

Retraction

Retracted: Research on Optimal Route Planning for Self-Driving Tour Based on Road Network Structure

Security and Communication Networks

Received 8 August 2023; Accepted 8 August 2023; Published 9 August 2023

Copyright © 2023 Security and Communication Networks. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] P. Zong, Y. Han, and C. Xu, "Research on Optimal Route Planning for Self-Driving Tour Based on Road Network Structure," *Security and Communication Networks*, vol. 2022, Article ID 6588288, 11 pages, 2022.

Research Article

Research on Optimal Route Planning for Self-Driving Tour Based on Road Network Structure

Ping Zong,¹ Yao Han,¹ and Chenbo Xu² 

¹Hilton School of Hospitality Management Sichuan Tourism University, Chengdu 610100, China

²Sichuan Normal University, Chengdu 610000, China

Correspondence should be addressed to Chenbo Xu; 20210083@sicnu.edu.cn

Received 5 July 2022; Revised 8 August 2022; Accepted 18 August 2022; Published 29 September 2022

Academic Editor: Hangjun Che

Copyright © 2022 Ping Zong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Today's social and economic development continues to improve people's quality of life, and private cars are widely popularized, and self-driving tours have developed. The rapid development of self-driving travel has played an important role in the development of the national economy, and self-driving travel has become popular. Because the development of self-driving tours has caused some problems, the road network structure has become more and more complex, and the roads have become very congested. Especially during the holidays, there are more private cars in tourist attractions and the roads are more congested. How to use the information of roads and attractions and then choose the optimal travel route becomes particularly important. In response to this problem, we first analyze the topology of the road network, then analyze the accessibility of scenic spots and related factors that affect self-driving travel, and use the A* algorithm, Dijkstra algorithm, and other calculation methods to calculate the optimal path. The experiment found that there are many influencing factors of self-driving travel, and the road network structure has the greatest influence on it. The A* algorithm has obvious advantages over the Dijkstra algorithm.

1. Introduction

With the continuous development of the global economy, people's demand for tourism is increasing [1], and the tourism destination focuses on the scenic spots in surrounding cities, expanding to neighboring provinces, cities, countries, and even the world. However, there is no more complete and systematic tourism route planning scheme. Therefore, this study hopes to give relevant recommendations according to the needs of tourists and plan the optimal route according to the departure and destination given by tourists [2].

Up to now, a large number of scholars have studied the related issues of tourism routes. Some scholars use the correlation function of travel time and travel cost to represent the tourist utility, because they find that the degree of crowding has an important impact on the tourist experience [3]; some scholars use A* algorithm [4], Dijkstra algorithm and other calculation methods [5], Floyd algorithm [6], "breadcrumb" fitting [7], and other research methods to plan travel routes [8].

2. Materials and Methods

We first analyze the road network structure and then calculate whether it is reachable or not. Combined with the relevant factors that affect the self-driving tour, a variety of algorithms are used to find the shortest path.

2.1. Results and Discussion. The experiment found that there are many influencing factors of self-driving travel. The road network structure has the greatest influence on it. The A* algorithm has obvious advantages over Dijkstra's algorithm.

2.2. Topological Structure Quantitative Index. This study aims to help the self-driving tourists to find a suitable travel route and to avoid spending a lot of time on the road. It also aims to effectively reduce traffic problems and reduce traffic congestion caused by holidays. Every holiday, the travel experience of tourists is often affected by traffic congestion, and it is not convenient for local residents to travel.

There are several common road network structures. A-side format: there are vertical or horizontal near-parallel roads at regular intervals. B circular radial: it consists of several circular roads connected to each other. C freestyle: the terrain is complex and has no fixed geometry. D mixed type: it is a combination of the above forms.

There are two methods for road network topology: path structure analysis and space syntax. Considering various spatial factors, this research mainly analyzes road network topology from the perspective of space syntax. In order to be applicable to various road networks, this study selects two indicators: comprehensibility and integration.

Comprehensibility: it refers to the coordination degree of the local and the whole of the road network [9].

Integration degree: the degree of scattered or dense distribution between road sections is an indicator of whether the road is congested or not. The greater the degree of integration, the more concentrated the roads, which means the more the center can be deviated and the more traffic can be attracted. The integration degree can be divided into global and local integration degrees: the global integration degree represents all road relationships of a complete road segment, and the local integration degree represents the road segment relationship of a partial area of a complete road segment, which are calculated as follows:

$$RA(x) = \frac{DM(x) - 1}{n/2 - 1},$$

$$AR = \frac{n|\log_2(n/3) - 1| + 1}{(n-1)(n-2)/2}, \quad (1)$$

$$\text{Intergration}(x) = \frac{AR}{RA(x)}.$$

In the formula, n is the number of axes; DM is the average depth; RA is the relative asymmetry; AR is the relative asymmetry value of the diamond-shaped topology; Intergration is the degree of integration.

Modern transportation is becoming more and more developed, and the road network structure is very complex. People who travel to other places are not familiar with the road network structure, and it is easy to make mistakes. Some tourists drive to Chongqing to travel, but they are not familiar with the road. They choose to follow the navigation and cannot find the place they want to go. In order to effectively avoid such situations, we analyze the road network structure and then avoid areas with complex road network structures, so that self-driving tourists can find suitable roads and scenic roads.

2.3. Accessibility Calculation. The different scenic spots to be visited by the self-driving tourist will have corresponding impacts on different groups of people and different vehicles. For example, if the road you are going to is uneven, there is deep water, and there are many mountain roads, it will be difficult for ordinary vehicles to travel, and off-road vehicles

are required to pass normally. People who do not have an off-road vehicle will have a lot of problems driving to it, and even the car will break down halfway down the road. Computational reachability is designed to avoid such problems.

By calculating the distance traveled on the GIS grid to deduce the accessibility [10] and fully considering various factors such as mountains and waters that hinder the movement, the accessibility to a certain location in the region can be better simulated. The weighted distance from one place on the GIS grid to another on the grid is calculated. This algorithm is a cumulative cost distance algorithm. The calculation formula is as follows:

$$Ki = \begin{cases} \frac{1}{2} \sum_{i=1}^n (Ci + Ci + 1), \\ \frac{\sqrt{2}}{2} \sum_{i=1}^n (Ci + Ci + 1). \end{cases} \quad (2)$$

In the formula, the time cost of the i -th pixel is Ci ; the time cost of the $i+1$ -th element along the movement direction is denoted by $Ci+1$; n represents the total number of pixels; the scenic spot accessibility of the i -th grid is denoted by Ki . The above formula is used to calculate the time cost in the horizontal or vertical direction of the grid surface, and the following formula is used to calculate the time cost in the diagonal direction of the grid surface.

2.4. Relevant Factors Affecting the Destination of Self-Driving Tours. The cost factors affecting the self-driving tour are as follows: ① gasoline cost: for ordinary private cars, highways are more than 1 yuan/km, and ordinary kilometers are 0.6 yuan/km (excluding ordinary toll roads). ② accommodation fee: the level is suitable for each person, and the standard room is 100–150 yuan per day. ③ meal fee: the amount varies from one person to another, generally speaking, about 60 yuan per person. ④ tickets: the price varies, so it is not easy to calculate; in many places, the elderly are free or half price, but tickets are still required for sightseeing cars and ropeways.

2.4.1. Tourist Factor. The scoring indicators of each scenic spot are mainly the data on major tourism websites and the data of travel notes learned by tourists [11]. The formula is as follows:

$$ti = \sum_{j=1}^s \frac{ni \cdot xij}{ni}. \quad (3)$$

where the evaluation of the j -th tourist of the tourist attraction i is represented by xij , the number of useful comments of the j -th tourist of the i -th tourist attraction is represented by ni , the sum of the useful numbers of the i -th tourist attraction is represented by ni , and the weighted score of the tourists of the i -th tourist attraction is expressed by ti .

2.4.2. Environmental Factor. It is the score data of tourists on the scenic spot environment, which is divided into very poor, poor, average, good, and very good. It is set to 1, 2, 3, 4, and 5 points [12]. The formula is as follows:

$$y_i = \frac{1}{4} \sum_{j=1}^4 X_j, \quad i = 1, 2, \dots, n, \quad (4)$$

where the three factors of scenic spot environment, traffic, and infrastructure are denoted by X_j , the comprehensive score of the i scenic spot environment is denoted by y_i , and n is the number of n scenic spots.

2.4.3. Road Factor. Relevant studies have shown that tourists prefer to travel to areas where the spatial distribution of attractions is more concentrated, which is convenient for travel. The accessibility of measuring the spatial distance is expressed by the spatial distance formula, and the formula [13] is as follows:

$$A_i = \sum_{j=1}^n l_{ij}. \quad (5)$$

Among them, the minimum distance between two scenic spots (i, j) is represented by l_{ij} , and the sum of the distances from the i scenic spot to the rest of the scenic spots is represented by A_i , and the number of scenic spots is n .

2.5. Reverse Order Recursion Model. The travel process is divided into six stages, each stage is represented by k , and S is used to represent the k -stage bit; that is, s is a dynamic variable [14], and the k -stage route is selected as Xk . The state transition equation is as follows:

$$Sk + 1 = Sk - Xk. \quad (6)$$

f_x indicates Sk the cost of the k -stage to the end point, and vk represents the k -stage distance. The recursion formula is as follows:

$$f_x = \min\{vk + fk + 1\}, \quad k = 6, 5, 4, 3, 2, 1, \quad (7)$$

where x_k^* represents the optimal decision; when the boundary condition $k=7$, $f_7=0$. The reverse order recursion equation is as follows:

$$f_k(S_k) = \min\{d(S_k, u_k) + f_{k+1}(S_{k+1})\}, \quad (8)$$

$$k = 6, 5, 4, 3, 2, 1; f_7(S_7) = 0.$$

Matlab is used to find the shortest path from the first to the sixth stage, and the following is obtained:

$$f_1(A) = \min\{d(A, C) + f_2(C), d(A, D) + f_2(D)\}. \quad (9)$$

2.6. Dijkstra Algorithm to Solve the Optimal Path. Dijkstra's algorithm is an efficient algorithm for solving the shortest path [15], and this study expresses its algorithm in the form of a flow chart. Dijkstra's algorithm will create two node sets during the search process, open, and closed sets.

The closed set represents the shortest node set to the target point, and the open represents other nodes. The starting value of the source node is 0, and the weight of other nodes is ∞ . When the target point reaches the closed set, the search is completed. The flow chart is shown in Figure 1.

2.7. DV Algorithm to Find the Optimal Path. DV algorithm, also known as Bellman-Fords routing algorithm [16], can also be used for shortest path solution. The Bellman-Ford equation is as follows:

$$dx(y) = \min_v \{c(x, v) + dv(y)\}. \quad (10)$$

In the above formula, the weight from node x to node v is denoted by $c(x, v)$, the weight of the shortest path from node x to node y is denoted by $dx(y)$, and the weight of the shortest path from node v to node y is denoted by $dv(y)$.

A, B, C, D, and E represent nodes, and c and d represent paths. Figure 2 is a schematic diagram of the Bellman-Ford algorithm.

The pseudocode of the Bellman-Ford algorithm is an intuitive embodiment of the algorithm, from which the optimal path can be directly obtained. The code calculates the optimal solution step by step [17]. The code is shown in Algorithm 1:

2.8. DI_END_CITY_DIS_and_NATURE Method. The following formula is used to evaluate the relevant cities:

$$\text{city_score} = \frac{(\sum_{i=1}^n X_i)}{n}. \quad (11)$$

In the above formula, X_i represents the popularity of the i scenic spot.

The following formula is used to find the shortest path:

$$\text{result} = \sqrt{D^2 + (1000H)^2}. \quad (12)$$

The following formula is used to derive the approximate number of kilometers traveled:

$$\text{KM} = 3 * \text{Days} * V. \quad (13)$$

Days means the number of days tourists travel, and V means the driving speed.

3. Shortest Path Optimization Algorithm

The above various algorithms are not calculated in combination with the topology structure of the road network, and the optimal path algorithm combined with the analysis of the topology structure of the road network will be used below.

3.1. A* Algorithm. Dijkstra's algorithm calculates the length of the shortest path from the source point to all other points, and A* focuses on the shortest path from point to point (including specific paths). The Dijkstra algorithm is based on a more abstract graph theory level, and the A* algorithm can be more easily used in things such as game map pathfinding.

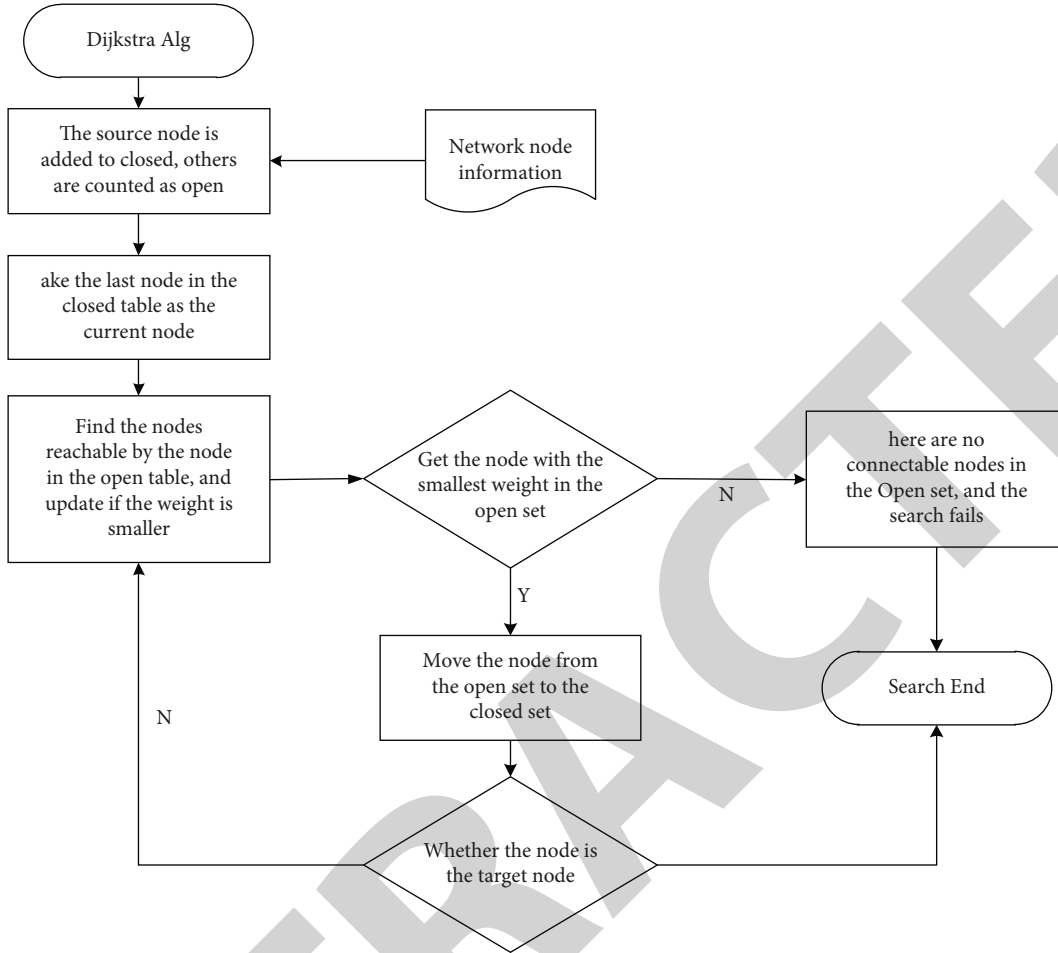


FIGURE 1: Flow chart of Dijkstra algorithm.

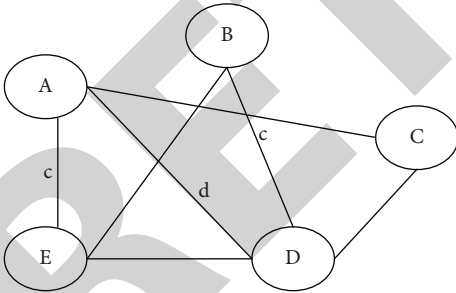


FIGURE 2: Schematic diagram of Bellman-Ford algorithm.

The essence of Dijkstra's algorithm is breadth-first search, which is a divergent search, so the space complexity and time complexity are relatively high. For the current point on the path, the A* algorithm not only records the cost to the source point, but also calculates the expected cost from the current point to the target point. It is a heuristic algorithm and can also be considered as a depth-first algorithm.

Dijkstra's algorithm is the most traditional routing algorithm [18], and there is a problem that the retrieval is blind and does not consider the target point, but only the source point. This problem can cause the search space to be too large, which is not conducive to computation.

A* algorithm is one of the optimal algorithms in static network, and A* algorithm is an improved algorithm based on Dijkstra algorithm. The A* algorithm introduces a weight function:

$$F(i) = G(i) + H(i). \quad (14)$$

The node depth is represented by b , and the performance evaluation index is represented by $b * \text{branching factor}$. The relationship can be represented by the following formula:

$$N + 1 = 1 + b * + (b *)^2 + \dots + (b *)^d. \quad (15)$$

There is a single node state, and the evaluation function $h(x)$ satisfies the formula:

$$|h(x) - h * (x)| = 0(\log h * (x)). \quad (16)$$

The A* algorithm is calculated as follows:

$$h = \frac{(n_i - n_r) \cos t_N}{\delta n_E}. \quad (17)$$

The A* process is divided into 4 steps. Step 1: set the source node $\{n_s, 0, 0, \text{null}\}$, the node enters the closed set, and the adjacent node enters the open set. The initial value of w is the current node and the source node of n_s , the link

```

Bellman-Ford Algorithm
Input  $k$ : information of network topology
Output: shortest path between nodes ( $x$  for start node,  $y$  for end node)
Initialization
(1) for all  $y$  in network
(2)    $dx(y) = c(x, y)$  /*if  $y$  is not a neighbor, then  $c(x, y) = \infty$ 
(3) for each neighbor  $w$ :
(4)    $dw(y) = \infty$  for  $dw(y) = \infty$ 
(5) for each neighbor  $w$ :
(6)   Send distance vector  $dx = [dx(y): y \text{ in } N]$  to  $w$ 
Loop:
(7) wait until a link cost change or receive a distance vector from neighbor
(8) for each  $y$  in network:
(9)    $dx(y) = \min_v \{c(x, y) + dv(y)\}$ 
(10) if  $dx(y)$  changed for any  $y$ :
(11)   Send distance vector  $dx = [dx(y): y \text{ in } N]$  to all neighbors
Forever

```

ALGORITHM 1: Pseudo code of Bellman-Ford algorithm

length. Step 2: find a node in the open set whose $w + h$ value is the smallest, and the node enters the closed set from the open set. Step 3: if $S_n = n_d$, the target node has been found and the algorithm ends. Step 4: no: go back to Step 2 to perform another calculation, so that the target node appears in the closed set, the search is completed, and the shortest path is obtained [19]. The flow chart is shown in Figure 3.

If $h(n)$ is always lower than the shortest distance value of n nodes to the target point, A^* can find the shortest path. Otherwise, the optimal path cannot be found.

This is proved by the backward method [20], taking the second-to-last node n as an example. Let the starting point be s , the target point be t , the next node selection of node n is j , q , the next node of node j , q is t , and the q node is in the optimal path. Among them, the optimal value of the link length of a and b from a to b is denoted by p_{ad} , and the evaluation value of a to b is denoted by hab . Node q is in the optimal path, so we have

$$g_{sn} + p_{nj} + p_{jt} > g_{sn} + p_{nq} + p_{qt}. \quad (18)$$

At n node selection, $h_{qt} > h_{jt}$ results in

$$g_{sn} + p_{nj} + p_{jt} < g_{sn} + p_{nq} + p_{qt}. \quad (19)$$

The current node selects the next search node j and $h_{qt} \leq p_{qt}$:

$$g_{sn} + p_{nj} + p_{jt} + 0 > g_{sn} + p_{nq} + p_{qt} \geq g_{sn} + p_{nq} + h_{qt}. \quad (20)$$

Therefore, the q node will be found again. By analogy, it can be proved that the path that can be finally found must be the shortest path. After finding the shortest path, the distance can be calculated in three ways [21].

(i) Harmanton distance:

$$h(x) = abs(x_2 - x_1) + abs(y_2 - y_1). \quad (21)$$

(ii) Chebyshev distance:

$$h(x) = \max\{abs(x_2 - x_1), abs(y_2 - y_1)\}. \quad (22)$$

(iii) Euclidean distance:

$$h(x) = \text{sqrt}((x_2 - x_3)^2 + (y_2 - y_1)^2). \quad (23)$$

(iv) Cosine distance:

$$\cos \theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + x_2^2} \sqrt{y_1^2 + y_2^2}}. \quad (24)$$

4. Analysis and Experiment of Road Network Topology

To find the shortest path, we must first analyze the road network structure. Table 1 takes the historical old city as an example and takes Prada [22], Gulou District, and the old city as samples to analyze the integration degree of the road network topology.

The analysis of these cities allows us to avoid cities with complicated road conditions and choose the optimal route when choosing a destination city for a self-driving tour. Prada, Gulou District, and the old city have different degrees of integration and understanding, but the difference is not big. When choosing a purpose, you can choose whatever you want.

It can be seen intuitively from Figure 4 that Prada, Gulou District, and the old city have different degrees of integration and understanding. The overall gap is not very big, and these cities can be used as destinations.

Table 2 shows the analysis of the peripheral area and the expansion area. Taking Jianye District, Glasgow, Bayswater, and the Outer Qinhuai River as samples [23], these areas are integrated and analyzed.

The road network analysis of the expansion area of the peripheral area is also to avoid the scenic spots with complicated road conditions and to help self-driving tourists to

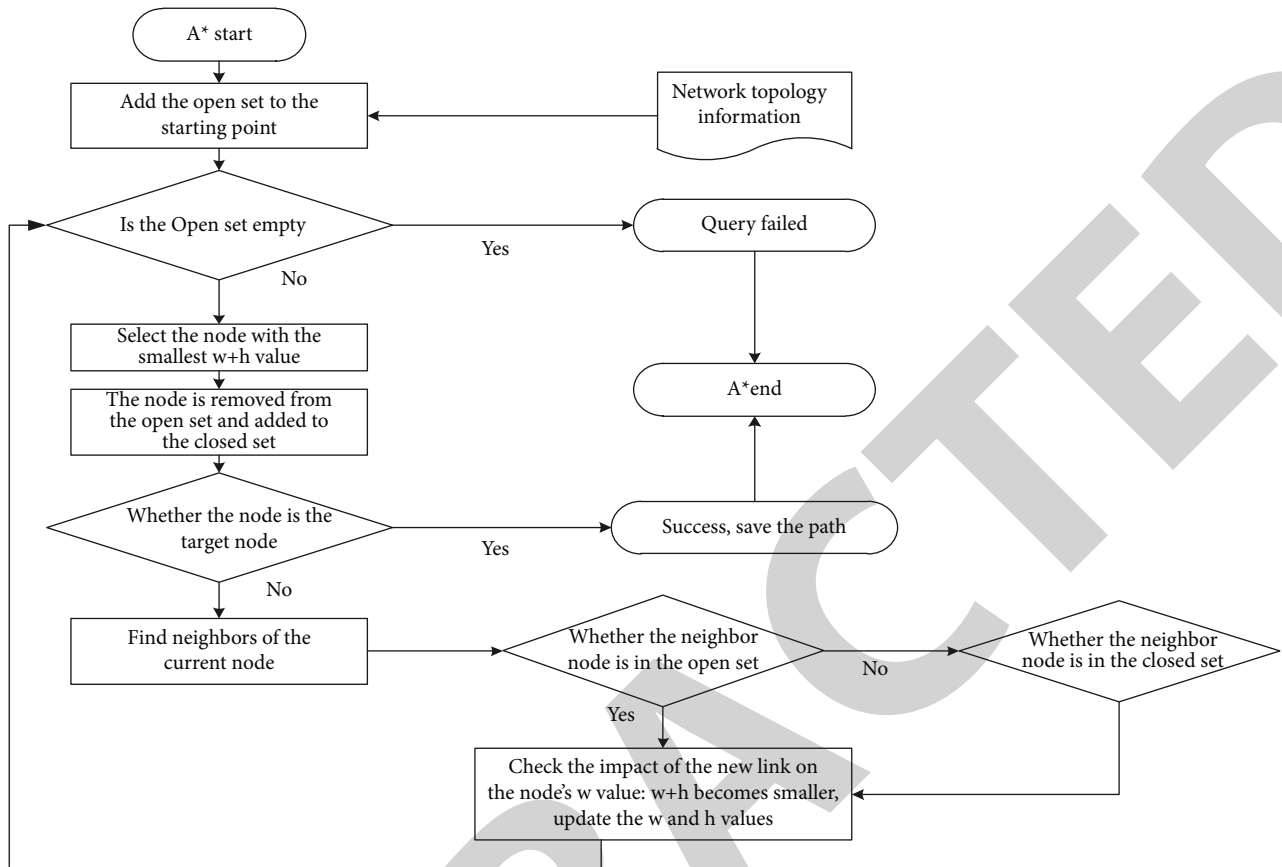


FIGURE 3: A* algorithm flow chart.

TABLE 1: Analysis of the road network structure in the historic old city.

Road network area	Road network sample	Global integration	Local integration	Intelligibility
Historic old town	Prada (Athens)	1.34	1.37	0.81
	Gulou District (Kaifeng)	1.68	1.91	0.94
	Old town (Suzhou)	1.78	2.03	0.95

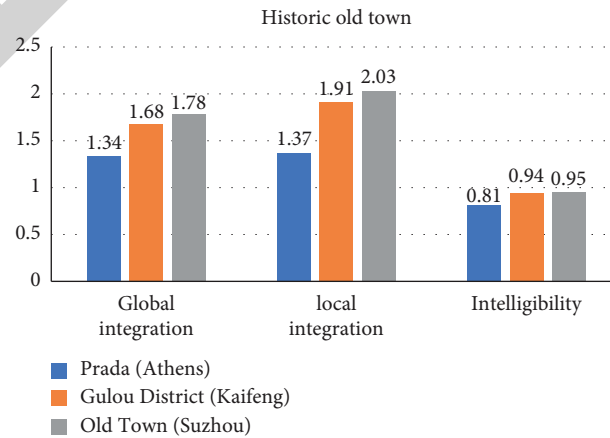


FIGURE 4: Analysis of the road network structure in the historic old city.

TABLE 2: Road network analysis of the expansion area and the peripheral area.

Road network area	Road network sample	Global integration	Local integration	Intelligibility
Expansion and outer zones	Jianye District (Nanjing)	2.51	2.55	0.99
	Glasgow (Scotland)	2.28	2.04	0.96
	Bayswater (London)	1.299	1.64	0.87
	Outer Qinhuai River (Nanjing)	0.73	1.29	0.64

choose. From the above table, it can be seen that the integration degree of Jianye District, Glasgow, Bayswater, and Outer Qinhuai River is quite different. Careful analysis is required when choosing a travel destination.

Figure 5 is an intuitive representation of the road network analysis of the expansion area of the peripheral area. The integration degree of Jianye District and Glasgow is much more understandable than that of Bayswater and the Outer Qinhuai River. Tourists choose the destination is best to choose the first two cities. The latter two are prone to traffic jams.

Table 3 is a comparison table of the historical old city and the expansion area and the outer area.

The comparison shows that the peripheral area and the expansion area are generally more integrated than the old city, which shows that the old city often has complex road conditions, while the peripheral area and the expansion area are relatively better and more suitable for self-driving tourists.

It can be seen intuitively from the statistical chart in Figure 6 that Jianye District has the greatest degree of integration and intelligibility, and the integration degree of several cities in the expansion area and peripheral area is higher than that of several cities in the historic old city. If the degree is larger, there will be intuitive reference data for the selection of tourist destinations for tourists.

4.1. Comparing the Performance of Different Algorithms. The following table is a comparison of the running time of different algorithms.

As can be seen from Table 4, with the increase of traffic volume, the running time of A* algorithm is shorter than that of Dijkstra algorithm and Bellman-Ford algorithm [24], which indicates that the more traffic volume, the better performance of A* algorithm. The Dijkstra algorithm and the Bellman-Ford algorithm take more time as the traffic increases.

From the broken line above in Figure 7, it can be intuitively shown that the running time of the A* algorithm has no obvious upward trend with the increase of the business volume, while the Dijkstra algorithm and the Bellman-Ford algorithm have an obvious upward trend. The shorter the search time, the better the search efficiency. The efficiency of the A* algorithm is better than that of the Dijkstra algorithm and the Bellman-Ford algorithm.

The following table is a comparison of the search space of different algorithms.

It can be seen from Table 5 that the search space of Dijkstra algorithm is generally larger than that of Bellman-Ford algorithm, and the search space of Bellman-Ford

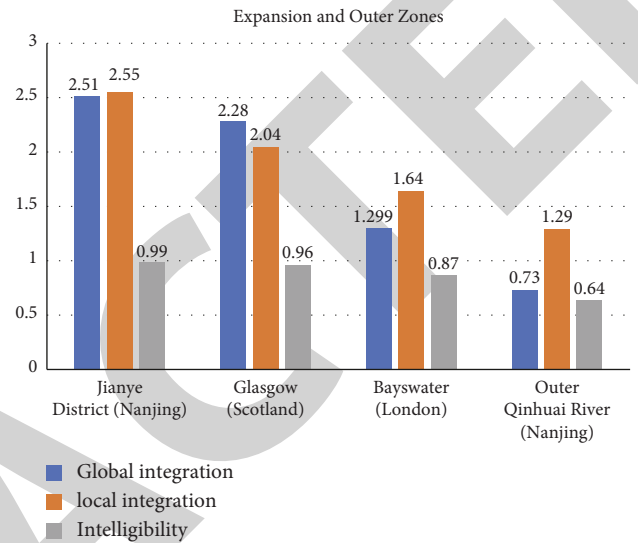


FIGURE 5: Road network analysis of the expansion area and the peripheral area.

algorithm is much larger than that of A* algorithm. The search space of Dijkstra algorithm and Bellman-Ford algorithm is too large, which will lead to inaccurate finding of the optimal path. The search space of A* algorithm is small, so the shortest path is more accurate.

Figure 8 more intuitively shows the search space size of Dijkstra algorithm, Bellman-Ford algorithm, and A* algorithm. It can be seen at a glance that A* algorithm is better than Dijkstra algorithm and Bellman-Ford algorithm. Because the search space of the A* algorithm is small, the search results are accurate, while other search spaces are too broad, and the results are not accurate enough.

4.2. Finding the Shortest Path. Because the A* algorithm is better than the Dijkstra algorithm and the Bellman-Ford algorithm, the A* algorithm is used to find the shortest path between the starting point A and the target point B [25]. The statistics are shown in Table 6.

The weight is the analysis of the road network structure, and the distance is the intuitive expression of finding the shortest path. The weight of path 6 is larger and the distance is the smallest. The other paths are either too small in weight or too far away, which are obviously not suitable for self-driving travel.

From Figure 9, it can be concluded that path 6 is the optimal path, which occupies a high weight and has the shortest distance. Other paths have different problems: either the path is long or the weight is small.

TABLE 3: Comparison table.

Road network area	Road network sample	Global integration	Local integration	Intelligibility
Historic old town	Prada (Athens)	1.34	1.37	0.81
	Gulou District (Kaifeng)	1.68	1.91	0.94
	Old Town (Suzhou)	1.78	2.03	0.95
Expansion and outer zones	Jianye District (Nanjing)	2.51	2.55	0.99
	Glasgow (Scotland)	2.28	2.04	0.96
	Bayswater (London)	1.29	1.64	0.87
	Outer Qinhuai River (Nanjing)	0.73	1.29	0.64

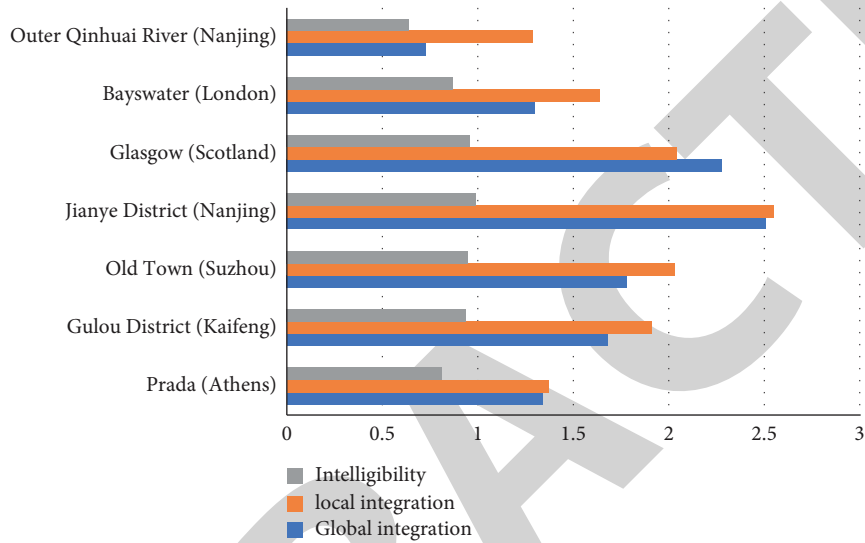


FIGURE 6: Comparison chart.

TABLE 4: Running time of different algorithms.

Business volume	Dijkstra's algorithm	Bellman-Ford algorithm	A* algorithm
	Running time (ms)	Running time (ms)	Running time (ms)
100	129	82	1741
300	342	265	1803
600	545	408	1769
900	1032	603	1796
1200	1360	1206	1864
1500	1707	1532	1872
1800	2013	1857	1900
2100	2365	2043	1934
2400	2681	2465	1989
2700	2964	2689	2041
3000	3335	3013	2103

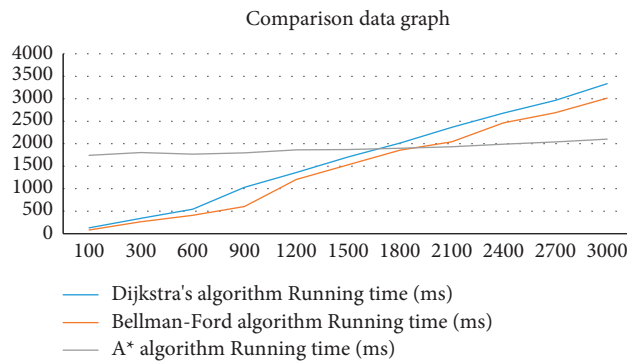


FIGURE 7: Running time of different algorithms.

TABLE 5: Search spaces of different algorithms.

	Dijkstra's algorithm	Bellman-Ford algorithm	A* algorithm
Business volume	Search space	Search space	Search space
100	13.84	11.21	5.43
300	12.75	10.32	5.68
600	13.21	11.32	6.96
900	13.95	10.35	4.83
1200	12.62	11.33	5.32
1500	13.78	11.47	5.23
1800	13.85	11.43	4.84
2100	12.96	11.68	6.32
2400	13.54	10.96	5.67
2700	13.47	11.54	6.23
3000	13.78	11.67	5.93

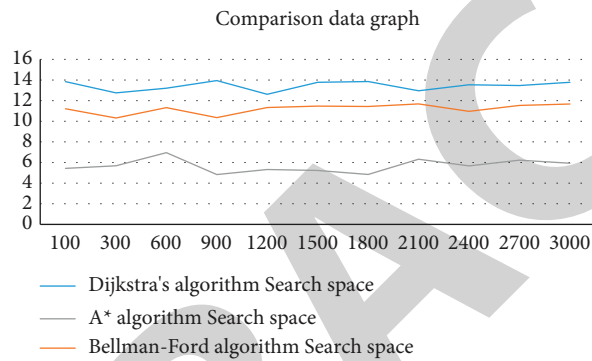


FIGURE 8: Search space comparison of different algorithms.

TABLE 6: Distance and weight of different paths.

	Attraction weight	Journey
Path one	4	6.5
Path two	3	5.5
Path three	4	4.3
Path four	3	5.7
Path five	6	6.2
Path six	6	4.1

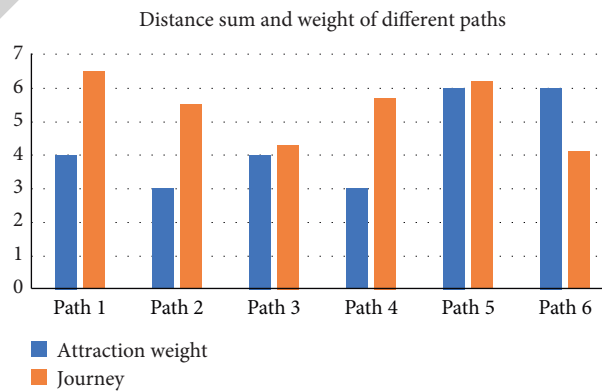


FIGURE 9: Distance and weight of different paths.

5. Conclusion

Recalling the preface, this study selects two indicators, comprehensibility and integration, to analyze the structure of the road network, then analyzes the accessibility to the destination, and calculates the relevant factors that affect self-driving travel. For the calculation of the optimal path, methods such as A* algorithm, Dijkstra algorithm, Bellman-Ford algorithm, and reverse order recursion model are introduced. The A* algorithm is an optimization of Dijkstra algorithm, Bellman-Ford algorithm, and other methods. The degree of integration and intelligibility of some road network topology structures are obtained through experiments. In the comparative experiments, the A* algorithm shows obvious advantages in calculating the shortest path. Finally, the A* method is used to find the shortest path from the starting point A to the target point B. It is hoped that research can become more rational and make people's travel life more convenient. With the in-depth research to develop good software, visitors only need to enter their destination, origin, vehicle model, budget, and travel time. You can quickly plan the best travel itinerary. If the information entered by the passenger is unreasonable, or the passenger's destination conflicts with various problems in the route, the software will remind the passenger to change the destination and will also give the passenger relevant recommendations.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declared that they have no conflicts of interest regarding this work.

Acknowledgments

This study was supported by the grant from The Sichuan-Tibet Synergism and Innovation Center of Tourism Industry Competitiveness (no. 21CZZX06) and fund project: Soft Science Research Project of Science and Technology Department in Sichuan Province "Study on the Income Effect of Tourism Poverty Alleviation and Prevention Mechanism of "Returning To Poverty" in Tibetan Areas of Sichuan" (Item no. 2021JDR0178).

References

- [1] C. Liu, R. Yu, and J. Zeng, "Complexity of spatial structure on the urban-rural road network in Wuhan Metropolitan Area," *Scientia Geographica Sinica*, vol. 32, no. 4, pp. 426–433, 2012.
- [2] M. Wei, Y. Huang, D. Wan, and L. K. Deng, "Emergency road network structure and planning optimization in mountainous regions in Southwest China under earthquake scenarios," *Journal of Mountain Science*, vol. 19, no. 3, pp. 771–780, 2022.
- [3] Kocur-Bera and Katarzyna, "Scale-free network theory in studying the structure of the road network in Poland," *Promet-traffic & transportation: Scientific journal on traffic and transportation research*, vol. 26, no. 3, pp. 235–242, 2014.
- [4] C. Araujo, E. Kowler, and M. Pavel, "Eye movements during visual search: the costs of choosing the optimal path," *Vision Research*, vol. 41, no. 25-26, pp. 3613–3625, 2001.
- [5] M. Sharifzadeh, M. Kolahdouzan, and C. Shahabi, "The optimal sequenced route query," *The VLDB Journal*, vol. 17, no. 4, pp. 765–787, 2008.
- [6] I. B. Schwartz, E. Forgoston, S. Bianco, and L. B. Shaw, "Converging towards the optimal path to extinction," *Journal of The Royal Society Interface*, vol. 8, no. 65, pp. 1699–1707, 2011.
- [7] A. Monokrousos, A. Bottaro, L. Brandt, A. Di Vita, and D. S. Henningson, "Nonequilibrium thermodynamics and the optimal path to turbulence in shear flows," *Physical Review Letters*, vol. 106, no. 13, Article ID 134502, 2011.
- [8] S. J. Weber, A. Chantasri, J. Dressel, A. N. Jordan, K. W. Murch, and I. Siddiqi, "Mapping the optimal route between two quantum states," *Nature*, vol. 511, no. 7511, pp. 570–573, 2014.
- [9] W. Tanaka, N. Yamanaka, M. Onishi, M. Ko, J. Yamanaka, and E. Okamoto, "Optimal route of administration of mixed endothelin receptor antagonist (TAK-044) in liver transplantation," *Journal of Gastroenterology*, vol. 35, no. 2, pp. 120–126, 2000.
- [10] Q. Xu, K. Yang, S. Peng, and L. Hong, "A distance-adaptive refueling recommendation algorithm for self-driving travel," *ISPRS International Journal of Geo-Information*, vol. 7, no. 3, p. 94, 2018.
- [11] K. Wang and G. Akar, "Effects of neighborhood environments on perceived risk of self-driving: evidence from the 2015 and 2017 Puget Sound Travel Surveys," *Transportation*, vol. 46, no. 6, pp. 2117–2136, 2019.
- [12] Q. C. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1533–1540, 2014.
- [13] C. Hallam, K. J. Harrison, and J. A. Ward, "A multiobjective optimal path algorithm," *Digital Signal Processing*, vol. 11, no. 2, pp. 133–143, 2001.
- [14] H. Zhang and Z. Li, "Weighted ego network for forming hierarchical structure of road networks," *International Journal of Geographical Information Science*, vol. 25, no. 2, pp. 255–272, 2011.
- [15] A. D. Barbour and G. Reinert, "The shortest distance in random multi-type intersection graphs," *Random Structures and Algorithms*, vol. 39, no. 2, pp. 179–209, 2011.
- [16] Y. Liu, "Research on optimal path planning for self-driving tour based on improved Dijkstra algorithm," *Heilongjiang Science and Technology Information*, vol. 000, no. 017, pp. 75–77, 2020.
- [17] H. Yan and Y. Liu, "Discussion on the shortest path algorithm of urban road network based on GIS," *Journal of Computers*, vol. 23, no. 2, p. 6, 2000.
- [18] F. Lu, "The shortest path algorithm: classification system and research progress," *Journal of Surveying and Mapping*, vol. 30, no. 3, p. 7, 2001.
- [19] Z. Jia, H. Jing, and W. Li, "Design of self-driving tour route based on GIS and traveling salesman problem," *Surveying and Mapping and Spatial Geographic Information*, vol. 37, no. 9, p. 4, 2014.
- [20] L. ianxiong and Y. Wang, "Innovation of the tourism industry in the Greater Bay Area based on the "double innovation" system of teacher-student collaboration--Taking the construction of the optimization platform for self-driving tour

- routes in Mijing as an example,” *Journal of Guangdong Agricultural and Commercial Vocational and Technical College*, vol. 38, no. 1, p. 5, 2022.
- [21] B. Lu, J. Yang, and X. Wang, “Urban suburban tourism route planning based on tourist experience,” *Journal of Chongqing Jianshu University: Natural Science Edition*, vol. 40, no. 10, p. 10, 2021.
- [22] X. Yan, R. Liu, and G. Xia, “Application of ant colony algorithm in self-driving tour route problem in chao hu lake area,” *Journal of Chaohu University*, vol. 18, no. 3, p. 8, 2016.
- [23] M. Song and Y. Wu, “Research on self-driving tour route planning based on adaptive ant colony algorithm for distribution uniformity,” *New Generation Information Technology*, vol. 3, no. 6, p. 8, 2020.
- [24] L. Ma and X. Cao, “A heuristic A* algorithm and a spatial accessibility calculation method for grid division,” *Geographical Research*, vol. 27, no. 1, p. 8, 2008.
- [25] H. Chen and Q. Zhang, “Application and implementation of A* algorithm in game map pathfinding,” *Computer Application and Software*, vol. 22, no. 12, p. 3, 2005.