / \sum_{i=1}^{n} f_i \tag{1}$$

2.2.2. Cross operation

The individuals generated by the selection operation are crossed according to different crossing operations. Crossing is the combination of two selected individuals and the use of a certain crossing method to replace, reorganize, and other changes in the individual's partial structure, thereby generating new individuals. The crossing methods include single point crossing, two point crossing, Davis command crossing, and so on. This article uses a single point crossover, and the operation is as follows:

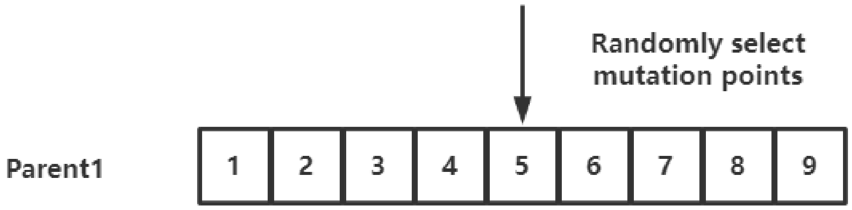


Individuals generated after crossing:

In this process, two individuals exchange some chromosomes to form a new individual. This operation can better complete the global search task and effectively improve the search ability of genetic algorithms.

2.2.3. Mutation operation

The mutation operation in genetic algorithm is performed after the crossover is completed, and it is generated based on the biological genetic variation in the theory of biological evolution. In this operation, a chromosome is randomly selected from the population based on the mutation probability to perform mutation on certain positions of the selected individual's chromatin. The mutation methods are also diverse, including basic bit mutation, Gaussian mutation, uniform mutation, etc. This article uses basic positional variation and operates as follows:



After mutation operation:

Mutation has two main functions: one is to improve the local random search ability of genetic algorithms. When the solution approaches the optimal value through the crossover operation of genetic algorithm, the use of mutation operation can enhance its local random search ability, thereby accelerating the convergence of the solution towards the optimal value; The second is to maintain the diversity of the population in genetic algorithms, thereby preventing the occurrence of premature puberty.

3. KM-GA algorithm

3.1. Combination of k-means clustering and genetic algorithm

Due to the problem of premature convergence and falling into local optima in simple genetic algorithms, a new improved algorithm has been proposed through research. The improved KM- GA algorithm first adds K-means clustering method before genetic algorithm, and divides it into different small populations through K-means clustering. Then, select, cross, and mutate in each small population separately, perform local optimization, retain the optimal value in each population, sort, and retain the overall optimal value to avoid falling into local optima.

After partitioning through the K-means clustering algorithm, local optimizations are carried out separately. The combination of K-means clustering and genetic algorithm can more effectively avoid the algorithm falling into local optima. After the local optima are completed, integration is carried out to more effectively achieve global optima [9].

3.2. Adaptive crossover and mutation operators

The selection of crossover and mutation probabilities in simple genetic algorithms also affects the overall convergence rate and whether the algorithm is excellent. A high probability of crossover will enhance the search ability of the algorithm, but at the same time, it will also make the population prone to degradation; A lower probability of crossover can also easily trap the algorithm into local optima. The mutation probability also has the same problem. The mutation probability can ensure the diversity of the population, but a higher mutation probability will make the algorithm easy to approach random search; A lower probability of variation cannot guarantee the diversity of the population effectively.

The lower probability of crossover and mutation can ensure that excellent individuals are not easy to be destroyed, which is suitable for the situation where there is a large difference in fitness between individuals in the population; However, when the individual fitness values in the population are not very different, it means that the algorithm tends to converge at this time, and the higher probability of crossover and mutation can effectively improve the search ability of the algorithm [10].

So in order to improve the search ability and performance of genetic algorithms, it is feasible to change the probability of crossover and mutation in genetic algorithms. Therefore, this paper adds an adaptive mechanism on the basis of simple genetic algorithm, and uses the traditional adaptive probability formula to obtain the changing crossover and mutation probability, so that individuals can make corresponding adjustments according to the current fitness value, thus effectively avoiding the emergence of population precocity and the algorithm jumping out of the local optimum [11]. The crossover and mutation probability formula:

Adaptive crossover probability:

$$P_c = \begin{cases} k_1 & , f_c \leq f_{avg} \\ k_2 \frac{f_{max}-f_c}{f_{max}-f_{avg}}, & f_c > f_{avg} \end{cases} \tag{1}$$

Wherein, is the size of fitness of two crossed individuals with higher fitness, $k_1$and k_2 is the constant of 0.8, and $k_1 \geq k_2$.

Adaptive mutation probability:

$$P_m = \begin{cases} k_3 & , f_m \leq f_{avg} \\ k_4 \frac{f_{max}-f_m}{f_{max}-f_{avg}}, & f_m > f_{avg} \end{cases} \tag{2}$$

Where is the size of the fitness of the current mutant, $k_3$and k_4 is a constant of 0.5, and $k_3 \geq k_4$.

3.3. Algorithm steps

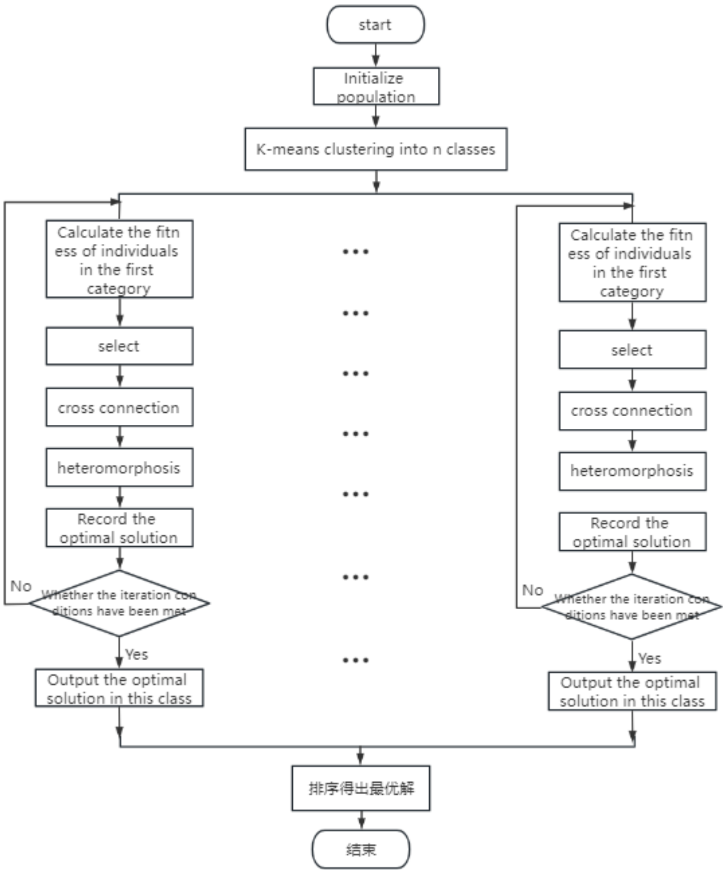


Fig. 1 Algorithm flowchart

(1) Using real number encoding, randomly generate N particles as the initial population;

(2) Divide the population into several subpopulations based on K-means clustering;

(3) Construct an appropriate fitness function to calculate the initial value of the population;

(4) Using roulette wheel, adaptive crossover, and mutation operators, iterate separately and record the optimal values of each subpopulation;

(5) Determine the termination condition, and if the termination condition is met, output the result and end the algorithm; Otherwise, the population will be used as the new initial population, and the algorithm will enter (3) to start a new iteration process

(6) After the iteration, several subpopulations obtain several optimal values and sort them to obtain the final optimal value.

4. Experimental Design and Results

4.1. Test Functions

In order to verify the feasibility of the improved algorithm, this article selects typical complex high-dimensional functions as the object of numerical simulation verification. In theory, the optimal value of this function is 0. The test function is as follows:

Sphere function:

$$f(x_i) = \sum_{i=1}^{D} x_i^2 \tag{3}$$

4.2. Experimental results

The parameter settings are as follows: the initial population size is 100, the iteration algebra is set to 100 generations, and the dimension is set to 10 dimensions.

4.3. Experimental Analysis

The improved KM-GA algorithm was tested with the same parameters, and the experimental results are as follows:

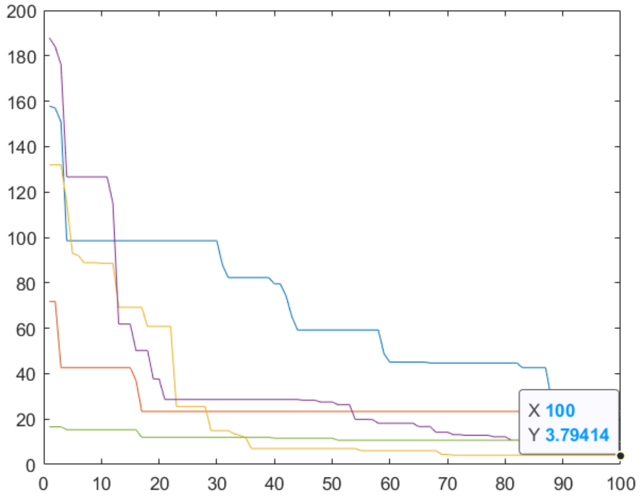


Fig. 2 Experimental results of improved KM-GA algorithm

The running results are shown in Figure 2. With an initial population size of 100 and 100 iterations, the minimum value obtained is 3.97414. From the figure, it can be seen that the algorithm proposed in this paper can effectively jump out of local optima. Therefore, the algorithm proposed in this paper can achieve certain results, but there are still shortcomings that need further improvement.

5. Conclusion

Genetic algorithm has always been plagued by the problem of being trapped in local optima. Inspired by the ideas of genetic algorithm and k-means algorithm, this algorithm has made improvements in solving function optimization problems. After conducting experiments on typical test functions, the results showed that the improved K-means clustering parallel genetic algorithm proposed in this paper has better performance than the standard genetic algorithm, and can effectively jump out of local optima.

References

[1] Ma Xiaoshu, Li Yulong, Yan Lang. Comparative review of traditional multi-objective optimization methods and multi-objective genetic algorithms [J]. Electrical Transmission Automation, 2010,32 (03): 48-50+53.

[2] Chai Yan, Zhou Yanzhao. Improvement of Mountain Climbing Method in Genetic Algorithm [J]. Journal of Liaoning University of Engineering and Technology (Natural Science Edition), 2014,33 (07): 996-999.

[3] Ge Jike, Qiu Yuhui, Wu Chunming, Pu Guolin. Overview of Genetic Algorithm Research [J]. Computer Application Research, 2008 (10): 2911-2916.

[4] Bian Runqiang, Chen Qiang, Yuan Zhuzhi. An Improved Genetic Algorithm and Its Application in system identification [J]. Control and Decision, 2000 (05): 623-625+640.DOI: 10.13195/j. cd. 2000.05.111.bianrq.029.

[5] J. MacQueen. Some methods for classification and analysis of multivariate observations[P]. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics,1967.

[6] Wang Sen, Liu Chen, Xing Shuaijie. Overview of K-means clustering algorithm research [J]. Journal of East China Jiaotong University, 2022, 39 (05): 119 126. DOI: 10.16749/j.cnki. jecjtu. 20220914.001.

[7] Jin Ling, Liu Xiaoli, Li Pengfei, Wang Yan. Overview of Genetic Algorithms [J]. Chinese Journal of Science, 2015 (27): 230.

[8] Li Yan, Yuan Hongyu, Yu Jiaqiao, Zhang Gengwei, Liu Keping. Overview of the Application of Genetic Algorithms in Optimization Problems [J]. Shandong Industrial Technology, 2019 (12): 242 243+180. DOI: 10.16640/j.cnki.37-1222/t.2019.12.210.

[9] Geng Yue, Ren Junhao, Ji Peiqi. Hybrid genetic algorithm clustering research based on K-Means mutation operator [J]. Computer Engineering and Applications, 2011,47 (29): 151-153+221.

[10] S. Rasoul Safavian,David A. Landgrebe. A survey of decision tree classifier methodology.[J]. IEEE Trans. Systems, Man, and Cybernetics,1991,21(3).

[11] Shan Yuhan. Research on rectangular layout method based on adaptive genetic algorithm [J]. Computer and Digital Engineering, 2020,48 (10): 2343-2347+2399.

[12] S. Rasoul Safavian,David A. Landgrebe. A survey of decision tree classifier methodology.[J]. IEEE Trans. Systems, Man, and Cybernetics,1991,21(3).