Computer Life (CPL) ISSN: 1819-4818
DELIVERING QUALITY SCIENCE TO THE WORLD

Evacuation path planning algorithm design

Longtian Fu [1], Yumei Yu [2]

[1]Fuzhou Univercity of international studies and trade, Fuzhou, China

[2]Fujian Normal Univercity, Fuzhou, China

Abstract: When an emergency occurs on campus, students need to be evacuated to a safe area quickly. Dijkstra's algorithm is a classic shortest path algorithm. However, this algorithm has many problems, and the best evacuation path cannot be obtained in the evacuation path planning. For this reason, this paper proposes several improvements to this algorithm's shortcomings, especially adding crowd evacuation distribution schemes. Experiments show that crowd distribution schemes can make orderly evacuation and significantly reduce evacuation time.
Keywords: Path planning, shortest path, Dijkstra algorithm.

---

1. INTRODUCTION

With the continuous development of society, our people are getting richer, more and more concerned about their children's education, and more concerned about their children's safety. However, there have been frequent campus security incidents in recent years. For example, in December 2009, a stomping incident occurred at Yucai Middle School, a private school in Xiangxiang City, killing 12 people and injuring 27 [1]; In 2010, there was a stampede in Xinjiang Aksu Fifth Elementary School, 43 people were injured and 8 were seriously injured [2]; In 2013, a stomping incident occurred in a primary school in Laohekou, Xiangyang, in which 13 people were injured and 6 died [3]. Such campus safety issues have fully exposed the soft underbelly of campus digitalization[4], that is, campus safety issues have not received sufficient attention.

How can we minimize injuries during an emergency? That is to evacuate students as quickly as possible. Xiao-Bing Hu and other scholars found k shortest paths through the ripple expansion algorithm, and simulation experiments proved that the method is very effective [5]; Ma Zhangyi and other scholars introduced the Dijkstra algorithm in sentinel satellites that are widely used to monitor surface deformation for Finding the shortest path to generate the optimal registration image pair. Experiments with this method can achieve the purpose of effective precise registration and make up for the shortcomings of the NESD method [6]. Hu Ruiting and other scholars use the shortest path cache algorithm to find real-time indoor paths Solving this problem has achieved good results[7]; Ni Yudong and other scholars use artificial bee colony algorithm to dynamically plan the path for the robot. Experiments prove that this method only considers the shortest path, but considers road conditions[8]. The above scholars only focused on the shortest path and did not consider other factors.

In this article, while planning the shortest path for the evacuation path, the road's crowd carrying capacity and crowd evacuation plan were considered.

2. Evacuation route design

2.1 Dijkstra algorithm

Dijkstra's algorithm was proposed by the famous Dutch scientist Dijkstra in 1959 [9], so this algorithm can also be called Dijkstra's algorithm. The basic idea of the algorithm is to start from a certain starting point, find other vertices that are connected to the starting point, choose the one with the smallest weight (that is, the edge length between the two vertices), and then use this as the starting point and continue iterating until the end Up to [10].

(1) Weighted graph. Let $G = \{V, E\}$, $V = \{v_1, v_2, ..., v_n\}$ represent all vertices in the graph, and $E = \{e_1, e_2, ..., em\}$ represent the vertices. The edges are the weights mentioned by Dijkstra's algorithm, and eij represents the length of the edges between the vertices v_i and v_j.

(2) Find the shortest path. Let v_1 ($v_1 \in V$) as the starting point and v_k ($v_k \in V$) as the end point. From v_1, find the vertex with the smallest weight in e_{1j}, and use this as the starting point to continue to find the next The vertex with the smallest weight is iterated until the end point v_k is found, and the vertex along the road is the shortest path. The specific implementation process is shown below。

```
Algorithm: Dijkstra algorithm
Input: a graph, starting point Vs and ending point Vt
Output: shortest path sequence
begin
   Input all vertices V in the graph, and the side length E of each connected vertex;
   W initialization;
   Flag = 0; // initialize the vertex with the smallest weight
   While (traverse the vertices in the graph,and Vs! = Vt)
   {
      For (i = 1; i <number of vertices connected to v1; i ++)
      {
         If (w> esi)
         {
           W = esi; // get the smallest weight
           Flag = i; // record vertices with minimum weight
         }
}
Vs = Vflag;
ListA.Add (Vflag); // temporary vertices are in the list
   }
   Output the vertices in ListA; // output the path along the way
end
```

2.2 Improvement based on Dijkstra algorithm

Although the Dijkstra algorithm is a very effective algorithm, its shortcomings are also very obvious, and it is specifically manifested in the following aspects. The improved algorithm is named Dijkstra_Improved.

(1) The shortest path found by Dijkstra's algorithm is not the true shortest path, because the vertices with the shortest edge connected to the starting point are searched each time, so the path found is not the shortest [33].

(2) Dijkstra's algorithm only considers the edge length between vertices. Because this algorithm uses this algorithm to plan the evacuation path of people, the crowd carrying capacity of the path between vertices must be considered, because the width of the path has a great constraint on crowd evacuation . In view of the above-mentioned obvious defects, this paper will improve the algorithm for the above-mentioned defects, so as to be able to adapt to the planning function of the evacuation path. This article has made improvements in the following areas.

(1) Increase the crowd carrying capacity between vertices, and set the undirected weighted graph $G = \{V, E, N\}$, $V = \{v_1, v_2, ..., v_n\}$ to represent all vertices in the graph, and $E = \{e_1, e_2, ..., e_m\}$ represents the length of the edge between the vertices, and $N = \{n_1, n_2, ..., n_m\}$ represents the crowd carrying capacity of the path between the vertices.

(2) Shortest path calculation. Because the "shortest" path obtained by Dijkstra's algorithm is not the shortest path, it is assumed that there are p paths between the start point and the end point. (This can be obtained by using the depth traversal or breadth traversal of the graph. (More details), then the length of each path is the accumulation of the side lengths between the vertices along the path. Assuming that each path has q vertices and q-1 edges, the shortest path calculation is as follows.

$$L_{shortest} = Min(\sum_{i=1}^{q-1} e_i) \tag{1}$$

(3) Evacuation planning. This article needs to plan the evacuation path. Therefore, the more endpoints, the stronger the evacuation capability, and the more evacuation paths there are multiple endpoints, the "shortest path calculation" can be used to obtain the starting point for each endpoint The shortest path, how to distribute the crowd among the various paths is the key point of evacuation planning. Let $V_T = \{v_1, v_2, ..., v_r\}$ denote all termination points (ie, r termination points, each starting point has at most r shortest evacuation paths), and $v_r \in V$. Set the weight $W = \{w_1, w_2, ..., w_r\}$. The starting point is that there are C individuals who need to be evacuated. The weight of each evacuation path and the calculation of the staffing plan are calculated as follows.

$$\omega_i = \sum_{i=1}^{q-1} \frac{n_i}{e_i} \tag{2}$$

$$w_i = \frac{\omega_i}{\sum_{i=1}^{r} \omega_i} \tag{3}$$

$$L_i = w_i * C \tag{4}$$

Formula (2) represents the reciprocal of all side lengths on each path multiplied by the carrying capacity of the crowd and accumulates; Formula (3) represents the normalization process for each evacuation path to obtain the distribution weight value of crowd evacuation Formula (4) represents the number of evacuation people allocated per evacuation route.

3. SIMULATION

According to the experimental environment settings in the previous chapter, this experiment will be simulated in the track and field. In order to complete the experiment more conveniently, the author of this article drew the simulation site in the track and field as shown in Figure 5-2.

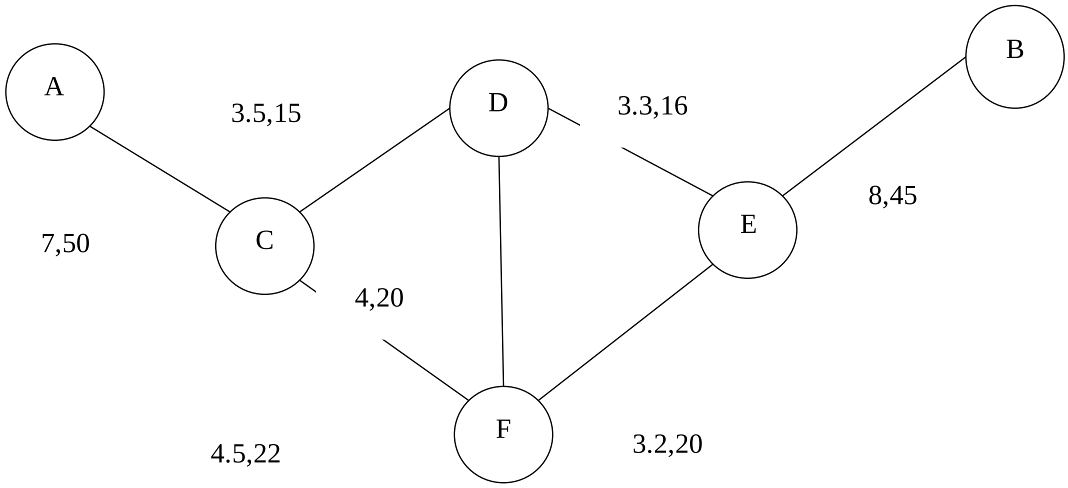

Figure 1 Simulation field map

As shown in Figure 1, AB is the exit (end point). There are crowds at the two vertices of the DF that need to be evacuated. The path length and crowd carrying capacity have been marked in the figure. For example, "7, 50" between the vertices AC indicates edges The length is 7m and the carrying capacity of the crowd is 50 people. The shortest paths according to the Dijkstra algorithm and Dijkstra_Improved algorithm are shown in Table 1.

Table 1 Shortest path comparison table

NO	Item	Dijkstra algorithm	Dijkstra_Improved algorithm
1	D's shortest path to A	DEFCA	DCA
2	D's shortest path to B	DEB	DEB
3	F's shortest path to A	FDCA	FCA
4	F's shortest path to B	FDEB	FEB

It can be seen from Table 1 that the "shortest path" obtained by the Dijkstra_Improved algorithm is the true shortest path, and the number of vertices is also small.

In the sixth set of experiments, 150 people were set at the vertex D. The Dijkstra algorithm does not involve the crowd distribution scheme. It is assumed that the evacuation is free along the "DEFCA" and "DEB" shortest paths (see Table 1). "And" DEB "evacuation, the population distribution scheme is shown in formula (4), and the experimental results are shown in Table 2.

Table 2 Result of the first group of experiments

NO	Item	Dijkstra algorithm	Dijkstra_Improved algorithm
1	Crowd evacuation takes time	33.8s	14.3s

From the experimental results, it can be seen that the Dijkstra_Improved algorithm takes only 14.3 seconds, so it has a clear advantage in crowd evacuation. The Dijkstra algorithm because the "shortest path" obtained is not the true shortest path, which virtually increases the evacuation time consumption; and There is no crowd distribution plan, resulting in a random flow of people, and there is a slight confusion. For example, some people are at a loss as to which exit to follow or follow the flow of people to cause a large crowd flow at an exit, which has led to increased time consumption. In contrast, Dijkstra_Improved algorithm has a crowd distribution scheme, that is, who is going where, the purpose is very clear, and the path is short, so the time consumption is greatly reduced. In the second group of experiments, there are only 8 classes (about 400 students) in the physical education class on the track and field, and the number is limited. Set 200 people at vertex *D* and 200 people at vertex *F* and conduct evacuation at the same time. The experimental results are shown in Table 3.

Table 3 Results of the second group of experiments

NO	Item	Dijkstra algorithm	Dijkstra_Improved algorithm
1	Crowd evacuation takes time	233.5s	42.7s

After this group of experiments was officially launched, when the Dijkstra algorithm was used to evacuate, the scene was more "spectacular" (because there was no crowd distribution scheme, the scene was very chaotic and crowded), which took 233.5 seconds and took a long time; when using the Dijkstra_Improved algorithm to evacuate, two fixed points When the evacuees evacuated, they met at the vertices C and E, and only a short period of "chaos" appeared at the intersection of the two vertices. At other times, the order was orderly, so the time consuming was very short.

In this experiment, the author profoundly appreciates the importance of crowd distribution schemes during evacuation. Without crowd distribution schemes, it is likely to cause congestion, cause panic, and even step on accidents. Therefore, in this paper, a crowd distribution scheme is added when the algorithm is improved, and experiments have proved that it plays a very important role in crowd evacuation.

4. CONCLUSION

In this paper, two sets of experiments are designed. From the experimental results, the improved algorithm Dijkstra_Improved has set up a crowd distribution scheme, which makes the evacuation crowd more orderly and the evacuation time is significantly reduced.

REFERENCES

[1] Zhang Min, He Lijuan. Safeguard mechanism of the Communist Youth League in colleges and universities in responding to violent and terrorist incidents on campus [J]. Journal of Hebei Youth Management Cadre College, 2019, 31 (05): 17-24.

[2] Liu Fuwu. Collaborative Governance of College Campus Security under the Concept of Security Governance [J]. Western Quality Education, 2019, 5 (09): 114-115.

[3] Jia Yufan. Causes and Prevention Mechanisms of Campus Security Incidents [J] .Legal Expo, 2019 (04): 27-29.

[4] Shi Wenhui. Strengthening safety education to prevent campus injuries [J]. Chinese School Health, 2018, 39 (02): 167-169.

[5]Xiao-Bing Hu,Chi Zhang,Gong-Peng Zhang,Ming-Kong Zhang,Hang Li,Mark S. Leeson,Jian-Qin Liao. Finding the k shortest paths by ripple-spreading algorithms[J]. Engineering Applications of Artificial Intelligence,2020,87.

[6] Ma Zhangye, Yue Dongjie, Jiang Mi, Liu Lian. Time series image registration of sentinel TOPS mode based on Dijkstra's shortest path algorithm [J / OL]. Journal of Wuhan University (Information Science Edition): 1-12 [2019 -12-26] .http: //kns.cnki.net/kcms/detail/42.1676.TN.20191205.1028.002.html.

[7] Hu Ruiting, Diao Mingguang. Shortest path caching algorithm for indoor real-time path planning [J]. Electronic Technology and Software Engineering, 2019 (22): 137-139.

[8] Ni Yudong, Li Yuanyuan, Shen Yindong, Fei Xuefang, Song Yangqin. Research on Mobile Robot Path Planning Based on Improved Artificial Bee Colony Algorithm [J]. Journal of Hefei University of Technology (Natural Science), 2019, 42 (11): 1569 -1575.
[9] Wang Teqi, Xie Yaqin. Design and Implementation of Campus Navigation System Based on Dijkstra Algorithm [J] .Communications Technology, 2019,52 (08): 1937-1943.
[10] Shi Jia. Optimization Problem of Disneyland Based on M / M / s / K Model and Dijkstra Algorithm [J]. Education and Teaching Forum, 2019(32): 256-258.
[11] Luo Wei, Zhang Xiaorong, Zhang Tian. Optimization of medical logistics distribution network based on improved Dijkstra algorithm [J]. Value Engineering, 2019, 38 (21): 121-122.