An Improved Genetic Algorithm and its Application in Routing Optimization

Jianwei Wang, Wenjuan Sun*
School of Computer and Artificial Intelligence, Chaohu University, HeFei, China

***Abstract*—Traditional routing algorithms can't adapt to the complex and changeable network environment, and the basic genetic algorithm can't be applied to solving routing optimization problems directly because of the lack of coding methods. An improved basic genetic algorithm was purposed to find the optimal or near-optimal routing. The network model and mathematical expression of routing optimization problem was defined, and the routing problem was transformed into a problem of finding the optimal solution. In order to meet the specific needs of network routing optimization, some key improvements of GA have been made, including the design of coding scheme, the generation of initial population, the construction of fitness function and the improvement of crossover operator and mutation operator. The simulation results of two typical network environments show that the improved GA has excellent performance in routing optimization. Compared with Dijkstra algorithm and Floyd algorithm, the improved GA in this paper not only has excellent robustness and adaptability in solving routing optimization problems, but also can effectively cope with the dynamic changes of network environment, providing an efficient and reliable routing solution for dynamic network environment.**

Keywords—Improvement of genetic algorithm; routing optimization; shortest path; crossover operator; mutation operator

I. Introduction

In the modern computer network environment, the process of determining the transmission path of data packets, that is, routing, is facing complex challenges. With the continuous expansion of the network and the increasing traffic, the network structure has become more and more complex, and the traditional routing methods have been difficult to meet the needs of modern networks. It has become very important to find an effective and least costly routing strategy to improve network efficiency and reduce congestion, which urges researchers to find more effective routing optimization methods.

Routing refers to arranging the communication link between the source node and the served object. Routing strategies commonly used are divided into fixed routing, flooding routing, random routing and adaptive routing [1]. The concrete forms of the shortest path problem include: the shortest path problem of determining the starting point, the ending point, the starting and the ending point and the global shortest path problem. Optimization tools are often used to find the optimal or near-optimal routing scheme in solving the shortest path optimization problem. Genetic Algorithm (GA) is a method to simulate natural evolution and find the optimal solution proposed by John Holland [2] in 1970s. Genetic algorithm is widely used in many fields as its strong global optimization ability, and some scholars have applied GA to solve routing optimization problems.

Obeidat A et al. [3] proposed a network routing method based on GA. Moza M et al. [4] put forward a method based on GA to find the k shortest paths in the network. Zhao Feng [5] described the realization principle of intelligent search algorithm such as genetic algorithm in dynamic routing optimization of computer network. Wang et al. [6] proposed a multi-path routing algorithm for WSN based on genetic algorithm. The fitness function was determined by calculating the node spacing, and a shared routing scheme was generated at the base station. Because of the fixed-length coding, the best path may be limited and the global optimization cannot be achieved. Gao Xia et al. [7] selected GA for routing operation, improved GA and applied it to the routing problem of WSN. The above research adopts GA directly or improves GA to solve the routing problem or the shortest path problem, but all of them are optimized under the same solution length condition, no research is made on paths with different lengths.

In view of the shortcomings of the above research, this paper designs the coding scheme, generates the initial population, constructs the fitness function and improves the core operation of the genetic algorithm. In order to reduce the time spent in path finding, the shortest path adaptive routing problem with simultaneous determination of starting point and ending point is optimized.

II. Improvement of Genetic Algorithm

A. Basic Genetic Algorithm

Genetic algorithm is an optimized search algorithm that simulates the natural selection and genetic mechanism of organisms. The optimization process of GA includes six processes: population initialization, individual evaluation, selection operation, crossover operation, mutation operation and termination condition judgment [8]. Fig. 1 shows the operation steps of the genetic algorithm.

The basic genetic algorithm is more suitable for finding an optimal path to traverse the whole network graph as the length of chromosomes generated by coding is fixed. The research background of this paper is that the optimal path of two nodes found by the basic genetic algorithm has certain constraints, and the optimal path may not be global optimal. The improved genetic algorithm in this paper uses variable-length chromosomes to encode the routing path, determines the neighborhood nodes of each node in the network in advance, and improves the crossover and mutation operations according to the characteristics of the path to prevent unqualified paths. When forming a routing path, factors such as the distance between path nodes, the total energy consumption of the path and the residual energy of the nodes are considered, and the fast global optimization ability of the genetic algorithm is integrated into the routing optimization to search the target routing path efficiently and comprehensively.

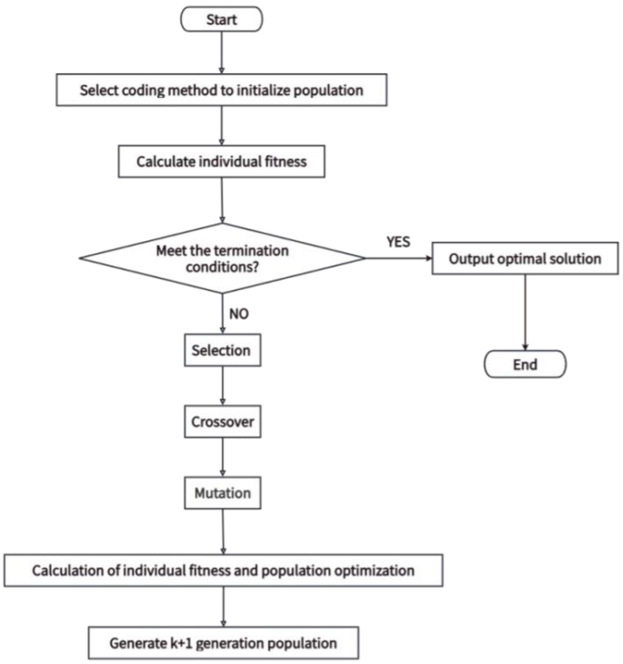

Fig. 1. Flow chart of genetic algorithm.

B. *Mathematical Network Model of Shortest Path Routing Problem*

Shortest path algorithm is a classical algorithm problem in graph theory, which aims to find the shortest path between two vertices in a graph. The shortest path refers to the path with the smallest sum of weights on each side that starts from a vertex and reaches another vertex along the edge of the graph.

The weighted undirected graph G can be used to represent the network when studying the routing problem, and G=(V,E). Where V(G) represents the vertex in graph G [9], which is the routing node, E(G) represents the set of relationships between vertices in graph G, which is the set of links between routing nodes. |V| represents the number of nodes in the network, |E| represents the number of links in the network, and the matrix W=(wij) represents the cost of link (i,j). The starting node and the destination node are represented by start and stop respectively, and the connection relationship between nodes is represented by matrix A=(aij), as shown in formula (1) [10].

$$a_{ij}=\begin{cases}1, & v_i \text{ is connected with } v_j \\ 0, & v_i \text{ and } v_j \text{ are not connected or the same}\end{cases} \tag{1}$$

If $a_{ij}=1$ and $a_{jk}=1$ indicate that it is reachable from node i to node k, all paths from *start* to *stop* can be found and the nodes can be saved in the array path, as shown in Formula (2).

$$path(i,j)=\text{node} \tag{2}$$

where 'i' is the *i*th path, 'j' is the *j*th node of the path, 'node' is the routing node number, which is a positive integer.

The problem of calculating the shortest path can be transformed into the optimization problem of the minimum value, and the objective function is as shown in Formula (3).

$$\min\sum_{j=1}^{m}\sum_{i=2}^{n} w_{path(j,i-1)\,path(j,i)} \tag{3}$$

Where 'j' represents the *j*th path, 'm' represents the number of paths, 'n' represents the length of the *j*th path.

C. *Improvement of Coding Mode and Population Initialization*

It is not suitable to adopt the coding method of basic genetic algorithm in the initial population operation as all the paths from the starting node to the destination node are different in length. Instead, the node number of the path is directly encoded and saved. This improved coding method is not only beneficial to the selection of fitness function and the calculation of fitness, but also more suitable for solving the shortest path routing problem.

In order to store the information of the network diagram effectively, it can be saved by transforming it into an adjacency matrix, which is an n-order matrix w, as shown in Formula (4).

$$W[i][j]=\begin{cases}w_{ij}, & \text{Represents the cost required from node i to node j} \\ 0, & i=j \text{ or nodes i and j have no direct path}\end{cases} \tag{4}$$

Saving information in this way is beneficial to coding and implementation. The matrix W is an $n\times n$ matrix, and W[i][j] represents the network consumption required to reach the node j from the node i. If the value is 0, it means that node i=j or there is no direct path from node i to node j.

The generation of the initial population should meet the requirement that the individuals in the generated initial population can't have open circuits and loops, otherwise, the individuals obtained in the next operation will have a high probability of open circuits and loops, which will lead to the increase of unreachable and cost, and there will be mistakes in solving the shortest path routing problem. In order to make the generated individuals meet the requirements, the starting point is input, starting from the starting point, a node directly connected with the starting point is selected randomly and add it into the individual. Find out whether there is a node connected directly to this node, if there is a next node connected to it, continue to add it, and so on until the end point is found. The way to save chromosomes in this paper is to initialize a zero array. Save the nodes into the array in turn according to the rules generated by individuals until the end point is saved. If there are redundant zero elements, they will be ignored in code recognition.

There is a path from node B to nod M (B,D,H,L,N,P,O,M). The path is saved as (2,4,8,12,14,16,15,13,0,0). The path length is 8, so the first 8 elements save the node, and all other elements are 0. As shown in Fig. 2, the initial random population value is 10.

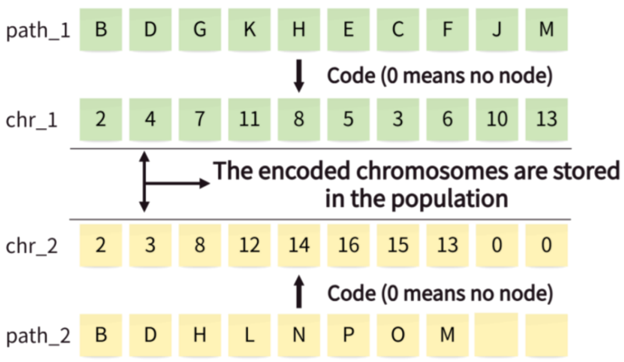

Fig. 2. Coding process diagram.

D. Design of Fitness Function

In genetic algorithm, fitness function is a very important concept. Fitness function is a mapping of optimization objectives, and each individual will be given a fitness value. The higher the fitness value, the more suitable the individual is for survival and reproduction. The fitness value is calculated and the parent is selected from the population. The calculation result of fitness value is used as the basis for selecting the parent, and roulette is used to select more excellent individuals. The fitness function can be used to measure the quality of chromosomes in the current iteration. The definition of fitness function in this scheme design is shown in Formula (5) [11].

$$f\left(path_i\right)=1-\frac{len(i,1)-\min len}{\max len-\min len+0.01} \tag{5}$$

where 'path$_i$' represents the ith path, 'len' represents the total cost of path consumption, 'maxlen' represents the cost of the path with the largest total cost, and 'minlen' represents the cost of the path with the smallest total cost.

The advantages and disadvantages of the solution can be compared by calculating the fitness of each individual, which is the comparison condition of iterative updating.

E. Design of Selection Operator

In order to avoid the premature convergence of the algorithm, this study adopts the method of combining the optimal individual retention strategy with roulette algorithm in population selection. Let the population size be m, select the n best individuals with the highest fitness in each round and keep them directly in the next generation population, and select the remaining M-N individuals by roulette algorithm. In roulette algorithm, the probability that an individual is selected is directly proportional to fitness [12].

F. Improvement of Crossover Operator

The traditional single-point or multi-point crossover operation can't be adopted in this study as the coding method is different from the basic GA. Traditional crossover is likely to lead to open circuit and evolutionary failure. The traditional single-point crossover is shown in Fig. 3.

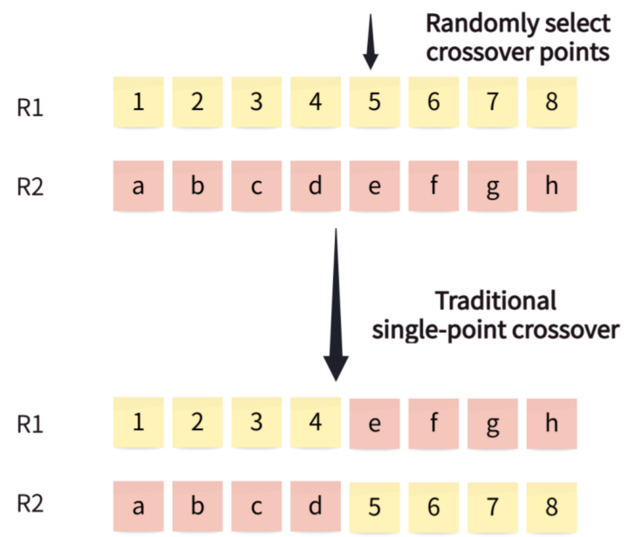

Fig. 3. Single-point crossover.

An improved single-point crossover method was proposed in order to solve the shortest path routing problem by GA. Different from the traditional single-point crossover, the improved single-point crossover can only be operated at nodes that are repeated except the starting point and the ending point of two paths, as shown in Fig. 4 [12].

The specific process is as follows:

a) Two individuals R_1 and R_2 from a population are selected.

b) Generate a pseudo-random number randomly, comparing it with the crossover probability. Judge whether to perform crossover operation or not. If yes, proceed to (c), otherwise, do not crossover.

c) Judge whether there are duplicate nodes except the start node and the end node. If so, save the nodes and carry out (d), otherwise, not crossing,

d) Select a node from the saved nodes randomly as a crossing point.

e) Cross the nodes after the crossover of R_1 and R_2 to obtain new individuals R_1' and R_2',

*f)*Perform loop elimination on R_1' and R_2' by eliminating the loop function. If there is no loop, it will not be eliminated and exit the function.

g) The two individuals after processing are the individuals without loops obtained by crossover.

The idea of eliminating loop function: suppose that the individual obtained after crossover operation is (B,D,H,L,O,P,N,L,I,M). The individual has a loop while (L,O,P,N,L) exists. (O,P,N,L) needs to be deleted from the individual. When an individual has more than one loop, the loop cancellation function can be called at the end of the loop cancellation function, which can ensure that the treated individual do not contain loops. Loop elimination is shown in Fig. 5. The crossover probability is 0.9.

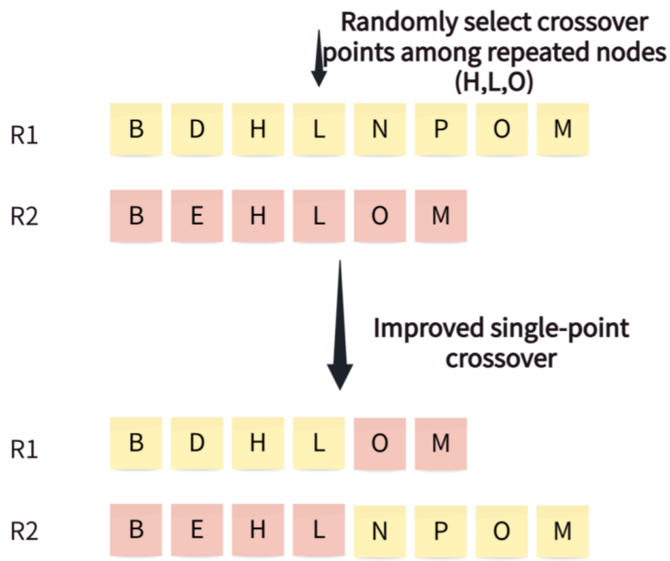

Fig. 4. Improved single-point crossover.

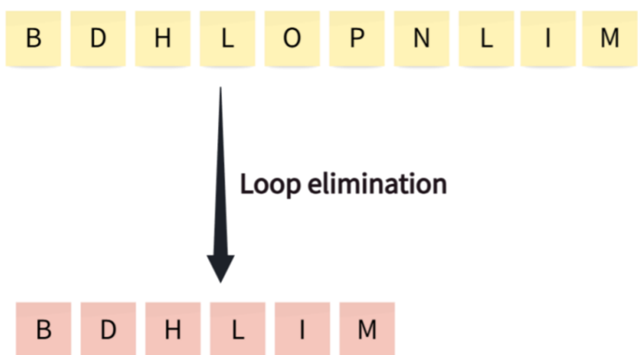

Fig. 5. Loop elimination diagram.

G. Improvement of Mutation Operator

The introduction of mutation operator can not only improve the diversity of the population, but also improve its ability to explore the unknown solution space, thus avoiding premature convergence. The mutation operation of the basic genetic algorithm includes basic bit mutation and reverse mutation commonly, as shown in Fig. 6.

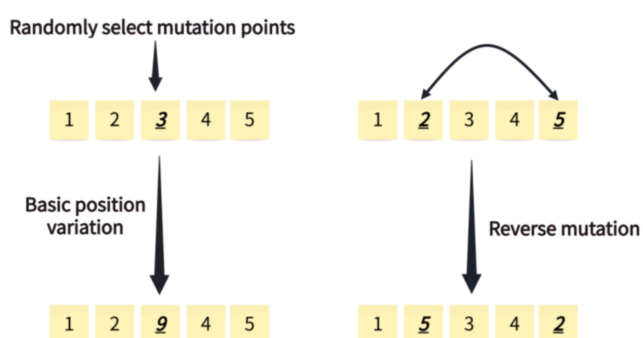

Fig. 6. Mutation operation.

If random mutation is applied to solve the shortest path routing optimization problem directly, it is easy for individuals to have open circuit or loop phenomenon because of the uncertainty. Therefore, this paper proposes an improved mutation operation.

The specific operation process is as follows:

a) Select an individual R_1 from a population.

b) Generate a pseudo-random number randomly, and comparing it with the mutation probability. Judge whether to perform crossover operation or not, if yes, performing (c), otherwise, not mutating.

c) Select a node *x* randomly except the starting node and the destination node.

d) Find all nodes directly connected with node *x* and save them in the aggregate.

e) Select a node *y* from the aggregate randomly.

*f)*Generate a shortest path P_1 from the starting node to the node *y*.

g) Generate a shortest path P_2 from node *y* to the destination node.

h) Merge paths P_1 and P_2 to obtain a new individual $R_1{}^{'}$.

*i)*A new mutated individual is obtained after loop elimination of the newly obtained individual $R_1{}^{'}$. The individual does not have open circuit and loop.

The above operation process is shown in Fig. 7. The mutation probability is 0.05.

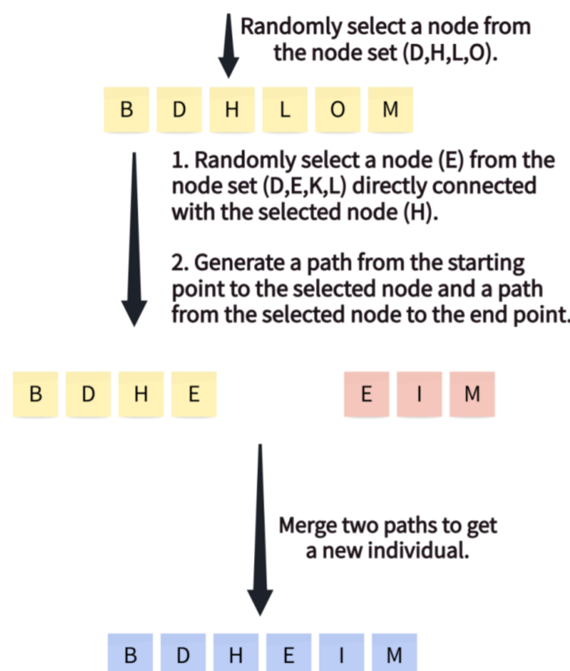

Fig. 7. Improved mutation operation.

H. Flow of Improved GA

The specific implementation process of the improved genetic algorithm in this paper is shown in Fig. 8. The execution flow of the improved crossover operator and mutation operator is described in detail.

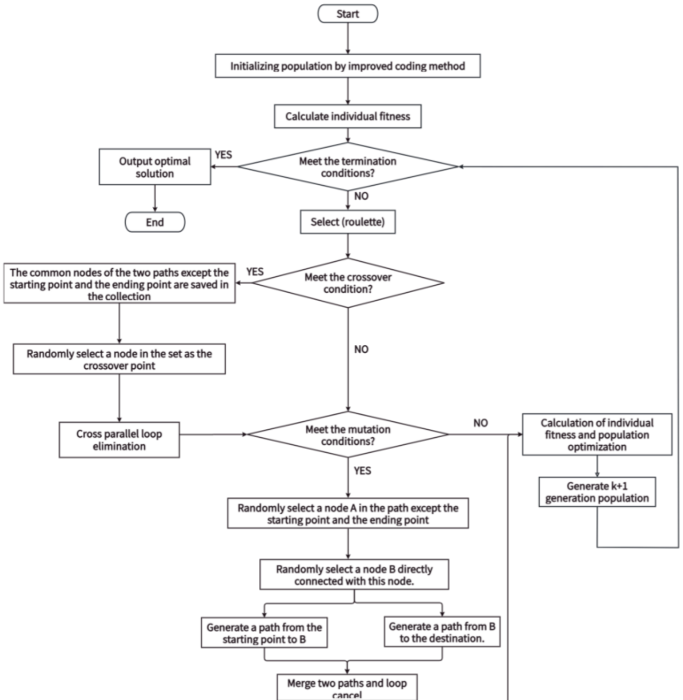

Fig. 8. Flow chart of improved genetic algorithm in this paper.

III. Comparative Analysis of Improved Genetic Algorithm in Routing Optimization Problem

In order to verify the effectiveness of the improved genetic algorithm in solving routing optimization problems, this paper chooses Dijkstra [13] algorithm and Floyd [14] algorithm as two classical adaptive routing algorithms for comparative analysis.

A. *Algorithm Parameter Setting*

The influences and differences of network nodes and different algorithms on routing are compared and analyzed by setting different values for network nodes and adopting different algorithms. See Table I for the specific parameter settings of the three algorithms.

TABLE I. PARAMETER TABLE

Parameters	Values
Number of network nodes	16/30
Population size	10
Iterations	20
Crossover probability	0.9
Mutation probability	0.05

The simulation of the above three algorithms is realized by MATLAB software in Windows10.

Two typical network environments will be selected for simulation and comparative analysis.

B. *Simple Network Environment*

The network consists of 16 nodes, which are represented by capital letters A-P respectively. The starting point is B and the ending point is P. The connection between nodes and the cost required are shown in Fig. 9. The red path represents the best routing path.

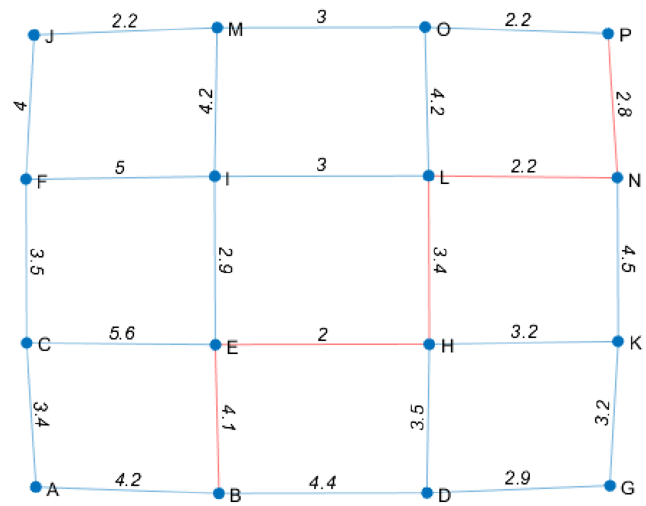

Fig. 9. Simple network environment.

The improved genetic algorithm in this paper is used to find the path from node B to node P The path result is shown in Fig. 10. The shortest path optimized by the improved genetic algorithm in this paper is $B \rightarrow E \rightarrow H \rightarrow L \rightarrow N \rightarrow P$. The cost of this path is 14.5, and the time to find this path is 0.006208 seconds.

```
Improved GA results are as follows:
Total cost:14.5
Elapsed time is 0.005810 seconds.
B -> E -> H -> L -> N -> P.
```

Fig. 10. Result of simple network.

The number of iterations of the improved genetic algorithm is 20. The evolution process of the shortest path length in each population, that is, the minimum required cost, with the number of iterations is shown in Fig. 11. The initial shortest path cost is between 22 and 23, and the optimal solution is 14.5 with the continuous evolution of population and genetic iteration.

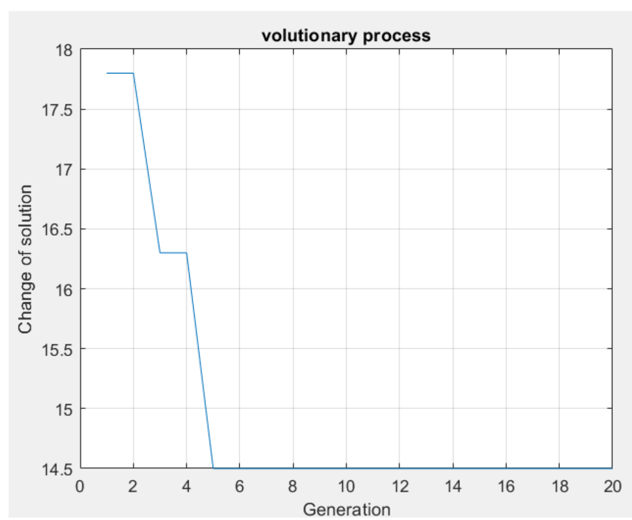

Fig. 11. Iterative process diagram of improved GA in this paper under simple network.

The comparison results of the improved genetic algorithm, Dijkstra [13] algorithm and Floyd [14] algorithm are shown in Table II. The shortest path obtained by the three algorithms is $B \to E \to H \to L \to N \to P$, and the cost of this path is 14.5. But the time of the three algorithms is different. The improved genetic algorithm in this paper takes the shortest time, which is 0.005810 seconds.

TABLE II. SIMPLE NETWORK RESULTS

Algorithm	The shortest path	Path cost	Time (s)
Improved GA in this paper	(B,J,M,O,Q,R)	14.5	**0.005810**
Dijkstra	(B,J,M,O,Q,R)	14.5	0.014124
Floyd	(B,J,M,O,Q,R)	14.5	0.018501

After repeated experiments for ten times, it is concluded that the time spent by the three algorithms is shown in Table III, and the time spent by the three algorithms in finding the way is shown and compared with the line chart as shown in Fig. 12. The results show that the improved genetic algorithm proposed in this paper is faster than Dijkstra algorithm and Floyd algorithm for finding the shortest path problem. The improved genetic algorithm is more efficient and has shorter running time when dealing with optimization problems.

TABLE III. SIMPLE NETWORK TIME STATISTICS

	Improved GA (s)	Dijkstra algorithm (s)	Floyd algorithm (s)
1	0.002859	0.0111225	0.0096672
2	0.003286	0.0124257	0.0121836
3	0.0030729	0.0123028	0.0117582
4	0.0035445	0.0127774	0.0137152
5	0.0029833	0.0124634	0.0124018
6	0.0027576	0.0156065	0.0133303
7	0.0027295	0.0136018	0.0123962
8	0.0029044	0.0150465	0.011854
9	0.0034026	0.0129538	0.0108338
10	0.0044602	0.0135091	0.0118217

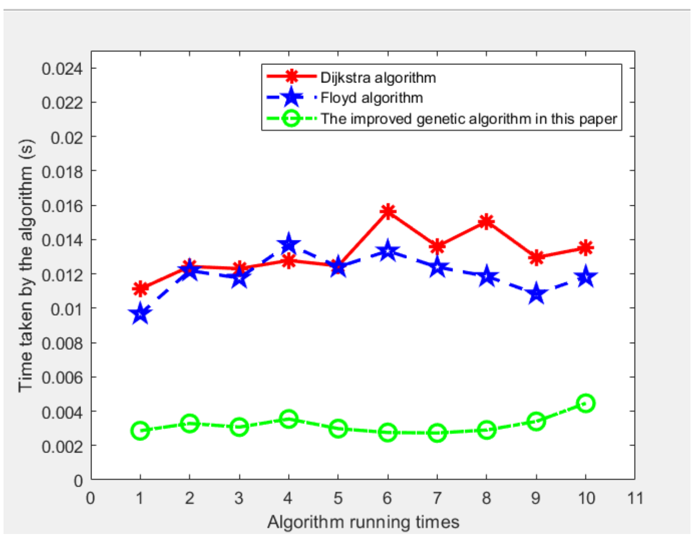

Fig. 12. Comparison of calculation time for simple network.

T-test the time obtained by 10 repeated experiments of three algorithms in simple network environment. The results are shown in Table IV and Table V.

TABLE IV. ANALYSIS RESULTS OF SIMPLE NETWORK IGA AND DIJKSTRA PAIRED T TEST

Name	Pairing (Average±Standard deviation)		Difference (Paired 1-Paired 2)	t	p
	Pairing 1	Pairing 2			
Improved GA pairing Dijkstra	0.003200±0.000521	0.013181±0.001332	-0.009981	-21.292659	0.000

As can be seen from the above table, the paired t test is used to study the differences of experimental data. As can be seen from the above table, a total of one group of paired data will show differences ($p<0.05$). According to the specific analysis, there is a significant level of 0.01 between improved GA in this paper and Dijkstra ($t=-21.293$, $p=0.000$), and the specific comparison shows that the average value of improved GA in this paper (0.0032) will be significantly lower than that of Dijkstra (0.01318095).

TABLE V. ANALYSIS RESULTS OF SIMPLE NETWORK IGA AND FLOYD PAIRED T-TEST

Name	Pairing (Average±Standard deviation)		Difference (Paired 1-Paired 2)	t	p
	Pairing 1	Pairing 2			
Improved GA pairing Floyd	0.003200±0.000521	0.011996±0.001154	-0.008796	-22.215274	0.000

As can be seen from the above table, the paired t test is used to study the differences of experimental data. As can be seen from the above table, a total of one group of paired data will show differences ($p<0.05$). The specific analysis shows that there is a significant level of 0.01 between improved GA in this paper and Floyd ($t=-22.215$, $p=0.000$), and the specific comparison shows that the average value of improved GA in this paper (0.0032) will be significantly lower than that of Floyd (0.0119962). A total of 1 set of paired data will all show differences.

C. Complex Network Environment

The improved genetic algorithm in this paper can adjust the parameters according to the number of network nodes, so as to find the best routing path in complex network environment. Because too many nodes will make the network structure diagram difficult to distinguish, this paper chooses a network composed of 30 nodes in complex network environment. In practical application, the improved genetic algorithm can be applied to a larger and more complex network environment.

The network consists of 30 nodes, which are represented by *Route1-Route*30 respectively. The starting point is *Route 1* and the ending point is *Route 29*. The connection situation and the cost between nodes are shown in Fig. 13. The nodes will be referred to as 1-30 for short, and the red path represents the best routing path.

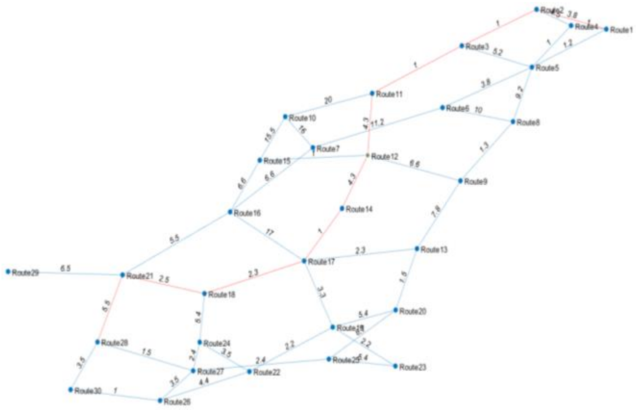

Fig. 13. Complex network environment.

Using the improved genetic algorithm to find the path from node 1 to node 29, the result is shown in Fig. 14. The shortest path found by the improved genetic algorithm is $1 \to 2 \to 3 \to 11 \to 12 \to 14 \to 17 \to 18 \to 21 \to 28$, the cost of this path is 25.7, and the time to find this path is 0.007212 seconds.

```
Improved GA results are as follows:
Total cost:25.7
Elapsed time is 0.007212 seconds.
Route1 -> Route2 -> Route3 -> Route11 -> Route12 -> Route14 -> Route17 -> Route18 -> Route21 -> Route28.
>>
```

Fig. 14. Experimental results of complex network.

The number of iterations of the improved genetic algorithm is 20. The evolution process of the shortest path length in each population, that is, the minimum required cost, with the number of iterations is shown in Fig. 15.

The initial shortest path cost is between 42 and 44, and the optimal solution is 25.7 with the continuous evolution of population and genetic iteration.

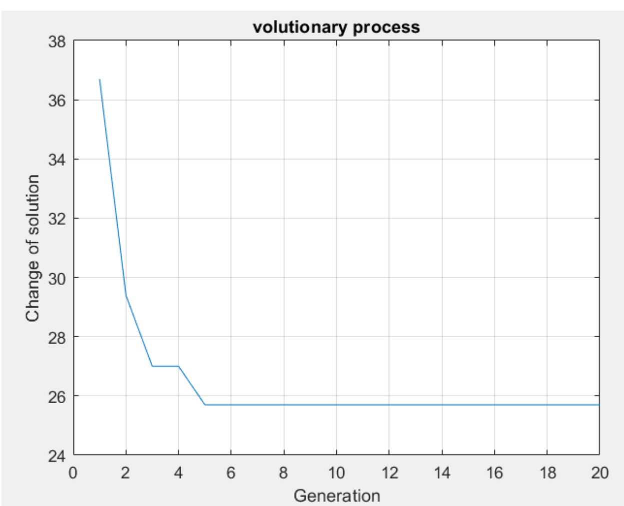

Fig. 15. Iterative process diagram of improved GA in this paper under complex network.

The comparison results for complex network using the improved genetic algorithm, Dijkstra [13] algorithm and Floyd [14] algorithm are shown in Table VI. The shortest path obtained by the three algorithms is $1 \to 2 \to 3 \to 11 \to 12 \to 14 \to 17 \to 18 \to 21 \to 28$, and the cost of this path is 26.7. But the time of the three algorithms is different. The improved genetic algorithm in this paper takes the shortest time, which is 0.007212 seconds. Compared with Dijkstra algorithm and algorithm, the improved genetic algorithm can find the optimal routing scheme faster in the two network environment, especially in complex or dynamic network topology, and the improved genetic algorithm shows better adaptability and optimization ability.

TABLE VI. COMPLEX NETWORK RESULTS

Algorithm	The shortest path	Path cost	Time (s)
Improved GA in this paper	(1,2,3,11,12,14,17,18,21,28)	25.7	**0.007212**
Dijkstra	(1,2,3,11,12,14,17,18,21,28)	25.7	0.010099
Floyd	(1,2,3,11,12,14,17,18,21,28)	25.7	0.011007

After repeated experiments for ten times, it is concluded that the time spent by the three algorithms is shown in Table VII, and the time spent by the three algorithms in finding paths is shown and compared by line charts as shown in Fig. 16. From this figure, we can find that the improved genetic algorithm proposed in this paper shows obvious advantages in performance, and the path-finding time is shorter and more stable.

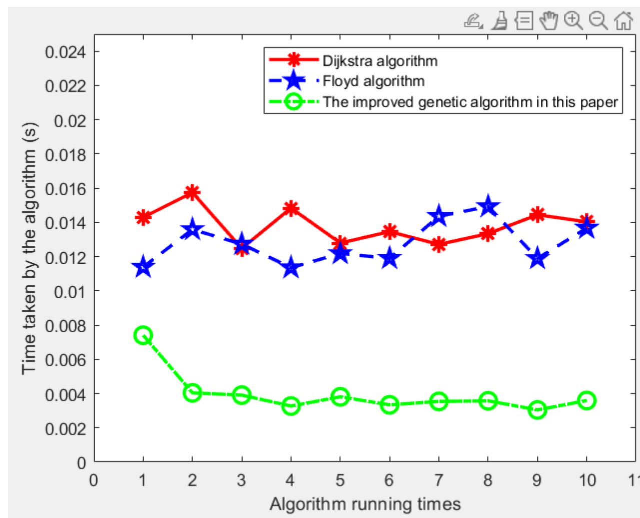

Fig. 16. Comparison of calculation time for complex network.

TABLE VII. COMPLEX NETWORK TIME STATISTICS

	Improved GA (s)	Dijkstra algorithm (s)	Floyd algorithm (s)
1	0.0074056	0.0142925	0.0113845
2	0.0040412	0.0157401	0.013584
3	0.0039107	0.0124758	0.0127307
4	0.0032705	0.0148344	0.0113531
5	0.0038107	0.0127909	0.012179
6	0.0033431	0.0134553	0.0119015
7	0.0035433	0.0127115	0.0143559
8	0.0035778	0.0133404	0.0149225
9	0.0030448	0.014441	0.0118913
10	0.003596	0.0140289	0.0136652

T-test the time obtained by 10 repeated experiments of three algorithms in complex network environment. The results are shown in Table VIII and Table IX.

TABLE VIII. ANALYSIS RESULTS OF PAIRED T TEST OF IGA AND DIJKSTRA IN COMPLEX NETWORKS

Name	Pairing (Average±Standard deviation)		Difference (Paired 1-Paired 2)	t	p
	Pairing 1	Pairing 2			
Improved GA pairing Dijkstra	0.003954±0.001249	0.013811±0.001045	-0.009857	-20.495974	0.000

As can be seen from the above table, the paired t test is used to study the differences of experimental data. As can be seen from the above table, a total of one group of paired data will show differences ($p<0.05$). According to the specific analysis, there is a significant level of 0.01 between improved GA in this paper and Dijkstra ($t=-20.496$, $p=0.000$), and the specific comparison shows that the average value of improved GA in this paper (0.00395437) will be significantly lower than that of Dijkstra (0.01381108). A total of 1 set of paired data will all show differences.

TABLE IX. ANALYSIS RESULTS OF PAIRED T TEST OF IGA AND FLOYD IN COMPLEX NETWORKS

Name	Pairing (Average±Standard deviation)		Difference (Paired 1-Paired 2)	t	p
	Pairing 1	Pairing 2			
Improved GA pairing Floyd	0.003954±0.001249	0.012797±0.001265	-0.008842	-13.851679	0.000**

As can be seen from the above table, the paired t test is used to study the differences of experimental data. As can be seen from the above table, a total of one group of paired data will show differences ($p<0.05$). According to the specific analysis, there is a significant level of 0.01 between improved GA in this paper and Floyd ($t=-13.852$, $p=0.000$), and the specific comparison shows that the average value of improved GA in this paper (0.00395437) will be significantly lower than that of Floyd (0.01279677). A total of 1 set of paired data will all show differences.

IV. CONCLUSION

An improved genetic algorithm is proposed and applied to solve the shortest path routing problem in order to improve the network performance. Compared with the traditional routing algorithm Dijkstra and Floyd algorithm, the improved genetic algorithm has excellent performance in dealing with the changeable network topology and dynamic changes, thus verifying its remarkable advantages in network routing optimization. The improved GA has stronger adaptability and better optimization ability in two typical network environments, which provides a novel and effective solution to the network routing problem.

ACKNOWLEDGMENT

This work is financially supported by the Natural Science Foundation of Anhui Province under Grant No.2023AH052100 and the Funding for Talent Project of Chaohu University under Grant No.KYQD-202218.

REFERENCES

[1] XC Guo, Cache placement and routing in wireless networks, Nanjing, Nanjing University, 2021.

[2] Holland J, Adaptation in natural and artificial systems : an introductory analysis with application to biology. Control & Artificial Intelligence 1975.

[3] A.Obeidat, AM.Shalabi, An Efficient Approach towards Network Routing using Genetic Algorithm, international journal of computers communications control, 17(5), 2022,pp.1-12.

[4] M. Moza, S. Kumar, K. Finding, Shortest Paths in a Network Using Genetic Algorithm, International Journal of Computer Network and Information Security(IJCNIS), 12(5),2020,pp.56-73.

[5] Zhao Feng. Application of Intelligent Search Algorithm in Dynamic Routing Optimization of Computer Network,Integrated Circuit Application, 41(05),2024.pp.292-293.

[6] WANG S,JIANG Y Z.Multipath routing based on genetic algorithm in wireless sensor networks,Mathematical problems in engineering, 2,2021.pp.1-6.

[7] X. Gao, RJ. Li. Application of improved genetic algorithm in WSN routing problem, Journal of Inner Mongolia University (Natural Science Edition), 51(03),2020,PP.322-328.

[8] WJ. Zhan, YK. Li. Overview of related research on path planning based on improved genetic algorithm, Computer and Digital Engineering, 51(7),2023,PP.1544-1550.

[9] C. Sun. Research on approximate algorithm of combinatorial optimization problem based on graph structure, Beijing University of Technology, 2022.

[10] X. Tang, XY. Xu, SM. Pan, Intelligent Routing Algorithm Based on Graph Convolution Neural Network, Computer Engineering, 48(3), 2022,PP. 38-45.

[11] Liu Yang, Cao Lijia, Yang Xu. Application of improved genetic algorithm in multi-AGV scheduling, computer applications and software, 41(04),2024,pp.86-89+105.

[12] Ding Fan, He Junyi, Chen Suxia, et al. Routing optimization of wireless sensor networks based on genetic algorithm,Journal of Henan Institute of Technology (Natural Science Edition), 35(04),2023,pp.39-44.

[13] T. Qi, SW. Zhang, FF. Chen. Traceability of turning traffic based on high-speed big data and Dijkstra algorithm, transportation enterprise management, 39(02),2024,PP.73-75.

[14] X. Long. Analysis and optimization design of public transport network in Chongqing city based on Floyd algorithm, Chongqing University of Posts and Telecommunications, 2021.