Shortest Path Algorithm in Dynamic Restricted Area Based on Unidirectional Road Network Model

Haitao Wei, Shusheng Zhang and Xiaohui He *

School of Earth Science and Technology, Zhengzhou University, No. 75 Daxue North Road, Erqi District, Zhengzhou 450052, China; zuu_wei@zzu.edu.cn (H.W.); zhangssgiser@gs.zzu.edu.cn (S.Z.)
* Correspondence: hexh@zzu.edu.cn

Abstract: Accurate and fast path calculation is essential for applications such as vehicle navigation systems and transportation network routing. Although many shortest path algorithms for restricted search areas have been developed in the past ten years to speed up the efficiency of path query, the performance including the practicability still needs to be improved. To settle this problem, this paper proposes a new method of calculating statistical parameters based on a unidirectional road network model that is more in line with the real world and a path planning algorithm for dynamically restricted search areas that constructs virtual boundaries at a lower confidence level. We conducted a detailed experiment on the proposed algorithm with the real road network in Zhengzhou. As the experiment shows, compared with the existing algorithms, the proposed algorithm improves the search performance significantly in the condition of optimal path under the premise of ensuring the optimal path solution.

Keywords: navigation; unidirectional road network; statistical parameter; path planning; restrict search area; Dijkstra algorithm

1. Introduction

With the rapid development and widespread application of technologies such as mobile geographic information systems (GIS), global positioning system (GPS), wireless sensor networks (WSN) and wireless communications [1–6], many people rely on the path provided by the navigation device to travel [7–9]. However, as the car ownership increases, the low capacity of urban roads limits the efficiency of traffic [10,11]. Therefore, how to choose an optimal driving route in the complex urban road network has risen lots of attention [12].

In network theory, this corresponds to the shortest path problem. The shortest path problem is to find the shortest directed path from a starting node to others in a directional network of arbitrary length which is one of the most basic problems in network optimization [13,14]. There are many algorithms for the shortest path including Dijkstra algorithm [15], A* algorithm [16], ant colony algorithm [17], genetic algorithm [18] and simulated annealing algorithm [19]. Dijkstra algorithm is the most commonly used algorithm to solve this problem on graphs with edge weights that are not negative. The main idea of Dijkstra algorithm is to spread circle by circle with the center, the starting node O, until the terminal node D is surrounded. Although the Dijkstra algorithm can effectively search for the shortest path between Origin–Destination (OD) pairs, it only considers the topological characteristics of the network in the design process. Dijkstra algorithm ignores the spatial distribution characteristics of the network so that the path search is not directional, so a large number of nodes that are not related to the shortest path are introduced into the calculation, which affects the efficiency of the algorithm. Therefore, higher resource cost and lower search efficiency limits the application of traditional Dijkstra algorithm in navigation software. Through the analysis of the optimization path based on the Dijkstra shortest path algorithm, it is found that the previous researchers mainly optimized the performance of the shortest path algorithm from four aspects: (1) optimization based on
Various optimization algorithms are developed to reduce the number of nodes because the time complexity of Dijkstra algorithm is proportional to the square of the number of nodes. Assuming that the number of nodes is proportional to the area of the search area, reducing the area of the search area is essential to this problem. In the transportation network, the direction of the route will be reflected if the starting point and ending point are determined. According to this network characteristic, Nordbeck [21] proposed a shortest path algorithm based on the ellipse, which limits the search node collection in a certain area and greatly narrows the search scale. Lu [31] proposed a shortest path algorithm with the minimum bounding rectangle of the ellipse as the restricted area. The experimental results showed that although the algorithm expanded the search area compared to the shortest path algorithm based on the ellipse restriction, there was no need for a large number of product and square root operations; therefore, the algorithm improved search efficiency. Fu [32] proposed that the line of the OD pair was set as the diagonal of the rectangle firstly, then, the threshold $T_1$ was determined to expand the original rectangle and get the large one $R$, and finally the two sides of the diagonal of the rectangle $R$ were set as the search area. Experimental results showed that compared with Dijkstra algorithm, the search scale of this algorithm was significantly reduced, which effectively improved the execution speed of the algorithm. Wang [22] proposed a shortest path algorithm based on a parallelogram to limit the search area. The algorithm used the smallest parallelogram containing an ellipse as the restricted search area. The experimental results showed that the search time required by the algorithm was less than the unlimited search in all cases and ellipses limit the search time required to search the area. Bu [33] proposed an improved algorithm that dynamically changed the search direction and restricted the search area in the dynamic search area based on the spatial characteristics of the road network. Compared with Dijkstra algorithm, this algorithm reduced the time and space complexity of the algorithm and improved the efficiency of the algorithm. Wang [34] proposed an optimal path algorithm that limited the search area based on the common characteristics of typical urban road networks. According to the different Euclidean distances of OD pairs, the algorithm searched for the shortest path in two types of ellipses of different sizes. Experimental results showed that when the Euclidean distance of the OD pair is far, the algorithm could reduce the time consumption by 33–47% compared with the ellipse restricted search area algorithm. Zhou [35] proposed a shortest path algorithm based on the corner direction rectangle. The algorithm was based on the direction rectangle and reduced the search area by cutting the corresponding corner area. The experimental results showed that the algorithm could significantly improve the search efficiency of the shortest path without affecting the reliability.

The various shortest path algorithms for restricted search areas proposed by the above scholars fully consider the characteristic attributes of the spatial distribution of the road network, including the positional relationship and distance relationship between the starting node and the ending node. The above algorithms all improve the efficiency of searching for the shortest path by narrowing the search area and reduce data redundancy. However, there are still some limitations, which can be summarized as follows:

1. Most of the road network data models used in the previous or current restricted search area algorithms are two-way road network models. However, the two-way road network model does not well represent the actual road conditions in the real world, and the statistical parameter results of the two road network models are quite different;
2. Most restricted search area algorithms are improved on the basis of the ellipse restricted search area algorithm. Therefore, a 95% confidence level is adopted in the process of calculating statistical parameters to ensure that most of the path solutions obtained are optimal. However, a higher confidence level has the problem of a larger search area;
In response to the above problems, this paper proposes unidirectional road network model (URNM) and shortest path algorithm based on virtual boundary in dynamic ellipse restricted area (SPAVBDERA). The data selected in this paper is the road network data of Zhengzhou, which includes highways, national highways, provincial highways, urban expressways and main urban roads in Zhengzhou. URNM was established on this basis. URNM regards the road as the basic element, uses the relationship between the roads, starts from the road network information and the path search efficiency, and establishes a road network model that meets the actual road needs. We have proved its practicality in the algorithm. Firstly, SPAVBDERA calculates different statistical parameters according to the Euclidean distance of different OD pairs, and selects a smaller confidence level to establish the algorithm search area, and then constructs a virtual boundary in the search area to ensure that the algorithm can obtain the complete solution of the optimal path. Finally, we compared the results of SPAVBDERA with Dijkstra algorithm and the shortest path algorithm for ellipse restricted search area.

The paper is organized as follows: In the Section 2, URNM and SPAVBDERA are elaborated. The Section 3 shows the experimental results of the proposed algorithm. The results of SPAVBDERA are further discussed in Section 4. The full text is summarized and the future research directions are prospected.

2. Algorithm Description

2.1. Unidirectional Road Network Model (URNM)

URNM is improved based on the graph structure model. The modeling idea of URNM regards the road as the basic element to build the model. URNM saves the attribute information and road traffic information of the road section on the road, which reduces data storage redundancy and improves the efficiency of querying the shortest path of the road network.

\[ G = <V, E, E_{\text{attr}} > \]  \hspace{1cm} (1)

In the Equation (1), \( G \) represents the road network; \( V = (v_0, ..., v_n) \) represents the set of nodes in the network, \( v_i \) is the intersection or endpoint of the road; \( E = <v_i, v_j> \) is the set of ordered vertex pairs, which indicates that there is one path from \( v_i \) to \( v_j \), \( v_i, v_j \in V \), the direction of each path is a vector; \( E_{\text{attr}} = \{E_{\text{ID}}, E_{\text{LongS}}^i, E_{\text{LatiS}}^i, E_{\text{LongE}}^i, E_{\text{LatiE}}^i, E_{\text{Length}}^i, E_{\text{Direction}}^i\} \) is a collection of path attributes, \( E_i = <v_i, v_{i+1}> \), \( E_i \in E \), \( E_{\text{ID}}^i \) represents the road to which the path belongs, and paths on the same road have the same ID; \( E_{\text{LongS}}^i \) represents the geographic abscissa of the path starting point; \( E_{\text{LatiS}}^i \) represents the geographic ordinate of the path starting point; \( E_{\text{LongE}}^i \) represents the geographic abscissa of the end of the path; \( E_{\text{LatiE}}^i \) represents the geographic ordinate of the end of the path; \( E_{\text{Length}}^i \) represents the length of the path, where the path length discussed in this paper is a scalar, non-negative number, and all lengths are positive; \( E_{\text{Direction}}^i \) is the path direction indicator, 1 means positive, 2 represents the reverse direction, and 3 represents the virtual boundary direction.

Compared with the ordinary plane network diagram, the road network diagram based on URNM usually has the following characteristics:

1. There are many points and edges, and the number of road sections connected to each node is mostly 2, 3, or 4, which is mostly a large-scale sparse network;
2. The network topology is complicated;
3. The side is one-way accessible;
4. There is a special reversing node in the network that indicates the vehicle turns around.

2.2. Dynamically Restricted Search Area Path Planning Algorithm

The classic Dijkstra [15] algorithm adopts a greedy strategy. This algorithm is a process of generating the shortest path in the order of increasing distance from the node to the starting node, which has great blindness and is always ready to go everywhere unfold. In this way, the search area finally swept is a circle with the starting point as the origin.
and the length of the line between the starting point and the terminal point as the radius. The time complexity of this algorithm is $O(n^2)$, where $n$ is the number of nodes in the road network. Therefore, the algorithm will generate a large number of redundant nodes in the process of traversing the nodes in the search area. This shortcoming promotes the research of related algorithms for reducing the search area. The traditional shortest path algorithm for restricted search area is proposed for the spatial distribution characteristics of road networks. Its purpose is to determine a small range in a large road network so that the shortest path between any given two nodes is most likely to be searched in this range. These algorithms improve the efficiency of searching for the shortest path and reduce data redundancy.

2.2.1. Shortest Path Algorithm Based on Ellipse Restricted Search Area (SPAERSA)

SPAERSA is the shortest path optimization algorithm first proposed by [21] to narrow the search range. The algorithm defines an ellipse as a restricted search area, which means that nodes outside the ellipse are ignored during the search for the shortest path, thereby improving the efficiency of the shortest path search. The algorithm uses the starting point $O$ and the terminal point $D$ as the focus of the ellipse to construct the ellipse. The key to the ellipse limitation in the algorithm is to find the appropriate major axis $MD$.

The length of $MD$ is calculated by statistical analysis. First, we systematically extract a certain number of representative nodes from the set of transportation network nodes, and construct two sets of nodes $G_O$ and $G_D$. The Cartesian product $G_{OD}$ of these two sets can be given by Equation (2).

$$G_{OD} = G_O \times G_D = \{(O, D) | (O \in G_O) \land (D \in G_D)\}$$

(2)

Each element in $G_{OD}$ can be regarded as the starting and terminal point of the shortest path to be found. Suppose the actual distance of a certain OD pair is $F_{OD}$, and the Euclidean distance is $E_{OD}$, so the ratio coefficient $R_{OD} = F_{OD}/E_{OD}$ can be obtained. For sample, the ratio coefficient set $R_{OD}$ can be obtained. Then we perform statistical analysis on the elements in the $R_{OD}$ to obtain a specific value $f$ so that the total number in the $R_{OD}$ is the elements that meet a certain confidence level, and this value is less than $f$. Finally, we use $f$ as a constant multiplier and use the coordinates of the starting and terminal points to obtain the $MD$ of the major axis of the ellipse. The specific process of the shortest path algorithm based on the ellipse restricted search area is shown in Algorithm 1. $\alpha$ is the Euclidean distance between two points, OPEN represents the temporary label set of nodes, and CLOSE represents the permanent label set of nodes. $\Gamma$ represents the set of nodes in the restricted search area, CurrentNode represents the current node, AdjacentNode represents the adjacent node of CurrentNode, and S represents the shortest path.

The statistical parameter $f$ calculated by SPAERSA is generally a constant multiplier that makes the total number of elements in the $R_{OD}$ meet the 95% confidence level. The algorithm takes into account the position relationship between the starting node and the terminal node and limits the search node set to an ellipse, which reduces the number of nodes that needs to be traversed during the search process, thereby achieving the goal of reducing time complexity. However, the higher confidence level makes the algorithm still need to traverse more nodes in the process of searching for the shortest path. Especially when the Euclidean distance of the OD pair is large, the algorithm will still generate a large number of redundant nodes, so the efficiency of the algorithm is sometimes even not as good as the omnidirectional search algorithm of unlimited search area.
2.2.2. Fitting of Statistical Parameter Functions

Aiming at the problem of low search efficiency in SPAERSA when the Euclidean distance of the OD pair is large, this paper analyzes $R_{OD}$ and $E_{OD}$ data. Take Zhengzhou as an example. First, 400 nodes are systematically extracted from the transportation network of Zhengzhou to construct the set $G_{OD}$, so the $G_{OD}$ includes 40,000 sample elements. We solve $F_{OD}^{i}, E_{OD}^{i}$ and $R_{OD}^{i}$ separately for each element in $G_{OD}$. If the aggregate $R_{OD}$ is counted, we can get that the value of the statistical parameter $f$ is 1.611 at the 95% confidence level. Figure 1 shows the numerical distribution of $R_{OD}$ and $E_{OD}$.

![Figure 1. Sample element distribution.](image-url)
It can be seen from Figure 1 that the linear distance between the statistical parameter \( f \) and the OD pair has a certain functional relationship. The smaller the Euclidean distance in the OD pair, the greater the probability that \( f \) will get a larger value. The greater the value of Euclidean distance in the OD pair, the greater the probability that the value of \( f \) tends to 1. Based on this phenomenon, we propose a new calculation method for statistical parameters.

Assuming that the statistical parameter function \( R \) satisfies the 95% confidence level, the specific calculation method is:

First, the 40,000 sample elements are sorted according to the Euclidean distance from small to large. We set the Euclidean distance as a group every 2 kilometers and assume that there are \( C_i \) elements in the \( i \)-th group. Then we sort the \( f \) values in each group from large to small and select the \( C_i \times 5\% \) number in each group to form a new set \( SE \). Finally, we perform curve fitting on the sample elements in the new set. We perform curve fitting on the sample elements in \( SE \) to obtain the curve Equation (3) in Figure 2.

\[
F = -2.065 \times x^{(0.06163)} + 5.406
\]  

(3)

Figure 2. Curve fitting of sample elements in \( SE \).

It can be seen from Figure 2 that the \( f \) value of the OD pair decreases with the increase of the Euclidean distance, which avoids the generation of a large number of redundant nodes in the process of traversing the nodes in the search area when the OD pair Euclidean distance is large. The statistical parameter function effectively solves the problem of low search efficiency of SPAERSA when the Euclidean distance of the OD pair is large. However, there are sample points above the curve, which means that a part of the points above the curve cannot obtain the complete solution \( S = (s_1, s_2, \ldots, s_n) \) of the optimal path in the search area.
2.2.3. The Shortest Path Algorithm Based on the Dynamic Ellipse Limit Search Area of the Virtual Boundary

Since the connection direction from the starting point O to the terminal point D basically represents the approximate direction of the shortest path when we plan the path for a given OD pair in the actual urban road network. This means that the final shortest path is basically on both sides of the two nodes, and usually near them. Therefore, OD pairs can get most of the solutions of the optimal path when the complete solution of the optimal path cannot be obtained in the appropriate search area.

Based on the above ideas, this paper proposes a shortest path algorithm based on virtual boundary in dynamic ellipse restricted search area, and the specific process is shown in Algorithm 2. $\gamma$ is a statistical parameter. $E_i$ represents an edge in graph $G$. $G'$ represents the edge in the restricted search area.

| Algorithm 2 The Shortest Path Algorithm based on Virtual Boundary in Dynamic Ellipse Restricted Area: FindPathSPAVBDERA(G, O, D) |
|---|
| **Input:** | Graph $G$, origin O, destination D. |
| **Output:** | The shortest path from O to D. |
| 1: | $\gamma \leftarrow -2.065 \times \sqrt{\alpha_{OD}, 0.06163} + 5.406$; |
| 2: | Set O and D as the focal points, $\alpha_{OD}$ as the focal length, and $\gamma\alpha_{OD}$ as the major axis length to establish the ellipse equation $P$; |
| 3: | while $\exists E_i \in G$ and $E_i$.traverse = false do |
| 4: | if $E_i$.origin $\in P$ and $E_i$.destination $\in P$ then |
| 5: | $\Omega \leftarrow 1$, $G'_i \leftarrow E_i$, $E_i$.traverse = true; |
| 6: | else if $E_i$.origin $\notin P$ and $E_i$.destination $\notin P$ then |
| 7: | $\Omega \leftarrow 2$, $E_i$.traverse = true; |
| 8: | else if $E_i$.origin $\in P$ and $E_i$.destination $\notin P$ then |
| 9: | $\Omega \leftarrow 3$, $\Psi \leftarrow E_i$, $G'_i \leftarrow E_i$, $E_i$.traverse = true; |
| 10: | else |
| 11: | $\Omega \leftarrow 4$, $\Psi \leftarrow E_i$, $G'_i \leftarrow E_i$, $E_i$.traverse = true; |
| 12: | Take the midpoint of the OD pair as the origin, and arrange the elements in $\Psi$ counterclockwise; |
| 13: | while $\exists E_j \in \Psi$ and $E_j$.traverse = true do |
| 14: | Connect the nodes of the edges away from the center of the ellipse in order; |
| 15: | if $\Omega = 3$ then |
| 16: | $E_i^{\text{Length}} \leftarrow a_{E_i,E_{i+1}}$, $E_i^{\text{Direction}} \leftarrow 3$; |
| 17: | else |
| 18: | $E_i^{\text{Length}} \leftarrow \beta_{E_i,E_{i+1}}$, $E_i^{\text{Direction}} \leftarrow 3$; |
| 19: | $S' \leftarrow$ the shortest path in $G'$; |
| 20: | vs. $\leftarrow$ Set of $n$ consecutive virtual solutions $VS_n$ in $S$; |
| 21: | if vs. = null then |
| 22: | return $S'$; |
| 23: | else |
| 24: | while $\exists VS_j \in$ vs. do |
| 25: | $Q_j \leftarrow$ Find the shortest path solution based on $VS_j$.origin and $VS_j$.destination in $G'$ in $G$-$G'$; |
| 26: | $S \leftarrow (S' \cup \text{vs.}) \cup Q_j$ |
| 27: | return $S$; |

The Algorithm 2 first constructs a virtual boundary as a virtual solution $VS = (vs_1, \ldots, vs_j)$ to replace the solution outside the ellipse area to obtain a virtual complete solution $S'$.
Sensors 2021, 21, 203

= \left( s_1, s_2, \ldots, s_i, \ldots, s_j, \ldots, s_n \right) in the ellipse area when the OD pair cannot obtain the complete solution \( S \) of the optimal path in the ellipse search area. Then find the optimal complete solution \( OS = \left( s_i, \ldots, s_j \right) \) outside the ellipse for each continuous virtual solution according to the starting point and ending point. Finally, replace the virtual solution in \( S' \) with the solution in \( OS \) to get the complete solution \( S = \left( s_1, s_2, \ldots, s_i, \ldots, s_j, \ldots, s_n \right) \).

This algorithm can effectively solve the problem that some OD pairs cannot get the optimal path solution due to the small \( f \) value. The weight value \( E_{Length}^i \) of the virtual edge is divided into different assignment methods according to the value of \( E_{ID}^i \) in adjacent nodes. If the \( E_{ID}^i \) of two adjacent nodes are the same, the value of \( E_{Length}^i \) is the Euclidean distance between the two points. Otherwise, the value of \( E_{Length}^i \) is the product of the Manhattan distance between the two points and the statistical parameter.

2.3. Algorithm Performance Analysis

2.3.1. Algorithm Reliability Analysis

Aiming at the problem that the higher confidence level makes the calculated statistical parameter function value too large, which in turn makes the algorithm search area too large. This paper proposes to choose the statistical parameter function derived from the low confidence level combined with SPAVBDERA to improve the shortest path algorithm of the ellipse restricted search area.

First, we calculate the statistical parameter function under different confidence levels. We select the six confidence levels of 95%, 90%, 85%, 80%, 75%, and 70%, and calculate the statistical parameter functions under these six confidence levels, as shown in Table 1.

| Confidence Level | Statistical Parameter Function |
|------------------|--------------------------------|
| 95%              | \( F = -2.065 \times x^{(0.06163)} + 5.406 \) |
| 90%              | \( F = 4.997 \times x^{-0.08526} - 0.6799 \) |
| 85%              | \( F = -0.574 \times x^{(0.09626)} + 2.906 \) |
| 80%              | \( F = -0.08607 \times x^{(0.1943)} + 1.962 \) |
| 75%              | \( F = -0.00278 \times x^{(0.4298)} + 1.534 \) |
| 70%              | \( F = -0.001674 \times x^{(0.4611)} + 1.471 \) |

Then 100 sample elements are randomly selected from the road network to calculate the optimal path solution of these 100 sample elements under six confidence levels. Because the Dijkstra algorithm can search for the global optimal solution of the sample elements, the optimal path solution obtained under different confidence levels is compared with the optimal path solution obtained by Dijkstra algorithm to determine whether the optimal path solution obtained under different confidence levels is the global optimal solution, the experimental results are shown in Figure 3.

Finally, the analysis of the sample results shows that SPAVBDERA shows good results at six different confidence levels. Especially SPAVBDERA under the three confidence levels of 95%, 90%, and 85% obtains the global optimal solution in 100 sample elements. Although there are a few sample elements in the remaining three confidence levels that have not obtained the global optimal solution, they have obtained the local optimal solution. It is optimal and reliable to select the 85% confidence level comprehensively considering the accuracy and time complexity of the algorithm.
2.3.2. Algorithm Validity Analysis

If it is assumed that the nodes in the road network are uniformly distributed in the entire road network plane (that is, the number of nodes is proportional to the area occupied by the area, even if the distribution of local sites is uneven). The actual number of nodes that need to be visited in the search process can be expressed by the area \(A\) of the searched area, and the time complexity of the search can also be represented as \(O(A^2)\). Assuming that the time complexity of the shortest path algorithm in the search area restricted by the constant multiplier at the 95% confidence level is expressed as \(O(A^2)\), the time complexity of SPAVBDERA with the statistical parameter function curve at the 95% confidence level is expressed as \(O(A_{95\%}^2)\), the time complexity of SPAVBDERA of the statistical parameter function curve at the 85% confidence level is expressed as \(O(A_{85\%}^2)\). The three statistical parameter function curves are shown in Figure 4.

It can be seen from Figure 4 that the value of the constant multiplier \(f\) is 1.611, the value range of the statistical parameter \(f_{95\%}\) under the 95% confidence level is (1.1603, 2.2451), and the value range of the statistical parameter \(f_{85\%}\) under the 85% confidence level is (1.1366, 1.7899). Since the time complexity of the algorithm search is proportional to the area of the search area [36], it is easy to get Equation (4).

\[
\begin{align*}
\frac{O(A^2)}{O(A_{95\%}^2)} &= \left(\frac{\pi f_{95\%} E_{OD}}{2} \sqrt{\left(f_{95\%} E_{OD}^2 - E_{OD}^2\right)^2} \right)^2 = \frac{f^4 - f^2}{f_{95\%}^4 - f_{95\%}^2} \\
\end{align*}
\]

It can be seen from Equation (4) and Figure 4 that the time complexity of SPAVBDERA under the 95% confidence level is reduced more than SPAERSA when the Euclidean distance of the OD pair is larger. Similarly, the Equation (5) can be calculated.
It can be obtained that the time complexity of SPAVBDERA at 85% confidence level for most OD pairs can be reduced by 21–66% compared to SPAVBDERA at 95% confidence level from Equation (5).

\[
O\left(A_{95\%}^2\right) / O\left(A_{85\%}^2\right) = r_{95\%}^2 \left( r_{85\%} - 1 \right) / r_{85\%}^2 \left( r_{85\%} - 1 \right)
\]  

(5)

3. Results

This chapter mainly compares and evaluates the reliability and effectiveness of the three algorithms (SPAERSA, SPAVBDERA under 95% confidence level and SPAVBDERA under 85% confidence level) by measuring the running time of the algorithm under different Euclidean distances and the number of algorithm traversal nodes on the road network of Zhengzhou. The integrated development environment for running the three algorithms is Microsoft Visual Studio 2017, the development language is C++, the processor is Intel Core i5-4590 @ 3.30GHz quad-core, and the memory is 8GB, the computer operating system is Windows 10, and the database is PostgreSQL 10.0.

First, we combine the actual application of the path planning algorithm in the real-time vehicle navigation system, and apply the three algorithms to the vectorized Zhengzhou road network (the entire road network has 26,206 edges), as shown in Figure 5. Then we solve the shortest path given different starting points and ending points. Due to space limitations, only the starting node 3427 and the terminal node 1057 are given here. Finally, the search results of the three algorithms are obtained. Figure 6 are the search results of SPAERSA, Figure 7 are the search results of SPAVBDERA under the 95% confidence level, and Figure 8 are the search results of SPAVBDERA under the 85% confidence level. The black lines in the three figures indicate the edges of the road network, the blue highlight lines indicate the edges visited by the corresponding algorithm in the process of searching for the shortest path, and the yellow highlight lines indicate the shortest path finally searched results by the corresponding algorithm.

Comparing Figures 6–8, it can be seen that the search area scanned by the traditional SPAERSA is an ellipse with the starting node 3424 to the terminal node 1057 as the focal points and 1.611 times the Euclidean distance between the two points as the major axis to obtain the shortest path solution from the starting node 3427 to the terminal node 1057. Although the search area scanned by SPAVBDERA under the 95% confidence level and the
85% confidence level is still an ellipse, the area of the ellipse under these two confidence levels is significantly smaller than the search area scanned by the traditional SPAERSA. After calculation, the search area scanned by SPAVBDERA at the 85% confidence level is an ellipse with the starting node 3424 and the terminal node 1057 as the focal points and 1.324 times the Euclidean distance between the two points as the major axis.

Figure 5. Zhengzhou road network.

Figure 6. Search results of the shortest path algorithm in the traditional ellipse restricted search area.
Figure 6. Search results of the shortest path algorithm in the traditional ellipse restricted search area.

Figure 7. The search results of the shortest path algorithm based on the dynamic ellipse limit search area of the virtual boundary under the 95% confidence level.

Figure 8. The search results of the shortest path algorithm based on the dynamic ellipse limit search area of the virtual boundary under the 85% confidence level.

It can be seen from Figure 6 that the search area scanned by the traditional SPAERSA is able to find the solution of the shortest path, which means that SPAERSA with a lower confidence level cannot search the solution of the shortest path in the search area. It can be seen from Figure 8, however, SPAVBDERA still obtains the shortest path solution under the 85% confidence level. The reason for this phenomenon is that SPAVBDERA obtains the virtual shortest path solution by replacing the solution outside the search area with a
virtual solution, then searches for the shortest path solution in a small area. This is basically consistent with the previous theoretical analysis of the algorithm. The performance comparisons of the search results of the three algorithms are shown in Table 2, Figures 9 and 10, respectively.

Table 2 lists the shortest path solutions of the three restricted search area algorithms and Dijkstra algorithm under nine groups of OD pairs with different Euclidean distance, where \( S_{Dij} \), \( S_{SPA} \), \( S_{95\%} \), and \( S_{85\%} \) are represent the shortest path lengths obtained by the Dijkstra algorithm, SPAERSA, and SPAVBDERA, respectively, at 95% confidence level whereas the SPAVBDERA has a confidence level of 85%. In Table 2, the NULL means that the shortest path has not been searched. Figure 9 shows the number of nodes accessed by the three restricted search area algorithms under 9 sets of OD pairs with different Euclidean distances, excluding the number of repeated visits; Figure 10 shows the search time used by the three restricted search area algorithms under nine sets of OD pairs with different Euclidean distances.

**Table 2.** Comparison of the shortest path length between Dijkstra algorithm and the three restricted search area algorithms.

| Group | \( S_{Dij}/m \) | \( S_{SPA}/m \) | \( S_{95\%}/m \) | \( S_{85\%}/m \) |
|-------|----------------|----------------|----------------|----------------|
| 1     | 11,093.9       | 11,093.9       | 11,093.9       | 11,093.9       |
| 2     | 27,898.5       | 27,898.5       | 27,898.5       | 27,898.5       |
| 3     | 50,367.1       | NULL           | 50,367.1       | 50,367.1       |
| 4     | 75,030.9       | 75,030.9       | 75,030.9       | 75,030.9       |
| 5     | 52,795.6       | 52,795.6       | 52,795.6       | 52,795.6       |
| 6     | 49,771.5       | 49,771.5       | 49,771.5       | 49,771.5       |
| 7     | 84,104.7       | 84,104.7       | 84,104.7       | 84,104.7       |
| 8     | 99,393.8       | 99,393.8       | 99,393.8       | 99,393.8       |
| 9     | 100,826        | 100,826        | 100,826        | 100,826        |

Figure 9. The number of nodes traversed by the three algorithms under different Euclidean distances.
From Table 2, the optimal path solution obtained by SPAVBDERA proposed in this paper is always reliable, because the shortest path obtained by SPAVBDERA and Dijkstra algorithm is always consistent. The third set of the data shows that the shortest path solution cannot be obtained in some cases under SPAERSA. This is because of the small statistical parameters of SPAERSA at the 95% confidence level. Therefore, theoretically, the probability when the shortest path solution cannot be obtained is less than 5%, which is a small probability event. It is generally considered that the 5% can be ignored [37].

It can be seen that the number of nodes visited by the algorithm and the running time of the algorithm are directly related to the OD to the Euclidean distance from Figures 9 and 10. As can be predicted, the number of nodes visited by the algorithm and the running time of the algorithm both increase as the Euclidean distance of the OD pair increases. This is because that as the Euclidean distance of the OD pair increases, the search range of the algorithm increases, which increases the number of nodes visited by the algorithm as well as the running time of the algorithm.

Because the statistical parameters of SPAERSA are lower than those of SPAVBDERA under the two confidence levels when the Euclidean distance of the OD pair is small, the search time and the number of search nodes of SPAERSA are lower than that of SPAVBDERA. When the Euclidean distance of the OD pair is larger; however, SPAVBDERA under the two confidence levels shows better performance than SPAERSA.

It can be seen from the nine sets of the experiments that SPAVBDERA under the 85% confidence level is better than under the 95% confidence level in the number of nodes searched and the search time under the premise of ensuring the shortest path solution. Therefore, the application of SPAVBDERA under the 85% confidence level is reliable in this road network.

4. Discussion

With the continuous increase of car ownership, more and more vehicles use navigation systems to select routes. The shortest path algorithm is one of the most basic algorithms of the navigation system; therefore, the optimization of the shortest path algorithm is increasingly important.

Accordingly, we proposed a shortest path algorithm based on virtual boundary in dynamic ellipse restricted search to improve the traditional shortest path algorithm in re-
restricted search area. Firstly, we proposed a new calculation method of statistical parameter function based on a more realistic unidirectional road network model. Then, we compared the statistical parameter function at different confidence levels and found that the statistical parameter function at the 85% confidence level is optimal. Consequently we constructed a virtual boundary based on the statistical parameter function under the 85% confidence level, so as to achieve the shortest path solution between two points.

We compared the two groups of SPAVBDERA under different confidence levels and SPAERSA with real road network in Zhengzhou. The experimental results show that under the premise of ensuring the shortest path solution, SPAVBDERA can better adapt to the actual road network. In addition, when the Euclidean distance of the OD pair is slightly larger, the number of search nodes and the search time of the algorithm proposed by SPAVBDERA under the 95% confidence level are significantly reduced compared with SPAERSA, and the reliability of the algorithm is significantly improved as well. Then, we further compared SPAVBDERA under different confidence levels. The experimental results show that the selected SPAVBDERA under the 85% confidence level can theoretically reduce the time consumption by 21–66% compared to the SPAVBDERA under the 95% confidence level.

In the future work, more experiments will be conducted. The necessary traffic control lights in the road network, traffic flow and other factors will be considered as well for in-depth path planning research, realizing the load balance of the road network, which can be better applied to the actual navigation.

**Author Contributions:** Conceptualization, H.W. and S.Z.; methodology, X.H. and S.Z.; software, H.W. and S.Z.; validation, H.W., S.Z., and X.H.; writing—original draft preparation, S.Z.; writing—review and editing, H.W.; funding acquisition, X.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Key R&D Program of China under Grant, grant number 2018YFB0505000, and Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, grant number 18102.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Jung, H.; Lee, K.; Chun, W. Integration of GIS, GPS, and optimization technologies for the effective control of parcel delivery service. *Comput. Ind. Eng.* 2006, 51, 154–162. [CrossRef]
2. Garcia-Cruz, X.M.; Sergiyenko, O.Y.; Tyrsa, V.; Rivas-Lopez, M.; Hernandez-Balbuena, D.; Rodriguez-Quinonez, J.C.; Basaca-Preciado, L.C.; Mercorelli, P. Optimization of 3D laser scanning speed by use of combined variable step. *Opt. Lasers Eng.* 2014, 54, 141–151. [CrossRef]
3. Gao, Z.; Yu, Y. Interacting multiple model for improving the precision of vehicle-mounted global position system. *Comput. Electr. Eng.* 2016, 51, 370–375. [CrossRef]
4. Amer, H.; Salman, N.; Hawes, M.; Chaqfeh, M.; Mihaylova, I.; Mayfield, M. An improved simulated annealing technique for enhanced mobility in smart cities. *Sensors* 2016, 16, 1013. [CrossRef] [PubMed]
5. Leandro, D.O.S.; Renata, A.D.M.B.; Gouvêa Campos, V.B. Proposal to planning facility location using UAV and geographic information systems in a post-disaster scenario. *Int. J. Disaster Risk Reduct.* 2019, 36, 101080.
6. Leesmiller, J.D.; Wilson, R.E. Hidden Markov models for vehicle tracking with Bluetooth. In Proceedings of the Transportation Research Board 92nd Annual Meeting, Washington, DC, USA, 13–17 January 2013; p. 14.
7. Song, T.; Capurso, N.; Cheng, X.; Yu, J.; Chen, B.; Zhao, W. Enhancing gps with lane-level navigation to facilitate highway driving. *IEEE Trans. Veh. Technol.* 2017, 4579–4591. [CrossRef]
8. Liu, X.; Khan, K.N.; Farooq, Q.; Hao, Y.; Arshad, M.S. Obstacle Avoidance through Gesture Recognition: Business Advancement Potential in Robot Navigation Socio-Technology. *Robotica* 2019, 37, 1663–1676. [CrossRef]
9. Zou, D.; Niu, S.; Chen, S.; Su, B.; Li, Y. A smart city used low-latency seamless positioning system based on inverse global navigation satellite system technology. *Int. J. Distrib. Sens. Netw.* 2019, 15. [CrossRef]
10. Zhu, D.; Du, H.; Sun, Y.; Cao, N. Research on path planning model based on short-term traffic flow prediction in intelligent transportation system. *Sensors* 2018, 18, 4275. [CrossRef]
11. Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Lenin-Guillermo, L.Z.; Gayraud, T. A Centralized route-management solution for autonomous vehicles in urban areas. *Electronics* 2019, 8, 722. [CrossRef]
12. Song, Q.; Li, M.; Li, X. Accurate and fast path computation on large urban road networks: A general approach. *PLoS ONE* 2018, 13, e0192274. [CrossRef] [PubMed]

13. Lu, X.; Camitz, M. Finding the shortest paths by node combination. *Appl. Math. Comput.* 2011, 217, 6401–6408. [CrossRef]

14. Sedeño-noda, A.; Colebrook, M. A biobjective Dijkstra algorithm. *Eur. J. Oper. Res.* 2019, 276, 106–118. [CrossRef]

15. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* 1959, 1, 269–271. [CrossRef]

16. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* 1968, 4, 100–107. [CrossRef]

17. Dorigo, M.; Maniezzo, V.; Colomi, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man. Cybern. Part. B* 1996, 26, 29–41. [CrossRef]

18. Lin, L.; Wu, C.; Ma, L. A genetic algorithm for the fuzzy shortest path problem in a fuzzy network. *Complex. Intell. Syst.* 2020, 3, 1–10. [CrossRef]

19. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* 1983, 220, 671. [CrossRef]

20. Yue, Y.; Gong, J. An efficient implementation of shortest path algorithm based on Dijkstra algorithm. *J. Wuhan Tech. Univ. Surv. Ma.* 1999, 209–212.

21. Nordbeck, S.; Rystedt, B. *Computer Cartography: Shortest Route Programs*; Royal University of Lund, Department of Geography; Gleerup: Malmo, Sweden, 1969.

22. Wang, X.; Yang, Z.; Lv, X.; Zhao, B. Shortest path algorithm based on limiting parallelogram and its application in traffic networks. *J. Jilin Univ.* 2006, 1, 123–127.

23. Bing, L. Route finding by using knowledge about the road network. *IEEE Trans. Syst. ManCybern Part A Syst. Hum.* 1997, 27, 436–448. [CrossRef]

24. Möhring, R.H.; Schilling, H.; Schütz, B.; Wagner, D.; Willhalm, T. Partitioning graphs to speed up Dijkstra’s algorithm. *J. Exp. Algorithmics* 2005, 11, 1–29.

25. Chondrogiannis, T.; Gamper, J. Exploring graph partitioning for shortest path queries on road networks. In Proceedings of the 26th GI-Workshop Grundlagen von Datenbanken: GvDB’14, Bozen, Italy, 21–24 October 2014; pp. 71–76.

26. Jung, S.; Pramanik, S. An efficient path computation model for hierarchically structured topographical road maps. *IEEE Trans. Knowl. Data Eng.* 2015, 14, 1029–1046. [CrossRef]

27. Climaco, J.C.N.; Martins, E.Q.V. On the determination of the nondominated paths in a multiobjective network problem. *Methods. Oper Res.* 1981, 40, 255–258.

28. Mote, J.; Murthy, I.; Olson, D.L. A parametric approach to solving bicriterion shortest path problems. *Eur. J. Oper. Res.* 1991, 53, 81–92. [CrossRef]

29. Raith, A.; Ehrgott, M. A comparison of solution strategies for biobjective shortest path problems. *Comput. Oper. Res.* 2009, 36, 1299–1331. [CrossRef]

30. Duque, D.; Lozano, L.; Medaglia, A.L. An exact method for the biobjective shortest path problem for large-scale road networks. *Eur. J. Oper. Res.* 2015, 242, 788–797. [CrossRef]

31. Lu, F.; Lu, D.; Cui, W. Time shortest path algorithm for restricted searching area in transportation networks. *J. Image Graph.* 1999, 10, 3–5.

32. Fu, M.; Li, J.; Deng, Z. A route planning algorithm for the shortest distance within a restricted searching area. *J. Beijing Inst. Technol.* 2004, 10, 881–884.

33. Zhou, W.; Qiu, Q.; Luo, P.; Fang, P. An improved shortest path algorithm based on orientation rectangle for restricted searching area. In Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Whistler, BC, Canada, 27–29 June 2013; pp. 692–697.

34. Stutzle, T.; Dorigo, M. A short convergence proof for a class of ant colony optimization algorithms. *IEEE Trans. Evol. Comput.* 2002, 6, 358–365. [CrossRef]