2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)

Improved genetic algorithm based on K-Means to solve path planning problem

Tingyu Ma
Jinan University
School of Electrical Engineering
Zhuhai City, Guangdong Province
804974193@qq.com

Tao Wang
Jinan University
School of Electrical Engineering
Zhuhai City, Guangdong Province
767320864@qq.com

Dongsong Yan*
Jinan University
School of Electrical Engineering
Zhuhai City, Guangdong Province
* Corresponding author: Tyands@qq.com

Jianshuo Hu
Jinan University
School of Electrical Engineering
Zhuhai City, Guangdong Province
2877747854@qq.com

***Abstract*—This paper regarded the path planning problem as a TSP problem, and an improved genetic algorithm based on K-Means was proposed. The algorithm firstly decomposes the large-scale TSP problem into multiple small TSP problems through K-Means clustering, and then optimizes each cluster separately using the improved genetic algorithm, and finally merges all clusters to obtain the final route. Experiments show that compared with the standard genetic algorithm, the improved algorithm avoids the solution falling into the local optimal solution and accelerates the convergence of the algorithm. Therefore, this algorithm can save cost and improve efficiency when applied to path planning.**

Keywords- path planning; TSP; K-Means; clustering; genetic algorithm

I. INTRODUCTION

Path planning refers to finding a moving track that meets a certain performance index from the initial point to the target point under certain constraints. This problem is the traveling sales-man problem (TSP). The traveling salesman problem (TSP) is the problem of searching the shortest Hamiltonian route among all target points. The earliest mathematical model was proposed by Dantzig (1959) and others. It is one of the most classic NP difficult combinatorial optimization problems and has great value in economics and engineering. Therefore, it has always been concerned by researchers.

The algorithms used to solve the TSP fall into two categories: exact algorithm and heuristic algorithm. Accurate algorithms can ensure that the final solution is optimal. For example: branch and bound method, linear programming algorithm [1]. But this kind of algorithm cannot deal with the large-scale exponential explosion problem effectively. Therefore, the search for an effective approximate solution has greater use value. In the 1980s, people combined heuristic algorithms with artificial intelligence, biological evolution, neural systems and other fields to form modern intelligent optimization algorithms, the combination of these algorithms makes it possible to solve TSP in a short time. Among them, Yang *et al.* proposed a multi-scale adaptive quantum free particle optimization algorithm for heuristic algorithms to solve slow problems in solving combinatorial optimization problems[2]. Jiang *et al.* used CUDA technology on GPU to implement parallel genetic algorithm to solve TSP, which proves that this method is more powerful in global optimization and takes less time[3]. Qiao *et al.* proposed an improved strategy to solve the problem of slow convergence and premature stagnation of the ant colony algorithm, while maintaining a superior global search capability[4]. Gao & Meng proposed a greedy stochastic adaptive gray wolf optimization algorithm based on the two defects that the TSP problem is easy to cause the population diversity to decline and fall into local optimality as the number of iterations increases [5]. Luo applied the genetic algorithm to the Hopfield network to solve the problem of poor convergence of the Hopfield network when solving the TSP, while improving the solution rate[6].

Genetic Algorithm (GA) is an algorithm based on natural selection. It is characterized by searching from the set of problem solutions, dealing with multiple individuals in the group at the same time, no need to search for spatial knowledge or other auxiliary information, and its fitness function is not constrained by continuous differentiability, and the definition domain of fitness function can be set arbitrarily, These characteristics are not possessed by traditional optimization methods, so genetic algorithm has great advantages in solving TSP[7].

However, due to the randomness of the genetic algorithm, the results are volatile and the calculation efficiency is low. With the increase of the data scale, there are great requirements for the calculation time and calculation capability. Therefore, we considered clustering analysis of TSP through K-Means clustering algorithm, using improved genetic algorithm to optimize various types, and finally connecting each cluster into a complete path. In this way, the algorithm can get the approximate solution more quickly and accurately.

II. METHOD

A. Improved genetic algorithm

1) Path planning problem

978-1-7281-9668-8/20/$31.00 ©2020 IEEE
DOI 10.1109/ISPDS51347.2020.00065

Assuming that the number of target points is n, the set of target points is as follows:

$$A = \{a_1, a_2, \dots, a_n\} \tag{1}$$

2) Coding

This paper mainly uses path notation coding. Encode each target in A with a natural number from 1 to n. A chromosome represents the path through n target points. For example, when the traversed target point is 10, the individual code of the particle is [8 5 7 6 2 1 3 4 9 10], which means that the target point will be traversed from the target point 8, after 5, 7, 6, 2,… , 10 finally returns to the target point 8, thus completing the TSP traversal[8].

3) Population initialization

In this paper, the greedy algorithm is used to optimize the initial individuals. First, a target point is randomly selected as the current target point, and it is added to the individual. Then searching all the target points that are not added to the individual, and finding the target point nearest to the current target point, add it to the individual as the current target point, continue to search and add the next nearest target point, until all the target points join the individual, thus the initial optimization of the individual. The experimental results show that the initial population generated by the greedy algorithm is random, and the overall quality is improved, which is helpful to speed up the optimization[9].

4) Fitness function

The fitness function can be said to be the natural living environment in group evolution and a natural choice for simulating the survival of the fittest in the evolution of the biological world. Therefore, in the genetic algorithm, how to rationally design the fitness function is crucial[10].Let the $D_{a_i a_j}$ is the actual distance from the target point a_i to the target point a_j, the individual's fitness expression is as follows:

$$fitness = \frac{1}{\sum_{i=1}^{n-1} D_{a_i a_j} + D_{a_n a_i}} \tag{2}$$

Formula 2 shows that the fitness function is the reciprocal of the total distance traveled through all target points and then back to the starting point. Taking the fitness function value as the optimization goal, the chromosome with the larger fitness function value has a better circuit.

5) Selection

Roulette selection is also called fitness proportional selection method or Monte Carlo selection method. In this method, the selection probability of each individual is proportional to its fitness value[11].Let the population size be n, where the fitness of individual i is f_i, then the probability P_{si} that i will be selected is as follows:

$$P_{si} = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{3}$$

It can be seen from formula 3 that the probability P_{si} reflects the proportion of the fitness of individual i in the sum of individual fitness of the entire group.The greater the individual fitness, the higher the probability of being selected, on the contrary, the smaller the individual fitness, the lower the probability of being selected.

6) Crisscross

The selected path is cross-operated, several gene loci are identified in the parent gene loci, and then two parent individuals are selected at random for gene exchange, resulting in new individuals. First, two positions where the crossover is needed are selected, and then the two parent individuals are exchanged numerically at the position.

7) Mutation

According to a certain probability of mutation, the genes in chromosomes are mutated to produce new offspring. Mutation operation can also speed up the convergence speed of the algorithm. The concrete steps are: In an individual's vector, select two locations that need to be mutated, and exchange the target point codes on the two locations[12].

8) Termination

According to the number of iteration, if it is reached, the calculation stops, and the chromosome corresponding to the value of the current optimal fitness function is selected as the optimal solution, that is, the optimal path; If it is not reached, the algorithm enters to the next loop calculation.

9) Output the path map

According to the location map of each point in the coordinates, the path map is output on the basis of the calculated results.

B. K-Means

The classic clustering Algorithm is K-Means. It has the advantages of wide applicability, fast clustering speed and low complexity. The function of the algorithm is to solve the clustering problem, and classify the data objects into different clusters according to the condition that the data objects in the same cluster have high similarity and the data objects in other clusters have low similarity[13].

The algorithm is divided into the following steps:

Step 1: Enter n data objects.

Step 2: Determine k, the number of clusters, and then randomly select k cluster center points from the data object $C_i (1 \le i \le k)$.

Step 3: Calculate the Euclidean distance between the remaining data objects and the cluster center point C_i, then compare the distances and classify them into the cluster of the cluster center closest to it.

The Euclidean distance between the data object and the cluster center in space is calculated as follows:

$$d(x, C_i) = \sqrt{\sum_{j=1}^{m} \left(x_j - C_{ij}\right)^2} \tag{4}$$

Where, x is the data object,$C_i$is the $i-th$ cluster center, m is the dimension of the data object， x_j and $C_{ij}$are the $j-th$ attribute value of x and C_i.

Step 4: Through calculation, all the data and the mean value of the objects in k clusters are taken as the new cluster center.

Step 5: If the cluster center no longer changes or reaches the maximum number of iterations, then the iteration is stopped and the result is output, otherwise the third step cycle is returned.

C. *Improved genetic algorithm based on K-Means*

When using genetic algorithm to solve large-scale TSP problem, there is the problem of too long solution time. In this paper, using the idea of "Divide and rule" and the simple K-means algorithm, we decompose the large-scale TSP into several small-scale TSP problems, and then solve each small-scale TSP problem with the improved genetic algorithm proposed in the previous section, then, the results of the subproblem are connected among clusters to obtain a satisfactory solution of the original TSP[14].

III. RESULTS

In order to verify the performance of the Algorithm, a simulation experiment is carried out in the environment of MATLAB R2019b. The subjects were selected from Eil101, Eil51, Berlin52, PR76 data in the TSPLIB Test Library. Taking Eil101 as an example, the first step is to cluster the target points.

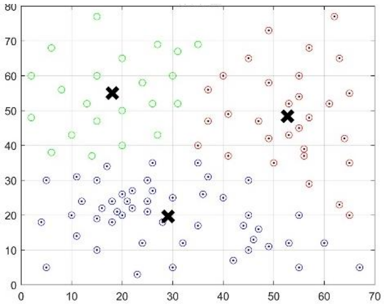

Figure 1. Target point clustering.

Then, genetic algorithm is used to find the shortest path for each cluster.

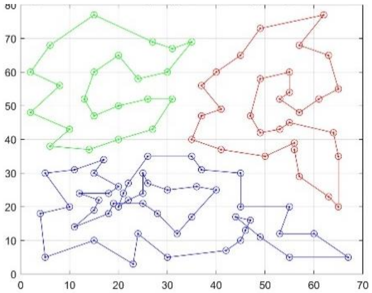

Figure 2. The shortest path of each cluster.

Finally, all clusters are connected as a whole according to the connection rules.

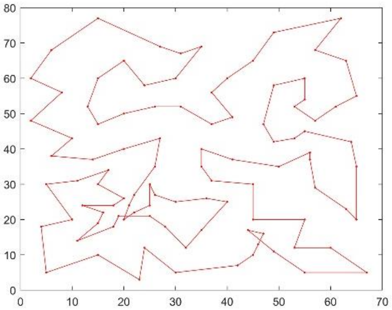

Figure 3. Total path map.

The standard genetic algorithm and the improved genetic algorithm based on K-means are used to solve the TSP problem, and the basic parameters of the two algorithms are the same. To prevent the computation time from being too long, the initial population is set to 500 and the maximum genetic algebra is 200. Each is repeated 20 times, and the simulation results are as follows: (EIL51)

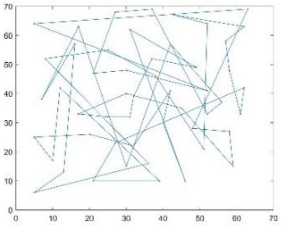

Figure 4. Test results of standard genetic algorithm on Eil51.

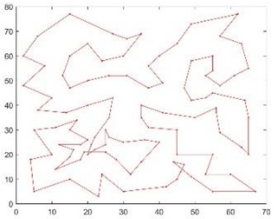

Figure 5. Test results of the improved algorithm on Eil51.

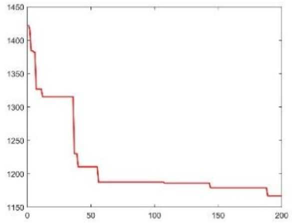

Figure 6. The curve of path length changing with iteration times under Eil51 test of standard genetic algorithm.

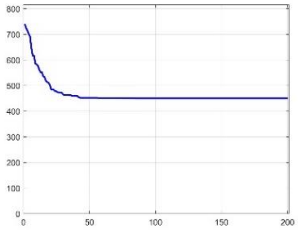

Figure 7**.** The curve of path length changing with iteration times under Eil51 test of the improved algorithm.

Figures 4~7 show a comparison of the output using the standard genetic algorithm and the improved genetic algorithm based on K-means in different cases. It can be seen that the improved genetic algorithm can simplify the complex path and shorten the total path length. At the same time, the convergence speed of the improved algorithm is faster than that of the standard algorithm. After about 160 iterations, they both converge. But the effective solution of the improved algorithm is obviously better than that of the Standard Algorithm, which avoids falling into the local optimum.

TABLE I. COMPARISON OF AVERAGE SOLUTION AND OPTIMAL SOLUTION BETWEEN STANDARD GENETIC ALGORITHM AND IMPROVED ALGORITHM.

		eil51	berlin52	pr76	eil101
Path total length based on standard genetic algorithm	Average solution	1214.2387	21379.0076	434038.9148	2679.7467
	Optimal solution	1159.725	19724.1724	403852.7135	2603.5654
Path total length based on improved genetic algorithm of K-Means	Average solution	448.5532	8392.3653	115068.9299	694.4413
	Optimal Solution	440.4593	8263.1140	110202.7221	672.8239

It can be seen from table1 that the improved genetic algorithm based on K-Means has higher solving ability than standard genetic algorithm. And the more the number of target points, the more obvious the advantages of the algorithm. The total path obtained by the improved genetic algorithm based on K-means is 1 / 4 to 1 / 2 of the total path obtained by the standard genetic algorithm, which greatly shortens the total path distance. Therefore, the improved genetic algorithm based on K-means for path planning can save cost and improve efficiency.

IV. DISCUSSION

The percentage deviation of the average solution from the optimal solution of the algorithm compared with the known optimal solution in the TSPlib library is derived from formula 5 and formula 6:

$$PD_{av} = \frac{S_{ave} - S_{know\_best}}{S_{know\ best}} \tag{5}$$

$$PD_{best} = \frac{S_{best} - S_{know\_best}}{S_{know\ best}} \tag{6}$$

The current TSPlib optimal solutions for Eil51, Berlin52, Pr76 and Eil101 are: 108159,7542,629. The deviation percentages of the improved genetic algorithm based on K-means on Eil51 are 5.29% and 3.39% respectively, and on Berlin52 are 11.27% and 9.56% respectively, the deviation percentages obtained on Pr76 were 6.38% and 1.88% respectively, and on Eil 101 were 10.4% and 6.96% respectively.

V. CONCLUSION

This paper presents an improved genetic algorithm based on K-means for path planning. Divide-and-conquer method was used to decompose the large-scale TSP problem into several sub-problems, which greatly shortens the path length. The experimental results show that the proposed algorithm is superior to the genetic algorithm in solving the optimal path, and the optimal path length is 1 / 4 to 1 / 2 of that of the standard genetic algorithm, which can be used to solve the problem well. At the same time, it takes a much shorter time.

REFERENCES

[1] Ma,L. An overview of Travelling Salesman Problem's algorithms . Practice and understanding of mathematics,2000(02):156-165.

[2] Yang, Y.T., Wang, P. Multi Scale Quantum free particle optimization algorithm for solving Travelling Salesman Problem. Computer application,1-6.

[3] Jiang,J.W. Parallel Genetic Algorithm based on GPU for TSP [C]. System Simulation Professional Committee of CAS, Simulation Technology Application Professional Committee of CAS, University of Science and Technology of China. Proceedings of the 20th annual conference on system simulation technology and its applications in China（20th CCSSTA 2019）. System Simulation Committee of CAS, Simulation Technology Application Committee of CAS, System Simulation Committee of University of Science and Technology of China,2019:560-563.

[4] Qiao,D.P.,Pei,J.,Li,H.,Wen,X.Y. Research on improved ant colony algorithm for TSP[J]. Mechanical design and manufacturing, 201910:144-149.

[5] Gao,S.,Meng,L. Greedy stochastic adaptive grey Wolf Optimization Algorithm for TSP [J]. Modern Electronics,2019,42(14):46-50+54.

[6] Luo Y.F. Design and Improvement of Hopfield network for TSP[C]. Hubei Zhongke Institute of Geology and Environment Technology、Liverpool John Moores University,UK.Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science(AICS 2019). Hubei Zhongke Institute of Geology and Environment Technology、Liverpool John Moores University,UK: Hubei Zhongke Geological and Environmental Technology Service Center,2019:98-102.

[7] Wang,D.C. Application of an improved genetic algorithm in TSP[J]. Proceedings of the Liaoning University of Technology, 2019,3904:235-239.

[8] Liu,H.H.,Cui,C.,Chen,J. An improved genetic algorithm for Travelling Salesman Problem [J]. Journal of the Beijing Institute of Technology, 2013,3304:390-393.

[9] Yu,Y.Y.,Chen,Y.,Li,T.Y. An improved genetic algorithm for Travelling Salesman Problem [J]. Control and decision making, 2014,2908:1483-1488.

[10] Shi,L.P. Improved genetic algorithm for solving TSP problem [J]. Experimental Technology and Management, 2014, 31 (07): 61-64.

[11] Sun,L.,Hu,X.L.,Yu,L. Research on TSP of route planning based on genetic algorithm [J]. Journal of Bullets and Guidance, 2005 (S4): 446-448.

[12] Jiao,Z.T. Research on Forest Fire Prevention UAV System [D]. Xi'an University of Technology, 2019.

[13] Yang,J.C.,Zhao,C. Review of K-Means clustering algorithm research [J] .Computer Engineering and Applications, 2019,55 (23): 7-14 + 63.

[14] Cui,W.,Wu,Y.H. An improved genetic algorithm based on K-means to solve TSP [J]. Logistics Technology, 2011,30 (17): 160-162.

[15] Xu,N. Research on genetic algorithm for a large-scale TSP problem [D]. Xidian University, 2011.