Sensors and Materials, Vol. 34, No. 11 (2022) 4039–4051
MYU Tokyo

S & M 3095

K-shell Algorithm-based Method for Identifying Important Nodes in Rescue Paths

Qing Xu, Tianming Zhao,* Xinming Zhu, and Ruoxu Chen

College of Geo-Spatial Information, PLA Strategic Support Force Information Engineering University,
62 Science Road, Zhongyuan District, Zhengzhou 450001, China

(Received August 1, 2022; accepted October 25, 2022)

Keywords: K-shell, degree, complex network, shortest path, node importance

The importance evaluation of nodes is an important part of traffic network analysis. To avoid targeted assaults and resolve path planning issues in complicated traffic networks, it is advantageous to choose and evaluate important path nodes. This has important practical significance. In this study, we propose a method based on the K-shell algorithm for finding important nodes in a rescue path using the shortest rescue path with the provided starting and ending points. Firstly, a complex network is built using the original data of the road network, and the initial shortest path is solved on the basis of the starting and ending points. Then, Ks values of all nodes on the shortest path are calculated and the nodes are arranged hierarchically. Next, the shortest path is replanned by deleting some important nodes, which is followed by analyzing and comparing the change in the length of the updated shortest path. Finally, the above operations are performed in turn, and the important nodes in the process of planning the rescue path are screened out. To assess the rationality of the selection of important nodes, the degree D of the node and the quantity N of nodes in the two-step neighborhood of each important node are chosen. The experimental results show that the proposed method can quickly and effectively identify the important nodes that need key protection when planning rescue paths.

1. Introduction

As an important urban facility, the road network is the basis for planning optimal rescue paths. In rescue path planning, the identification and screening of key nodes can provide a basis for the risk assessment of nodes, the protection of important nodes, and the priority of risk disposal of nodes. Therefore, it is highly meaningful to study how to identify important nodes in complex networks or rescue paths. The key of this work is to determine the indicators for evaluating node importance, and then effectively guide the protection of emergency rescue paths.

At present, the evaluation indicators of complex network nodes mainly include the degree of importance centrality,(1) closeness centrality,(2) betweenness centrality,(3) centrifugal centrality,(4) compactness centrality,(5) eigenvectors,(6) and H-index.(7) Extensive research has

---

*Corresponding author: e-mail: 1961620500@qq.com

https://doi.org/10.18494/SAM4063

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

been conducted using complex network models to extract key nodes and rank their importance. Wang *et al.* evaluated the importance of nodes on the basis of their neighbor node degrees of importance, and believed that the greater the degrees of importance of a node's neighbor nodes, the more important the node is.(8) Lü *et al.* constructed the H-indicator centrality by considering the degrees of neighbor nodes and the degree of the node itself, so as to determine the importance of a node.(9) Makse *et al.* proposed a coarse-grained decomposition method based on the global structure of the network and judged the importance of the node by its position in the network, which was related to the integrity of the network.(10) Yang *et al.* considered the contact frequency and the length of the contact time between nodes and proposed a method for evaluating node importance based on the PageRank algorithm.(11) Wang and Guo constructed a multiple influence matrix and proposed a method for evaluating node importance based on a directed weighted network.(12) Kong *et al.* proposed a method for evaluating node importance based on the dynamic model of complex networks for undirected weighted networks.(13) Wang *et al.* used an improved fuzzy C-means clustering algorithm based on the K-means algorithm and weighted random forest to determine the importance of nodes in a transportation network.(14) The TOPSIS method adopted by Yu *et al.*,(15) the principal component analysis method used by Hu *et al.*,(16) and the multi-indicator evaluation method of node importance proposed by Li and Liu(17) all considered the importance of a node to be affected by various factors and integrated several different ranking methods. Zhang *et al.* analyzed the characteristics of many types of commonly used evaluation indicators and determined the importance of each node by the grey relational analysis method.(18)

Existing methods can extract key nodes under corresponding rules in complex networks and evaluate their importance. However, they face the problems of low accuracy, high computational complexity, and difficulty in taking into account global and local structures. To address these problems, in this study, we propose a method for identifying important nodes in rescue paths using the K-shell algorithm. The K-shell algorithm is used to sort the path nodes hierarchically, and the nodes in the layer with the highest *Ks* value when planning the path are sequentially extracted. Finally, the key nodes in planning rescue paths are screened out, providing a reasonable reference for the selection of rescue routes and the formulation of schemes in specific scenarios.

2. K-shell Algorithm and Analysis

2.1 K-shell algorithm

The K-shell algorithm is a global node importance evaluation method that assigns a *Ks* value to each node to quantify its importance.(19,20) The basic idea of the algorithm is to continuously delete the nodes with the same degree from the outer nodes with a degree of 1 in the network and assign the corresponding *Ks* value to the deleted nodes. Finally, the importance of the nodes can be judged by their *Ks* value.

The process of the K-shell algorithm is as follows.

Step 1: Construct the network graph G_1.
Step 2: Calculate the degree of each node in the network.
Step 3: Delete the nodes with degree $D = i$ ($i = 1, 2, 3...$).
Step 4: Classify the deleted nodes as the i-shell layer and assign Ks values to them as

$$Ks(i) = i\,. \tag{1}$$

Step 5: Repeat the above steps until nodes with degree i no longer appear in the network.
Step 6: Process all the values of D in turn, so that all nodes acquire their own Ks values.

The pseudocode of this algorithm is shown in Algorithm 1.

2.2 Analysis of K-shell algorithm

The K-shell algorithm judges the importance of nodes based on their positions in the network and has low time complexity, which is in line with the actual requirements in road network applications. This algorithm provides a theoretical basis for planning rescue paths. The K-shell algorithm itself also has certain defects. When implementing the layering operation in accordance with the K-shell algorithm, the obvious differences in importance between layers can be obtained, but the nodes in the same layer can hardly be distinguished.(21) A large area of nodes in the same layer will appear in a network with a relatively simple structure, making it difficult to extract key nodes.

3. Method of Identifying Important Nodes in Paths Using K-shell Algorithm

In view of the characteristics of the K-shell algorithm analyzed in the previous section, this paper focuses on the actual needs of rescue path planning, and we propose a method for identifying important nodes in the path using the K-shell algorithm.

3.1 Design ideas

The idea of the method is roughly as follows. First, the starting and ending points are determined, and the shortest rescue path is searched. The K-shell algorithm is used to calculate

Algorithm 1
K-shell algorithm.

```
Input: network graph $G = (G, V)$
Output: $Ks$ value of each node in the network
For node $V_1$ in the network graph
Repeat
For nodes with degree $K$ in the network graph
Assign its $Ks$ value with $K$
Delete node $V_1$ and its connected edges in the network graph
Update the value of the node
Until the degree of all remaining nodes in the network graph > $K$, $K = K + 1$
```

the *Ks* value of the path nodes between the starting and ending points and record them hierarchically. Then, the shortest path is replanned by deleting the nodes in the layer with the highest *Ks* value, and the length changes of the updated shortest path are analyzed and compared. Next, the above operations are performed in turn. Finally, the important nodes in the process of planning the rescue path are screened out.

3.2 Method design

The method flowchart is shown in Fig. 1, and the specific steps are as follows.

Step 1: Select the starting and ending points, and select the rescue starting point *o* and ending point *d* in the road network.

Step 2: Use the Dijkstra algorithm to find the initial shortest path S_0 between two points.

Step 3: Use the K-shell algorithm to calculate the *Ks* values for the nodes on the path and arrange them in layers in descending order.

Step 4: Extract the node in the highest layer or layers of the highest *Ks* value as the key node N_0 of the shortest path.

Step 5: Delete the key node N_0 of the shortest path to generate a new road network.

Step 6: Repeat Steps 2 to 5, and sequentially record the shortest path *Si* updated by each iteration and the key node *Ni* on the path until there is no path connecting the starting point and ending point.

Step 7: Calculate the difference ΔD_i between the updated shortest path *Si* and the initial shortest path *s*.

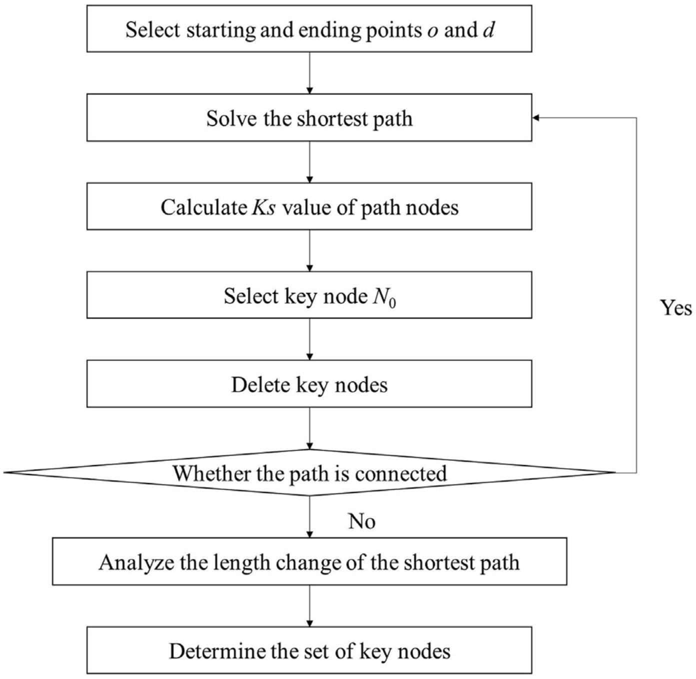


Fig. 1. (Color online) Flowchart of the method.

Algorithm 2.
Method of identifying important nodes in paths based on K-shell algorithm.

```
Input: road network $G = (V, E)$, rescue starting point $o$, ending point $d$
Output: set of critical nodes to be protected

Repeat
    Calculate the shortest path between the starting point $o$ and the ending point $d$
    For node $V$ on the path
        Repeat
            Calculate the $Ks$ value on the path node
            Extract and record the key node $N_0$ on the path
            Delete the key node $N_0$ in the network graph
            Rebuild the network diagram
        Until the starting point $o$ and the ending point $d$ in the network graph $G$ are no longer connected
Analyze the path length change $\Delta D_i$ and determine the set of important nodes
End
```

Step 8: Analyze any abnormal value of the path length change ΔD_i, and determine the set of important nodes of the path.

The pseudocode of this method is shown in Algorithm 2.

4. Experimental Results and Analysis

4.1 Experimental data

The experimental data is the navigation road data of the Zhengzhou High-tech Zone. The data has dense road nodes, the node spacing is relatively small, the connectivity between nodes is large, and the data redundancy is large, as shown in Fig. 2.

The Python programming language was used to process the shapefile data of roads in the Zhengzhou High-tech Zone to build a complex network. The generated network diagram

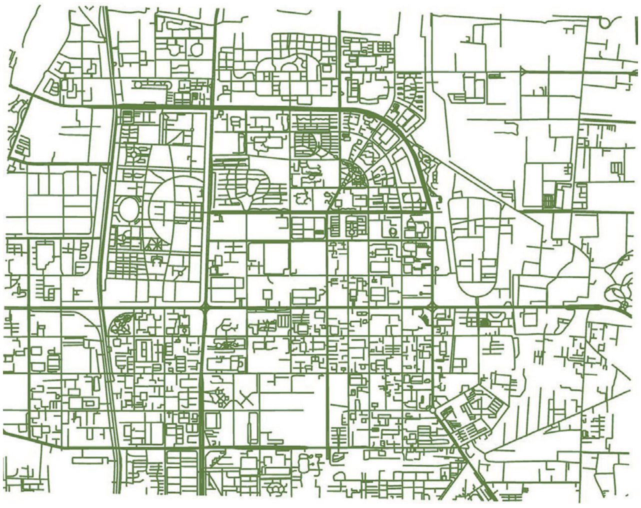


Fig. 2. (Color online) Experimental data of navigation roads in Zhengzhou High-tech Zone.

structure is shown in Fig. 3. The characteristic indicators of the road network were calculated. The results are shown in Table 1. In the road network of the Zhengzhou High-tech Zone, there are 5627 nodes and 7438 edges. The average degree of the network is 2.64, that is, each node in the network is connected to 2.64 nodes on average, and the average clustering coefficient and the overall efficiency of the network are low.

4.2 Experimental results

The method proposed in this paper was used for the calculation. When the number of iterations was 15, there was no road connecting the starting and ending points. The overall result and some intermediate results are respectively shown in Figs. 4 and 5. The green concentric circles near the top and bottom of Fig. 4 are starting point 2470 and ending point 3227, respectively, the blue line in each figure is the shortest path generated by the iteration process, and the red nodes represent the key nodes on the shortest path.

During the iterative process of the method, we recorded the calculated shortest path length, the difference from the initial shortest path length, the number of deleted important nodes, the *Ks* value, and the node numbers, as shown in Table 2, where the initial shortest path length is 4140.51 m.

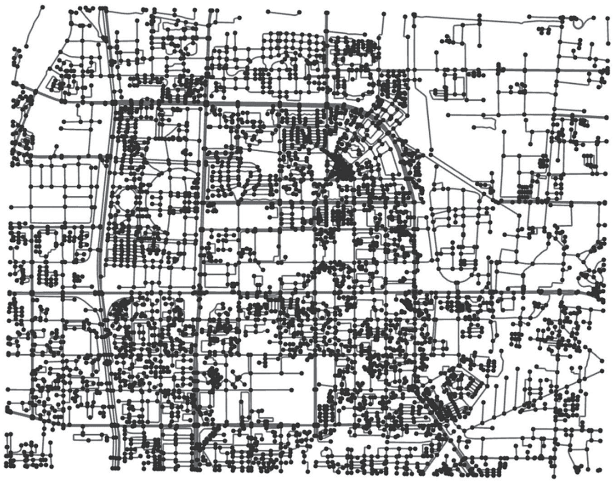


Fig. 3. Road network structure of Zhengzhou High-tech Zone.

Table 1
Road network characteristic indicators of Zhengzhou High-tech Zone.

Network characteristic indicator	Amount of computation
Number of nodes	5627
Number of edges	7438
Minimum degree	1
Maximum degree	8
Average degree	2.64
Average clustering coefficient	0.0058
Global network efficiency	0.0971

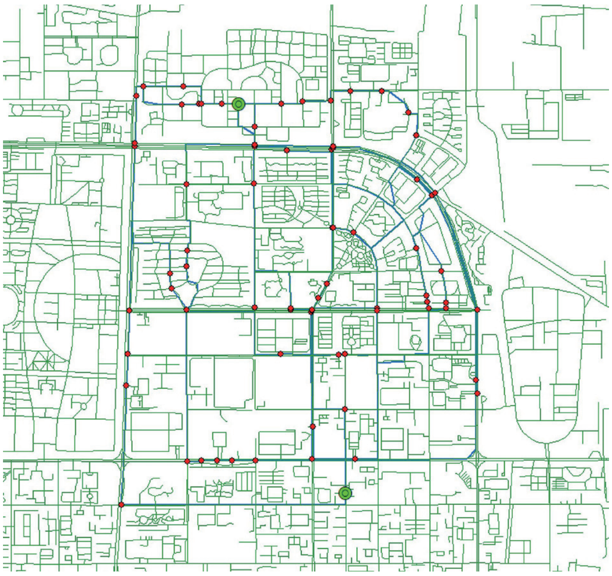


Fig. 4. (Color online) Overall experimental result.

Fig. 5. (Color online) Experimental results of the 1st, 2nd, and 10th iterations.

4.3 Experimental analysis and evaluation

4.3.1 Experimental analysis

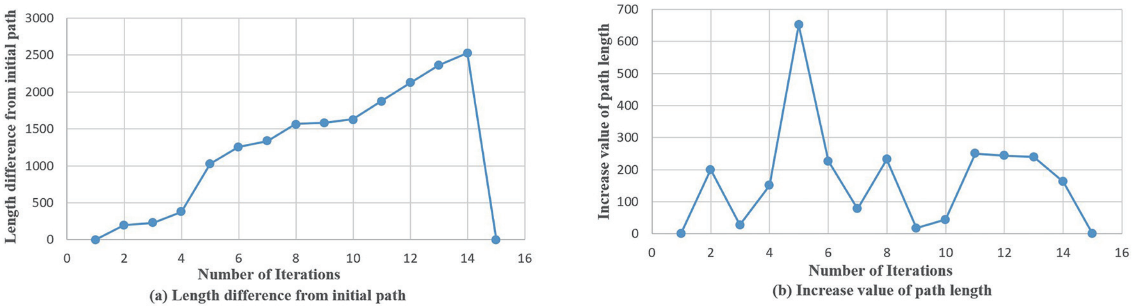
In the planning process of optimal rescue paths, depending on the goal of solving the path, rescue paths can be divided into five types: the path with the shortest travel distance, the path with the shortest travel time, the path with the least congestion, the path with the optimal road condition, and the path with the least travel cost. Considering the characteristics of the data, in our experiment, we adopted the travel distance as the indicator and the shortest rescue path as the object to identify the key nodes that need protection when planning the rescue path.

Table 2
Statistical data of the calculation results obtained by the proposed method.

Number of iterations	Number of deleted nodes	*Ks* value and node numbers	Shortest path length	Difference from initial path
1	5	*Ks* = 7: 2789, 2788, 2920, 2910, 2905	4141.56	1.04
2	4	*Ks* = 4: 2542, 2529, 2543, 3220	4341.23	200.72
3	3	*Ks* = 5: 3104; *Ks* = 4: 3038, 2938	4369.09	228.58
4	2	*Ks* = 4: 3117, 2989	4519.86	379.35
5	6	*Ks* = 5: 2536; *Ks* = 4: 2765, 3140, 3344, 3549, 3360	5171.31	1030.80
6	4	*Ks* = 6: 3959; *Ks* = 5: 3891, 3952, 3957	5397.89	1257.36
7	3	*Ks* = 5: 3548; *Ks* = 4: 3141, 2923	5476.23	1335.72
8	5	*Ks* = 7: 2210, 2273, 2363, 2434, 2556	5708.96	1568.45
9	2	*Ks* = 7: 3892, 4016	5726.87	1586.36
10	3	*Ks* = 5: 4072, 4121, 4116	5771.31	1630.80
11	2	*Ks* = 5: 3142, 4377	6021.84	1881.32
12	5	*Ks* = 5: 2540, 2188, 2205, 2202, 3218	6265.46	2124.95
13	10	*Ks* = 6: 3885; *Ks* = 5: 2765, 3140, 3344, 3549, 3360, 3827, 4038, 4403, 4406	6505.35	2364.84
14	6	*Ks* = 9: 1810; *Ks* = 6: 2263, 2158, 1805, 1799, 1773	6667.80	2527.29
15	12	*Ks* = 6, 2718; *Ks* = 5, 2390, 2271, 2163, 1920, 1857, 1848, 1845, 2099, 2101, 2190, 3173	Not connected	

During the experiment, with the successive deletion of key nodes on the shortest rescue path, the degree of damage of the road network structure in the area gradually increased. Therefore, when the key structures of the road network between the starting and ending points of the connection are all eliminated, the length of the newly generated shortest path will increase significantly. By observing the length difference between the updated shortest path and the initial shortest path in each iteration process in Table 2, we can analyze the rule underlying the change, as shown in Fig. 6. Figure 6(a) shows the length difference between the shortest path and the initial shortest path generated for each number of iterations. Figure 6(b) shows the increase in the generated shortest path length relative to the previous path.

The data in Table 2 and Fig. 6 shows that when the number of iterations is 1–4, the path length difference is 1.04, 200.72, 228.58, and 379.35, respectively. Figure 6 shows that when the number of iterations is 5, the length difference is 1030.80, 651.45 larger than that for the fourth iteration [Fig. 6(b)], whereas the results for the other iterations all show an increase between 0 and 300. Through this analysis, we infer that when the number of iterations is 5, the destructiveness of the cumulatively deleted nodes to the road network structure between the starting and ending points reaches a critical value, and the shortest path length between the starting and ending points consequently increases significantly. Therefore, we consider that the nodes deleted in the first five iterations need key protection in rescue path planning. The number of nodes in the finally obtained paths is 20, which are nodes 2789, 2788, 2920, 2910, 2905, 2542, 2529, 2543, 3220, 3104, 3038, 2938, 3117, 2989, 2536, 2765, 3140, 3344, 3549, and 3360, as shown in Fig. 7.



(a) (b)

Fig. 6. (Color online) Change law of the shortest path

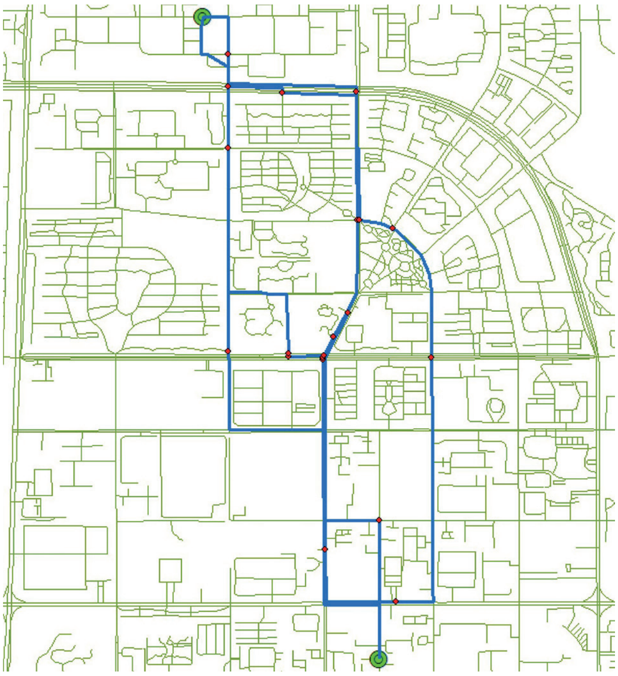


Fig. 7. (Color online) Distribution of the key nodes.

It can be seen from Fig. 7 that most of the identified key nodes are concentrated on the main roads between the starting and ending points. Among them, in the area between the starting and ending points, the number of key nodes on the east–west main road (Cuizhu Street) is 7 and the number of key nodes on the north–south main road (Xuesong Road) is 9; both of these roads make a strong contribution to the relatively high connectivity in this area. The other nodes are distributed on the streets such as Science Avenue and Lianhua Street, which are the core streets of the High-tech Zone and the main streets that bear the traffic flow. From the analysis results, the key nodes identified by this method are basically consistent with the nodes that play an important role in the connectivity of the road network, which is in line with the actual expected effect.

In the process of rescue path planning, priority should be given to protecting the intersections and main streets corresponding to key nodes. In this experiment, on the premise that the starting and ending points have been determined before planning the rescue paths, priority should be given to clearing the three main roads of Cuizhu Street, Lianhua Street, and Xuesong Road, and the intersection of Xuesong Road, Jinju Road, and Cuizhu Street should be particularly protected. A schematic diagram of the planning is shown in Fig. 8, in which the red circles represent the intersections that need key protection, and the thick purple lines represent the roads that need key unblocking.

4.3.2 Importance evaluation

In an urban road network, the importance of nodes in the entire network structure can be reflected in their connectivity and degree of aggregation. The degree *D* of a node can reflect its connectivity. It is generally believed that the higher the degree of a node, the greater its accessibility and the more important it is in the network. The number *N* of nodes in a neighborhood of one node can reflect the degree of aggregation of nodes to a certain extent. It is generally believed that the greater the number of nodes in a certain neighborhood, the more complex the local structure of the node and the more important it is in the network.

To evaluate the effectiveness of the proposed method, the degree *D* of nodes and the number of nodes *N* in the two-step neighborhood of a node are used to evaluate the rationality of the

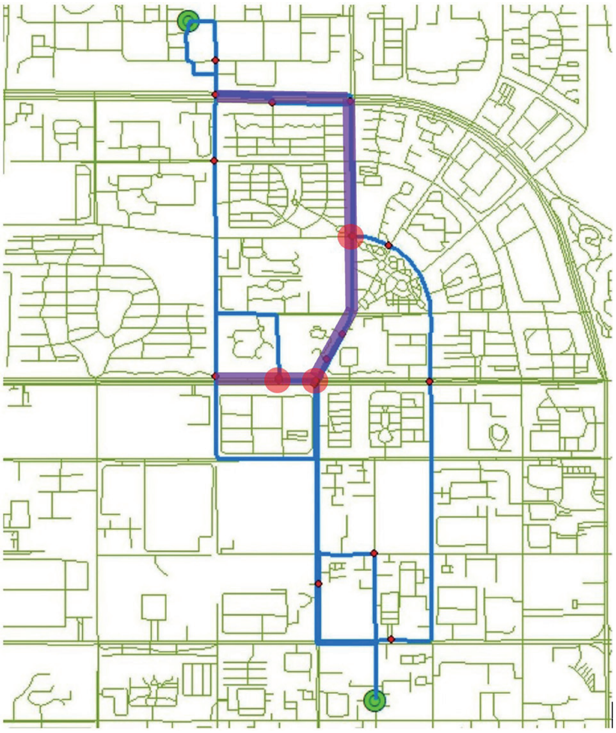


Fig. 8. (Color online) Schematic diagram of path planning.

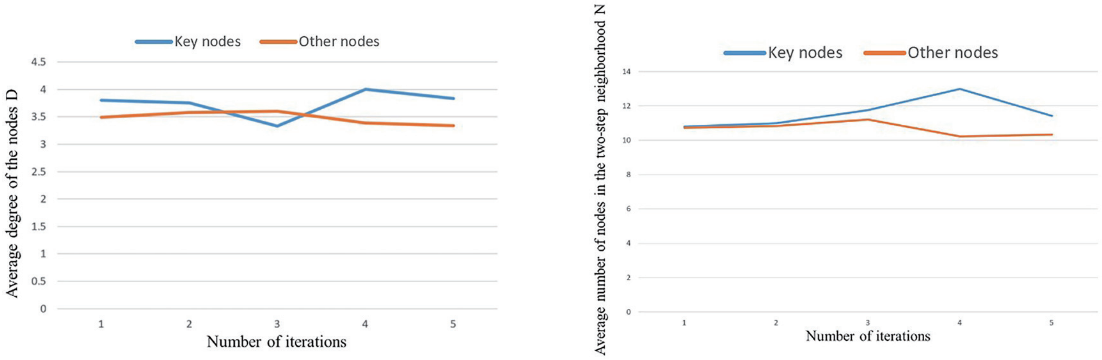
proposed method for identifying important nodes. Calculation results are shown in Table 3, which lists the degrees of key nodes, the average number of nodes in the two-step neighborhood, the degrees of the remaining nodes on the shortest path, and the average number of nodes in their two-step neighborhood.

It can be seen from Fig. 9(a) that when the number of iterations is 1, 2, 4, and 5, the average value of D of the key nodes on the shortest path is greater than that of other nodes on the path, and only when the number of iterations is 3 is the average value of D of the key nodes on the shortest path slightly smaller than that of other nodes on the path; overall, the degree of the key nodes on the shortest path is the largest. Figure 9(b) shows that in the iterative process, the average value of N in the two-step neighborhood of the key node on the shortest path is larger than that of other nodes on the path.

In addition, the betweenness centrality (BC) and closeness centrality (CC) are used for comparison to further verify the effectiveness of the proposed method. Table 4 lists the average

Table 3
Importance evaluation results.

Number of iterations	D value of key nodes	D value of remaining nodes	N value of key nodes	N value of remaining nodes
1	3.8	3.49	10.8	10.72
2	3.75	3.58	11	10.84
3	3.33	3.6	11.76	11.2
4	4	3.39	13	10.23
5	3.83	3.34	11.43	10.34



(a) (b)

Fig. 9. (Color online) Evaluation results of key nodes. (a) Average degree of the nodes. (b) Average number of nodes in the two-step neighborhood.

Table 4
Importance evaluation results of two methods.

Number of iterations	D value of key nodes by BC	N value of key nodes by BC	D value of key nodes by CC	N value of key nodes by CC
1	4	12.4	3.8	12
2	3.75	11.25	3.75	12.25
3	3.66	10.32	3.66	12
4	4	12.5	3.5	11
5	3.66	11.49	4.33	13.66

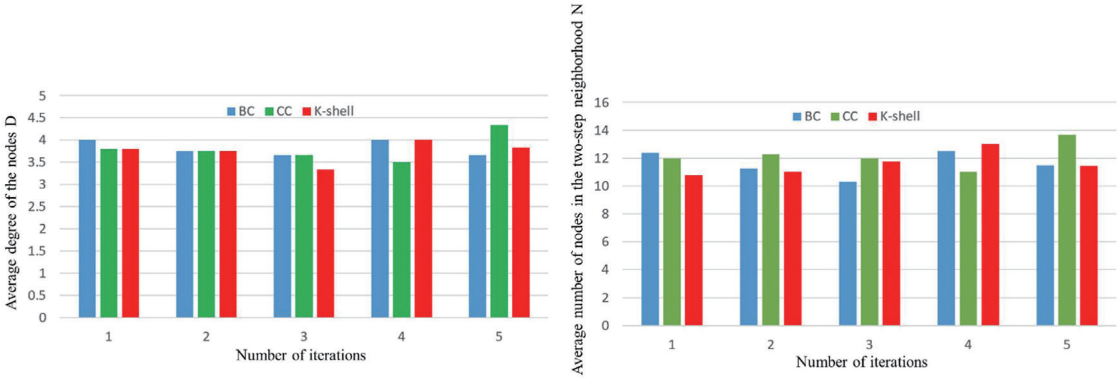


Fig. 10. (Color online) Comparison of the three methods.

values of *D* and *N* obtained by the BC and CC methods, which are different from those obtained by the proposed method. The results of the comparison are shown in Fig. 10.

The data in Table 4 and Fig. 10 show that, for the two indicators *D* and *N*, the average differences between the proposed method and the BC method are 0.15 and 0.77 and the maximum differences are 0.33 and 1.6, and the average differences between the proposed method and the CC method are 0.27 and 1.38 and the maximum differences are 0.5 and 2.23, respectively. The average values of *D* and *N* obtained by the three methods are roughly the same. In addition, the key nodes obtained by the three methods are mostly distributed on the main roads, mainly, Xuesong Road, Cuizhu Street, Lianhua Street, and Science Avenue. In terms of efficiency, under the experimental conditions of E-2176M, 2.7 GHz, and NVIDIA Quadro P600, the proposed method is the most efficient, with a running time of 40.06 s, compared with 246.04 and 282.55 s for the BC and CC methods, respectively.

In summary, from the perspective of the degree *D* of nodes and the number of nodes *N* in the two-step neighborhood, the nodes identified by the proposed method are all important in the road network with high accuracy. At the same time, the efficiency of the proposed method is better than those of the BC and CC methods. The proposed method can provide a basis for the protection of important nodes in the road network.

5. Conclusion

Taking the planning of rescue paths in complex traffic networks as a requirement, we comprehensively consider the complexity of the road network structure and propose a method for identifying important nodes in rescue paths using the K-shell algorithm, by which the important nodes in the process of planning rescue paths are screened out. The starting and ending points are randomly selected in the Zhengzhou High-tech Zone for simulation experiments, and the important nodes in rescue path planning are extracted by the proposed method. The experimental results show that most of the identified important nodes are located on the main roads and important intersections between the starting and ending points. In addition, the values of two indicators, i.e., the degree of important nodes and the number of nodes in the two-step neighborhood, of the important nodes are larger than those of the other nodes on the path between the starting and ending points. For specific scenarios and tasks, the proposed method can quickly and effectively identify the key nodes in complex networks for rescue path planning, improve the efficiency of road network safety protection and emergency responses, and provide a useful reference for formulating subsequent rescue plans.

The proposed method and its application only consider the road length, and future work will further consider the road width, grade, ancillary facilities, and other factors. In addition, information detected using sensor technology in the traffic field is important for urban traffic planning and traffic management. Using sensors to collect real-time traffic information and integrating the monitoring results into the proposed method can effectively improve the real-time formulation of planning rescue paths and further improve the scientificity and practicability of the method.

References

1 E. Y. Yu, Y. Fu, Q. Tang, J. Y. Zhao, and D. B. Chen: IEEE Access **8** (2020) 211281. https://doi.org/10.1109/ACCESS.2020.3038791
2 X. L. Yan, Y. P. Cui, and S. J. Ni: Chin. Phys. B **29** (2020) 048902. https://doi.org/10.1088/1674-1056/ab77fe
3 K. Berahmand, A. Bouyer, and N. Samadi: Computing **101** (2019) 1711. https://doi.org/10.1007/S00607-018-0684-8
4 P. Hage and F. Harary: Social Networks **17** (1995) 57. https://doi.org/10.1016/0378-8733(94)00248-9
5 S. P. Borgatti and M. G. Everett: Social Networks **28** (2006) 466. https://doi.org/10.1016/j.socnet.2005.11.005
6 M. E. J. Newman: Phys. Rev. E **74** (2004) 036104. https://doi.org/10.1103/PhysRevE.74.036104
7 Z. B. Wang, Y Wang, J. M. Ma, W. Y. Li, N. Chen, and X. Z. Zhu: Physica A **527** (2019) 121. https://doi.org/10.1016/j.physa.2019.121184
8 J. W. Wang, L. L. Rong, and T. Z. Guo: J. Dalian Univ. Technol. **05** (2010) 822. https://doi.org/10.7511/dllgxb201005035
9 L. Y. Lü, T. Zhou, Q. M. Zhang, and S. H. Eugene: Nat. Commun. **7** (2016) 440. https://doi.org/10.1038/ncomms10168
10 H. A. Makse, M. Kitsak, L. K. Gallos, S. Havlin, L. Muchnik, H. E. Stanley, and F. Liljeros: Nat. Phys. **6** (2010) 888. https://doi.org/10.1038/nphys1746
11 Y. Y. Yang, G. Xie, and J. Xie: Discrete Dyn. Nat. Soc. **2017** (2017) 1. https://doi.org/10.1155/2017/9741824
12 Y. Wang and J. L. Guo: Acta Phisica Sinica **66** (2017) 19. https://doi.org/10.7498/aps.66.050201
13 J. T. Kong, J. Huang, and J. X. Gong: Acta Phisica Sinica **67** (2018) 255. https://doi.org/10.7498/aps.67.20172295
14 L. L. Wang, M. Huang, and L. Gao: J. Transport Inf. Safety **2** (2020) 80. https://doi.org/10.3963/j.jssn.1674-4861.2020.02.010
15 H. Yu, Z. Liu, and Y. J. Li: Acta Physica Sinica **2** (2013) 54. https://doi.org/CNKI:SUN:WLXB.0.2013-02-008
16 G. Hu, X. Xu, W. M. Zhang, and J. Zhou: Acta Electronica Sinica **2** (2019) 358. https://doi.org/CNKI:SUN:DZXU.0.2019-02-015
17 L. Li and Y. Q. Liu: J. Naval Eng. Univ. **5** (2010) 69. https://doi.org/10.3969/j.issn.1009-3486.2010.05.015
18 K. Zhang, H. B. Shen, and H. Zhang: J. Nanjing Univ. Sci. Technol. **4** (2012) 579. https://doi.org/10.14177/j.cnki.32-1397n.2012.04.001
19 L. Q. Qiu, J. Y. Zhang, and X. B. Tian: Appl. Intell. **7** (2021) 1. https://doi.org/10.1007/S10489-020-02132-1
20 A. Ullah, B. Wang, J. F. Sheng, J. Long, and N. Khan: Complexity **2021** (2021). https://doi.org/10.1155/2021/8403738
21 L. X. Xie, H. H. Sun, H. Y. Yang, and L. Zhang: J. Tsinghua Univ. (Sci. Technol.) **5** (2022) 849. https://doi.org/10.16511/j.cnki.qhdxxb.2022.25.041