A Study on Escape Path Planning of Multi-Source/Multi-Sink for Public Buildings

Yi Zhang, Tianqi Liu, Chi Wang, and Chenlei Xie

Abstract With the development of urbanization and the rise of commercial center, the complex internal structure of public commercial buildings and the high density of people make the evacuation of people more aimless, and accurate path planning plays an important role in escape and evacuation. Dijkstra algorithm can only satisfy the single point to single point shortest path planning situation and cannot actually solve the shortest path planning problem with many source points and sink points. Thus, this paper studies the shortest path of escape with multi-source/multi-sink, and proposes an improved Dijkstra algorithm, which can solve the shortest path planning task of escape from multi-source point to multi-sink point and has fast and efficient characteristics of the original Dijkstra algorithm, thus improving the escape efficiency of personnel. In the end, this paper simulates the improved algorithm using the data of different distribution density of different people. The experimental results show that the proposed algorithm is more practical for the shortest path planning from multi-source to multi-sink under the condition of relatively dispersed personnel.

Keywords Dijkstra algorithm · Multi-source/multi-sink (MSMS) · Algorithm optimization · Path planning

1 Introduction

With the development of economy, the volume and function of buildings are increasing, and the structure is becoming more and more complex. Nowadays, large buildings often carry a large number of people who engage in various kinds of work for a long time. It is worrying that the complexity of the structure of the building makes the personnel in the interior often unable to move to a safe place in time

Y. Zhang · C. Xie (✉)
Anhui Province Key Laboratory of Intelligent Building and Building Energy Saving, Anhui Jianzhu University, Hefei, China
e-mail: 1033904749wc@gmail.com

T. Liu · C. Wang
School of Electronic and Information Engineering, Anhui Jianzhu University, Hefei, China

Y. Li et al. (eds.), *Advances in Simulation and Process Modelling*,
Advances in Intelligent Systems and Computing 1305,
https://doi.org/10.1007/978-981-33-4575-1_6

when they encounter emergency disasters such as fire, thus causing certain casualties. Therefore, in order to reduce or even avoid such a thing, we need to design a safe evacuation algorithm strategy, so that in the event of an emergency, the relevant institutions can conduct timely and effective path guidance to a large number of people in the building according to the algorithm strategy, so that people can evacuate in the shortest time through a safe route.

For the research about the path selection strategy, the Dijkstra algorithm [1] refers to the idea of greedy algorithm, which the source node traverses each node in the search graph until it reaches the target node, so that the shortest path from the source node to each node including the target node in the graph is obtained. And because Dijkstra algorithm is more usable and accessible, many scholars, based on the original algorithm, have proposed their own ideas about how to use or improve the Dijkstra algorithm. Gbadamosi and Aremu [2] noticed that sometimes finding the shortest path could cost huge energy, so they modified the Dijkstra algorithm by entailing a specially designed text file to get the alternate routes. And by comparing with the classical algorithm, the outcome of the new method was more effective. Deng, et al. [3] discussed how to properly use the Dijkstra algorithm under an uncertain environment, especially when the parameters are all fuzzy. And by adopting the graded mean integration representation of fuzzy numbers, they successfully improved the classical Dijkstra algorithm, and got a good outcome. And Fusic, et al. [4] also tried to implement Dijkstra algorithm in the mobile robot to let it find the most efficient path under the complicated environment. They modified the parameters of Dijkstra algorithm and adopted V-REP simulation software, the results were inspiring, and this proposed method could be used for path planning of robots. In this paper, we will discuss a new method to deal the multi-source point to multi-sink point problem by improving the Dijkstra algorithm.

2 Dijkstra Algorithms and Building Models

2.1 Dijkstra Algorithm

Dijkstra algorithm is proposed by Edsger Wybe Dijkstra in 1959 to find the shortest path of single point to single point [5], which is mainly used to solve the problem of finding the shortest path between vertices in directed weight graph. The algorithm has been used to build the intelligent fire evacuation system [6]. In addition, the algorithm adopts greedy algorithm strategy, which means that each time it will traverse to the node, which has not been visited, with the least distance from the source node, then one round of traversal is terminated. This will be repeated until searching to the sink point. According to the weight of the edge connected between vertices, the value of the distance between each vertex, which has been traversed, and corresponding source node is stored or updated, which can be used to update the “backtracking” of the weights.

The basic idea of Dijkstra algorithm is that first supposing a formula, named $G = (V, E)$, where G is used to represent a weight graph, which there are some connected edges, vertexes all with positive weights in it, and V E are used to represent the vertex set and the edge set, respectively. Then, dividing V into two groups, one group is named as S, where S is the vertex set that storing the node whose shortest path to the source node has been found, and the source node is represented as V_0. At the beginning, there is only V_0 in S, which can uses the formula, $S = \{V_0\}$, to represent. Then, a new vertex, whose shortest path to V_0 has been obtained, is added to S after each traversal. This loop will be continued until all vertexes in G are added to S, and if there are n nodes, the algorithm will have the n rounds. The other group is named as U, where U is the vertex set that used to store the node whose shortest path to V_0 has not been found, which can uses the formula, $U = \{V_1, V_2, V_3, \ldots, V_n\}$, to represent, which also means that all vertexes except V_0 are in U at the beginning. During each traversal, the length of the distance from each vertex in U to V_0 is arranged in an increasing order, and the vertex which has the smallest length will be selected to join S. In addition, each vertex corresponds to one particular length weights, which uses V_k to represent. And for one vertex in U, if it is found that the weights obtained from the previous rounds are greater than those obtained from the current traversal, then the number of the weights will be updated. The algorithm also stipulates that the length corresponding to each vertex in S set is the shortest distance from the source node to each of them. And the length corresponding to each vertex in U is the current shortest distance from the source node to each of them, which is determined only by the current conditions.

2.2 Building Mathematical Model Construction

Building a reasonable mathematical model is the prerequisite for the effective evacuation simulation. The core of the algorithm is how to choose the optimal path to reduce the evacuation time, and it should also limit the number of people. However, only the length weight of the edge is defined in the original Dijkstra algorithm, the edge, or saying corridor in fact, usually has the limited ability of carrying people. For that reason, based on the original length weight, we define the capacity weight to form the binary weight of the edge. In addition, we also adopt the method of two-dimensional model to find the path in the form of topological map. This map consists of nodes and edges, where nodes represent rooms, exits, etc., and edges represent corridors, passage, etc. To better clarify things in the map, we define some parameters. First, N_{p_i} is the position of a vertex, named i, which can also represent its coordinate, (x_i, y_i). And C_{ij} is used to represent the capacity of path, which means the maximum number of simultaneous evacuees allowed in the path from i to another vertex, named j. W and fp are used to represent the adjacency matrix of all the path and the position of fire points, respectively. n_k means the number of people now still in the fire point, named k. T is the total time consumed by the completion of

the evacuation. And v is the average speed of evacuees. And at the beginning of the algorithm, we will initialize all these parameters, and the process is shown in Fig. 1.

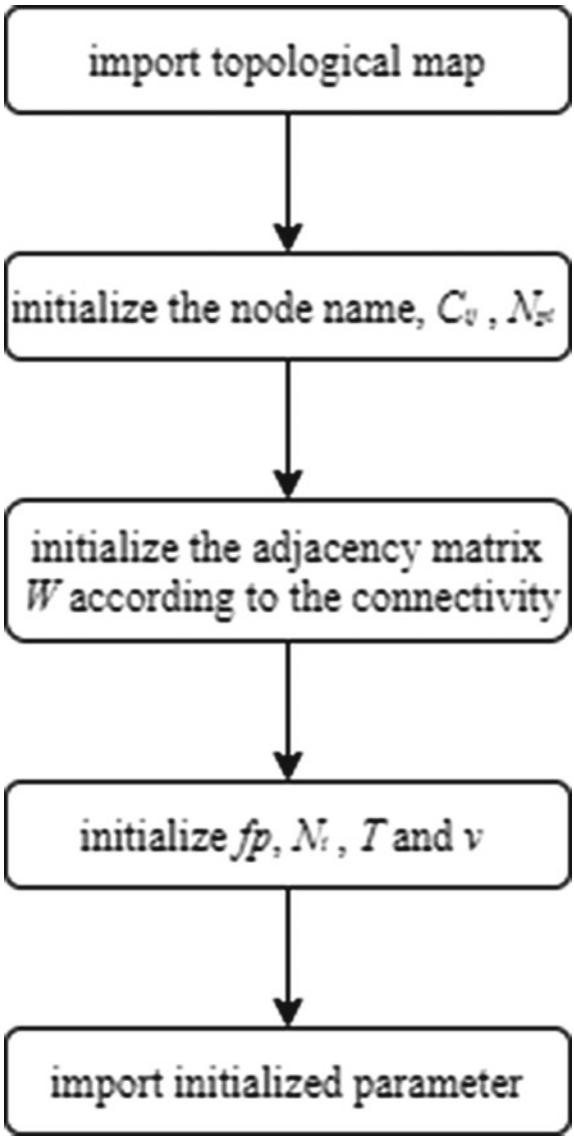

Fig. 1 Flowchart of parameters' initialization

3 Improved Dijkstra Evacuation Path Algorithm Design

3.1 Multi-Source/Multi-Sink Design

Typically, buildings have multiple emergency exits. And whether these buildings are used for office operations or commercial catering, people in them will not just gather in one place, but often disperse in many areas [7]. Therefore, when there is an emergency happening in the building, people in many internal areas all will need to be evacuated. In that case, both exits of building and internal areas with many people are more than one, so we need to consider a situation named multi-source/multi-sink (MSMS) [8]. And this paper designs an evacuation algorithm for MSMS building emergency based on Dijkstra algorithm, using Dijkstra algorithm to plan the shortest path. Dijkstra algorithm uses traversal as the way of searching, the shortest distance from the source point to the target node in the topology which can be found after one round. So at the beginning of our algorithm, the shortest path from each source point to each sink point is found by multiple calls of the Dijkstra algorithm, then these paths are sorted to find the shortest path out. By constantly looking for the shortest path, the personnel can be quickly allocated at the same time in order to make full use of the escape path, so as to solve the MSMS evacuation problem.

3.2 Algorithm Designed

Considering that the original algorithm used alone can only deal with the path searching problem of single point to single point, which cannot really meet the needs of evacuation in real life. So we take "using more escape paths in the shortest time" as the main idea of our algorithm. And this algorithm is implemented by two modules, which are first round of allocation and second to N round of allocation.

(1) First round of allocation
In the actual process, if the path allocation is used properly, there will be a higher probability that the all people can be allocated at once.
First we need to judge whether there is still at least one path that is accessible to exits and whether there is still a person who needs to escape at the fire point. And if so, we will call Dijkstra algorithm to traverse the path between each fire point and exit to find the shortest path. Then the minimum capacity of the whole path, which uses $C_{\min}$ to represent, will be allocated, and the C_{ij} and W will be updated. Finally, until all paths are occupied or all personnel have been evacuated, the first round is over. The specific process is shown in Fig. 2:

(2) Second to N round of allocation
Because there might be too many people in the building, the allocation cannot be down just in the first round. Therefore, we need to design second to N round to evacuate people. And first, we need to release the path capacity consumed by

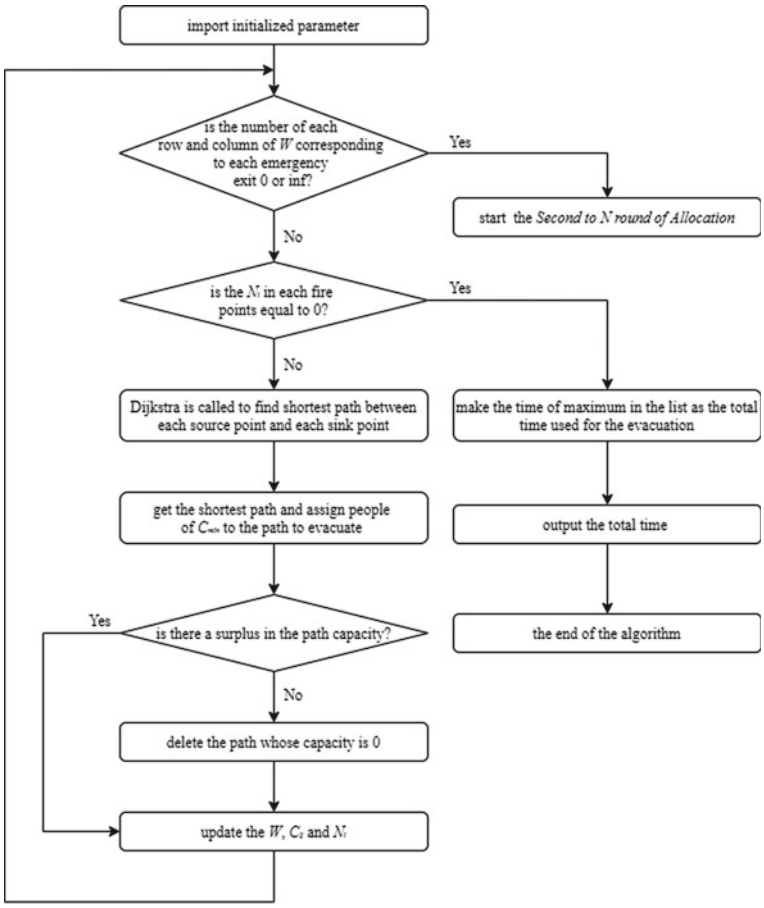

Fig. 2 First round allocation flowchart

the path which takes up the least escape time among all the paths allocated in the first round, and we use $P_{t_{\min}}$ to name it, and add the evacuation time to the time list. Then the W, path capacity and remaining number of fire points will be updated, and the same process as in the first round of the evacuation will also be continued. At last, either iteration will end when the exits are all not accessible, and then, the algorithm will enter the next round or the algorithm is over and outputs the evacuation time when there are no remaining personnel at the fire points. The specific process is shown in Fig. 3.

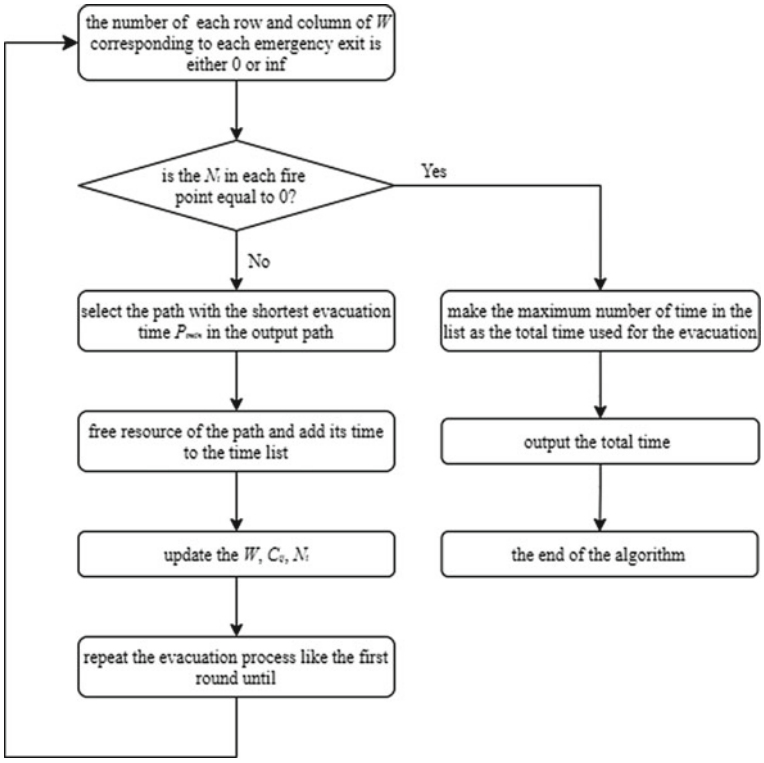

Fig. 3 Second to *N* round of allocation

4 Simulation Study and Performance Analysis

4.1 Simulation Scenario Description

In order to test the practicability of the algorithm, this simulation selected a commercial market as an example.[1] The mall has six safety exits and several internal areas. As the layout is complex, like that all kinds of stores, counters, corridors are mixing together, which makes the recognition very difficult, so in the topology map construction, we selected 17 internal typical areas, representing as nodes, and 38 paths to carry out the simulation of the algorithm [9]. The specific topology is modeled in Fig. 4: (circular nodes are internal regions and square nodes are emergency exits).

Under the condition that the algorithm can basically realize the evacuation of people in buildings, in order to further verify the performance of the algorithm, we have carried out the tests on different population density distribution with different

[1]The simulation study of this research work has been conducted with approval obtained from the owner of one market in Fuyang, China.

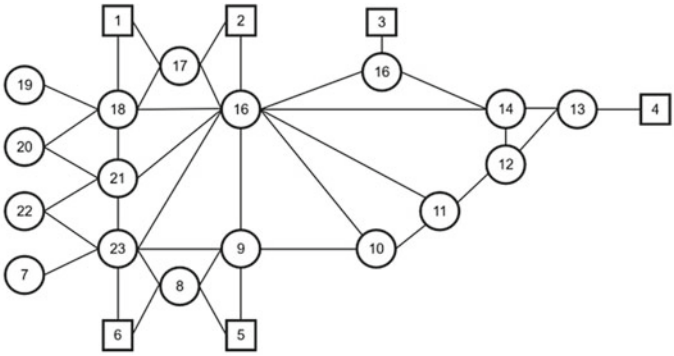

Fig. 4 Simulation building topology

number of people. And we also adopted the basic Dijkstra algorithm, which uses the way of single source to single sink, to compare with the improved algorithm under the same distribution. And the main focus of these tests are comparing the performance of above two algorithms under the different conditions and judging the influence of the difference in population density distribution on the simulation results. In addition, this paper designs two situations of crowd concentration and crowd dispersion, and simulates the evacuation process under the condition that the total number of evacuees is 180, 300, and 450, respectively.

In addition, in order to make the simulation of the evacuation algorithm be more realistic, this algorithm is preset in the following conditions:

(1) The location of the fire point and the distribution of the escape personnel are always initialized in a disorderly and random manner to ensure that the evacuation results are not affected by the regular distribution of the personnel;

(2) This algorithm is defined as a simulated evacuation drill. In the case of simulation, **it is easier for people to calm down, thus avoiding the possibility of making the wrong choice.** In addition, the simulation drill makes people more likely to follow the route planning made by the algorithm, so as to better test algorithm performance;

(3) In the evacuation simulation, the relevant personnel need to be evacuated in the form of trot to reduce the impact of the travel speed on the overall evacuation completion time. Taking into account the different physical fitness of individuals, so the speed of people in this simulation will take a fairly average value of 2 m/s, and evacuees, on the same round, in the same path will act together at the same time. After the path has been determined, the time required for the evacuation is obtained by subtracting 2 m/s from the path length.

4.2 Simulation Results

When the final result is output, each path determined in each round is shown by outputting a topology path selection diagram. The total output of the three simulations is 76 graphs, because of the limited space, here only random displaying of the distribution under the number and density of various people. The simulation results are shown in Figs. 5 and 6.

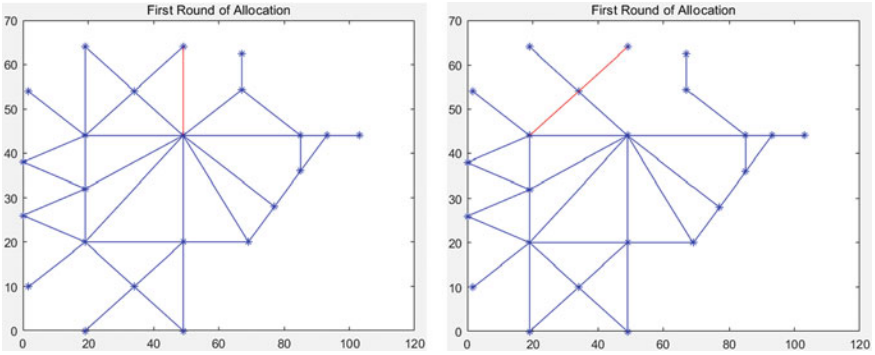

Fig. 5 Second time of allocation in the first round with 180 people in the state of dispersion and the fifth time of allocation in the first round of 180 people in the state of concentration

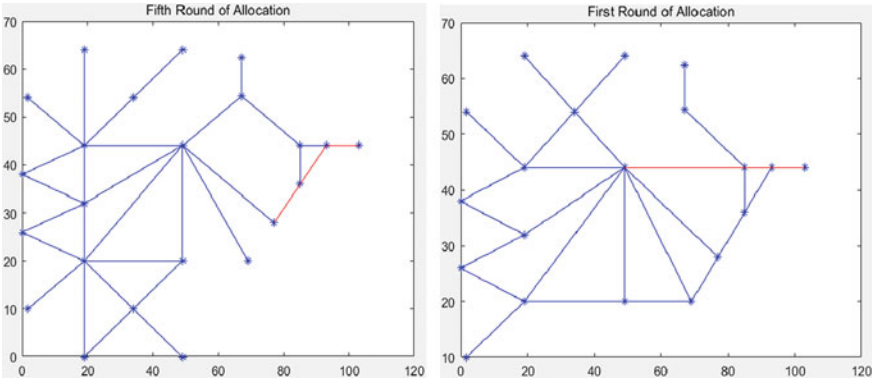

Fig. 6 Second time of allocation in fifth round with 300 people in the state of concentration and the tenth time of allocation in the first round with 450 people in the state of concentration

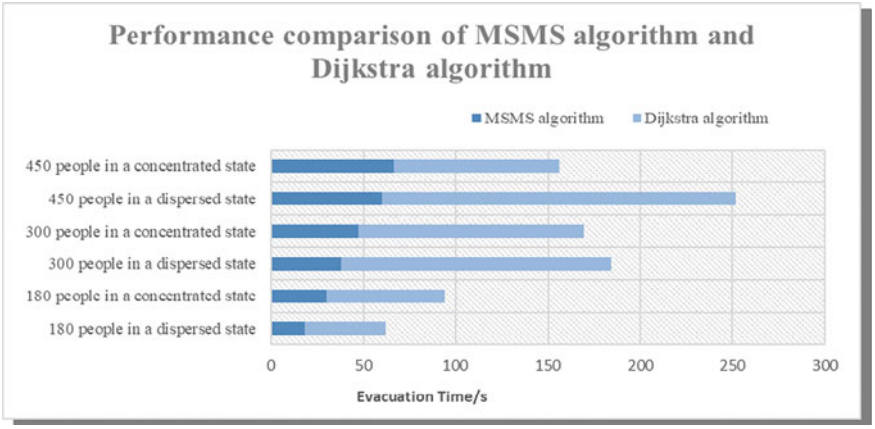

Fig. 7 Performance comparison of MSMS algorithm and Dijkstra algorithm

4.3 Simulation Performance Analysis

Both the improved algorithm and the basic algorithm conducted simulation tests in the MATLAB. And the results are shown in the form of a statistic chart as in Fig. 7 (the improved algorithm is named as MSMS algorithm).

From the chart, we can clearly see that the MSMS algorithm performs much better than the basic Dijkstra algorithm under different conditions. And by computing all the records, which records of MSMS algorithm are, respectively, 66.184 s, 60.000 s, 47.000 s, 37.621 s, 30.000 s and 18.028 s, and the records of Dijkstra algorithm are, respectively, 90.000 s, 191.962 s, 122.127 s, 146.446 s, 64.185 s, and 43.762 s, we concluded that **the performance of MSMS algorithm is improved by 154% than the basic Dijkstra algorithm.**

And by comparing the total evacuation time of the tests, which use the MSMS algorithm, we find that the performance of the MSMS algorithm designed in this paper is more outstanding and excellent in the state of dispersion, so it is more suitable for the evacuation in buildings with people in the state of dispersion.

5 Conclusion and Prospect

This paper improves the Dijkstra algorithm from some aspects. First, the shortest path from each source point to each sink point can be obtained by calling multiple times, and then, the shortest path in the global sense can be obtained by sorting. Second, the paths in the graph all have the corresponding carrying capacity. Therefore, except the basic weight of the edge length, the algorithm also defines the weight value of path capacity according to the actual situation and adopts the method of minimum capacity. Finally, the suitable paths are selected by two modules named the first round

of allocation and the second to N round of allocation, so as to solve the multi-source point to multi-sink point personnel escape route planning problem. At the same time, this paper selects a shopping mall as the modeling object and simulates the actual path selection and output of the algorithm under different personnel distribution, and finally draws the conclusion that the improved algorithm designed in this paper has better performance under the condition of disordered dispersion of personnel distribution.

The current research in this paper focuses on the algorithm improvement of the theoretical model, and fails to add the unstable factors caused by the subjective blindness of the crowd and the environmental interference factors of the integrated buildings of business and residence buildings. Therefore, it is necessary to consider the prediction of crowd behavior model [10] and the simulation of complex commercial and residential environment in the future research to make the algorithm more in line with the needs of real life.

Acknowledgements This work is supported by the following two funds as National Key Research and Development Project of China No. 2017YFC0704100 (entitled New Generation Intelligent Building Platform Techniques) and Foundation of Anhui Jianzhu University, Hefei, China, No. JZ192012.

References

1. Dijkstra, E.W.: A Note on two problems in connection with graphs. Numerische Math. **1**, 269–271 (1959)
2. Gbadamosi, O.A., Aremu, D.R.: Design of a modified Dijkstra's algorithm for finding alternate routes for shortest-path problems with huge costs. In: International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), pp. 1–6, IEEE, Ayobo, Ipaja, Lagos, Nigeria (2020)
3. Deng, Y., Chen, Y., Zhang, Y., et al.: Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. Appl. Soft Comput. **12**(3), 1231–1237 (2012)
4. Fusic S.J., Ramkumar, P., Hariharan, K.: Path planning of robot using modified dijkstra Algorithm. In: National Power Engineering Conference (NPEC), pp. 1–5, IEEE, Madurai (2018)
5. Sedeño-Noda, A., Colebrook, M.: A biobjective Dijkstra algorithm. Eur. J. Oper. Res. **276**(1), 106–118 (2019)
6. Xu, Y., Wang, Z., Zheng, Q., Han, Z.: The application of Dijkstra's algorithm in the intelligent fire evacuation system. In: 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 3–6, IEEE, Nanchang, Jiangxi (2012)
7. Zhang, F., Xu, B., Shao, Z.: Study on optimization of emergency evacuation mode of "Multi Source and Multi Sink" in high-rise apartments. IOP Conf. Ser. Earth Environ. Sci. **25**2(5), 052151(2019)
8. Lin, P., Lo, S.M., Huang, H.C., et al.: On the use of multi-stage time-varying quickest time approach for optimization of evacuation planning. Fire Saf. J. **43**(4), 282–290 (2008)
9. Wolfgang, M., Feldmann, A., Maennel, O., et al.: Building an AS-topology model that captures route diversity. ACM SIGCOMM Comput. Commun. Rev. **36**(4), 195–206 (2006)
10. Rozo, R.K., Arellana, J., Santander-Mercado, A., et al.: Modelling building emergency evacuation plans considering the dynamic behaviour of pedestrians using agent-based simulation. Saf. Sci. **113**, 276–284 (2019)