Application of Dijkstra's Algorithm in the Smart Exit Sign

Jehyun Cho[a], Ghang Lee[a], Jongsung Won[a] and Eunseo Ryu[a]

[a]Dept. of Architectural Engineering, University of Yonsei, South Korea
E-mail: jkun86@naver.com, glee@yonsei.ac.kr, quietman111@gmail.com, eunseo4773@naver.com

Abstract

Previous studies on automated fire-egress guidance systems have focused on providing the shortest path information from a specific person at a certain point to the closest exit mostly using a mobile device. This study aims to develop a Smart Exit Sign system that can detect dangerous areas in real time and direct evacuees to the shortest safe evacuation path by dynamically changing the direction signs to the safe egress. The challenge was to provide the shortest safe egress to any evacuees at any point. We have developed a sensor network and algorithm that could exclude unsafe paths and calculate the shortest safe path from multiple starting points to multiple exit points based on Dijkstra's algorithm—the most commonly used algorithm for finding the shortest path. The validity of the proposed system was tested through simulations of test cases.

Keywords -
Shortest path algorithm; Evacuation system; Exit sign

1 Introduction

Navigating the indoor spaces of large and complex buildings such as shopping malls is challenging. This makes building occupants often disoriented while they explore a building. Yet, this issue is more problematic during an emergent situation such as fires rather than general situations. During an emergent situation, exit signs may be helpful. However, traditional exit signs in buildings usually have a fixed direction sign towards exits and may direct evacuators to dangerous paths where a fire has broken out. Consequently, in this type of situation, individuals cannot dynamically change directions reflecting the fire hazard. In order to respond to this issue, we have developed a smart exit sign that can dynamically change direction signs and guide evacuators to safe evacuation paths. This paper presents an Automated Direction Setting Algorithm (ADSA), which was developed as part of the Smart Exit Sign system. The ADSA was designed to dynamically change the direction signs on a smart exit sign—reflecting the fire situations—and indicate the proper directions towards safer evacuation paths.

Previous research has studied evacuation path based on the shortest path algorithms such as Dijkstra's algorithm, Floyd-Warshall Algorithm, and A* algorithm. The main goals of those studies were to find the shortest evacuation path and guide a certain person to the path by showing the path on a mobile phone or on other systems. However, in an emergent case, such as fire, most evacuees may not have or may not be able to download the smart device application that is required to visualize the shortest evacuation path on their mobile devices.

The eventual goal of this study is to develop a Smart Exit Sign system that detects unsafe areas and calculate the shortest safe path in real time and dynamically change its direction sign toward the shortest safe path. In the first step, this paper presents an algorithm and a simulator that can calculate and visualize a network of smart exit signs, which can guide any evacuees at any point in a building to the nearest exit via a safe route. The algorithm, developed based on Dijkstra's algorithm, which is one of the famous shortest path algorithms, calculates safe evacuation paths from multiple starting points while directing individuals toward the nearest exits.

This study was conducted in the following order. First, through review of related literature, we determined the characteristics needed to make the algorithm for the system effective. Then, we developed an algorithm and a simulator. Finally, we tested the applicability and validity of the algorithm through simulations of test cases.

2 Literature Review

Many studies have been conducted to develop systems that could provide indoor navigation information in an unknown space during an emergent situation.

In a study conducted by Kobes et al., it examined how people determined evacuation paths, during fire related situations, through a series of tests that were conducted in a hotel building at night [1]. According to the research, 56.3% of people determined evacuation paths depending on exit signs when there was no smoke present; however, 81.8% determined an evacuation path depending on exit signs when their visibility was impaired due to smoke. These results show that the role of exit signs is of the utmost importance when individuals are put in adverse situations that require an exit strategy.

However, because the current exit signs installed in buildings have fixed directions, there is a possibility that the signs can lead the evacuators to dangerous area according to the study conducted by Choi [2]. Through a perception test, Choi discovered that the ambiguity of directions on exit signs could make it difficult for evacuees to recognize the correct direction towards an exit [2]. On the other hand, when animated direction images were used on exit signs, the number of people who followed the correct evacuation path doubled.

Jang et al. also argued that the evacuation guidance system with a fixed direction image could lead evacuees to dangerous areas by illustrating a simple case [3].

In regards to exit signs, Kim et al. proposed an ‘artificial intelligent directional escape light system’ based on the the Floyd-Warshall algorithm [4]. This system was designed to suggest directions for the nearest exits and also to indicate dangerous points where a fire has occurred, but has not been implemented.

Similarly, another group of researchers, Kim et al. [5], proposed a similar smart exit sign system. The system was composed of wireless sensors, had used Dijkstra’s algorithm, and was tested through simulations in virtual space. However, in the Kim’s simulation test, exit signs at several points could not find the correct direction towards the nearest exit.

3 Automated Direction Setting Algorithm

We developed the Automated Direction Setting Algorithm (ADSA) for the smart exit sign system, which offers evacuation direction for a number of evacuees at any location in a building on the basis of Dijkstra’s algorithm. Dijkstra`s algorithm is a representative of the shortest path algorithm. It is a type of *greedy* algorithm, which is used to search for the maximum value or the minimum value.

Dijkstra`s algorithm iteratively calculates the distance between a starting point to the other points in the network in turn; and, depending on which path is used to travel from the starting point to another point, the distance will vary. The starting point has the distance value of 0. The algorithm replaces the distance value stored at each point (node) from the starting point to each point with a new value if a shorter path is found.

Dijkstra’s algorithm has been used in various fields, such as Google Maps, navigation systems for automobiles, traffic engineering, robotics, and industrial engineering. Compared to Floyd algorithm, which is another popular shortest path algorithm, Dijkstra`s algorithm is better suited for calculating the shortest path in a huge, complex building because the time complexity of Dijkstra`s algorithm is $O(n^2)$ while the time complexity of Floyd algorithm is $O(n^3)$. It means that the amount of time taken by Dijkstra`s algorithm is shorter than that of Floyd algorithm as input size goes higher.

3.1 Additional Features of ADSA

There are two main differences between plain Dijkstra`s algorithm and ADSA for the smart exit sign system. The following subsections describe these differences in detail.

3.1.1 Exits as Starting Points

Basic concept of Dijkstra’s algorithm is checking the shortest paths from the single starting point to all the other points. However, unlike general Dijkstra’s algorithm, ADSA assumes that the exits are the starting points for the calculations of the evacuation paths. The reason being is because Dijkstra’s algorithm does not distinguish the exit from other points. Thus, by specifying exits as the starting point—the exits get the distance value of 0 and the modified algorithm—ADSA automatically recognizes the point with 0 value as an exit. Moreover, ADSA points the direction sign to the exit with the smallest distance value from each point. If a new exit closer from a specific point is found, the direction sign at the point is changed towards the new exit.

For example, Figures 1 to 5 show a topological representation of a building plan, which has one exit and Dijstra distance values at each point. In the plan, if we randomly pick point A as the starting point as shown in Figure 1, we know that the shortest distance from points A to C (D_{A-C}) is 22. However, we do not know which path is the shortest path and, thus, cannot show which direction evacuees at point A should go: from point A to point B or to point D.

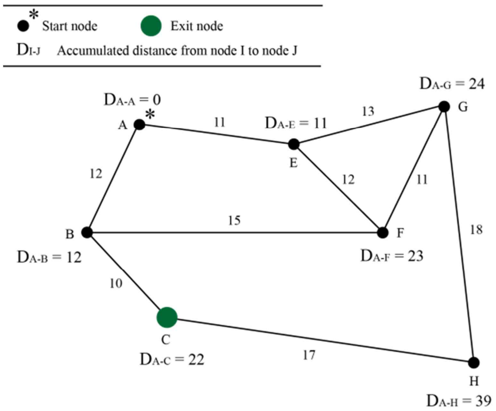


Figure 1. An example of specifying a random point as the starting point

On the other hand, if we assume that the exit (point C) in a plan is the starting point as shown in Figure 2, we can determine the direction to the shortest path by comparing the accumulated distance values from the exit (point C) to each neighbouring point: points B and E in this example. Since point B has a smaller accumulated distance value than point E, our algorithm leads evacuees at point A to point B.

However, normally, there exist multiple exits in buildings. So, in order to find the nearest exits from each point, the algorithm should be able to compare the distance from each point to multiple exits as shown in Figure 3. The algorithm chooses the smaller values. Figure 4 shows the final calculation results.

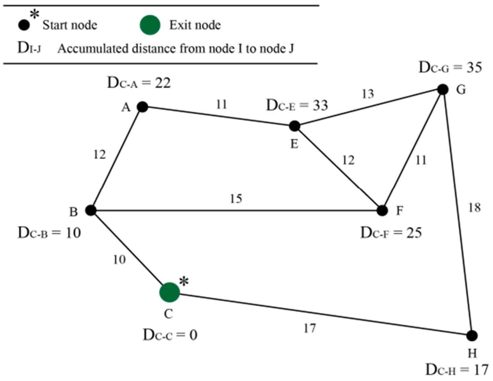


Figure 2. An example of specifying an exit as the starting point

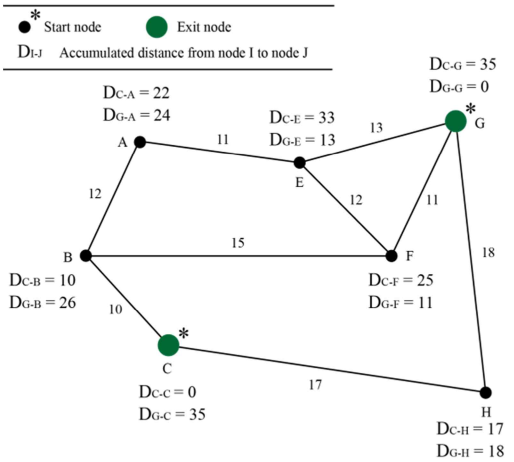


Figure 3. Path calculation between points C and G

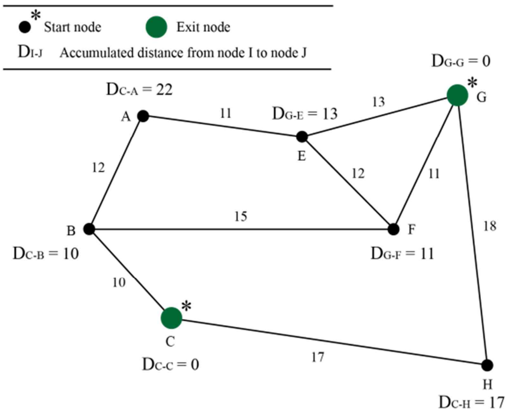


Figure 4. The final calculation results

3.1.2 Reconfiguration of a Path Network

Once the one of the signs which comprise the whole system notices any fire situations by heat detectors, smoke detectors or missed signal from the adjacent sign of breaking because of the fire, Originally established networks in a building must be reflect the current situation before suggest the direction change.

The ASDA reconfigures the path network and recalculates the directions from each point to the shortest safe path by excluding dangerous paths when a fire breaks out. Some corridor will be set as impassable according to the location of the fire. Figure 5 shows an example where a fire occurs at point F. The algorithm eliminated point F and all of the paths that are connected to point F and recalculated the distance.

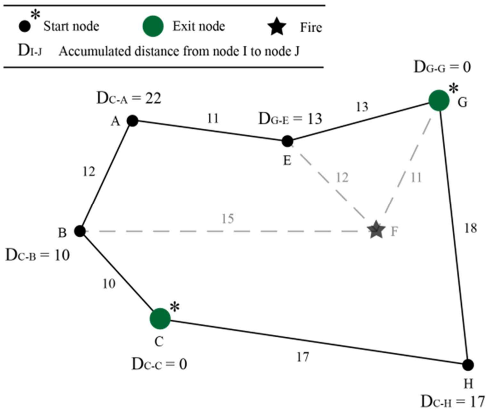


Figure 5. Recalculation of distances after eliminating dangerous paths

3.2 Implementation of ADSA in a Simulator

We developed a Smart Exit Sign simulator using the ADSA. The simulator is operated in the following three steps (Figure 6).

The first step is to extract a topological configuration of a building plan composed of nodes and lines from a drawing or from a Building Information Model (BIM). Particularly, the nodes are classified into four types: “exit,” ”intersection,” “dead-end,” and “exit sign.” The purpose of distinguishing “exit” from the other node types is to run the ADSA by using exits as starting points. And, “dead-end” is defined only for the graphical purpose in the simulation software.

Secondly, users specify the actual distance between any two selected nodes so that the simulator can place a small icon that represent an exit sign at every 15m, following the Korean national emergency management law that says exit signs must be placed at every 15m or less (National Emergency Management Agency, 2012, South Korea). This step does not have to do with the calculation of the shortest safe path, but it is an important step to show where Smart Exit Signs should be placed. The distances between all the other nodes can be automatically calculated.

Finally, each smart exit sign’s distance is calculated from to the nearest “exit.” Then, each exit sign automatically changes the direction sign to show the shortest and safest path. If a fire occurs, the third step is repeated after eliminating the paths where a fire breaks out.

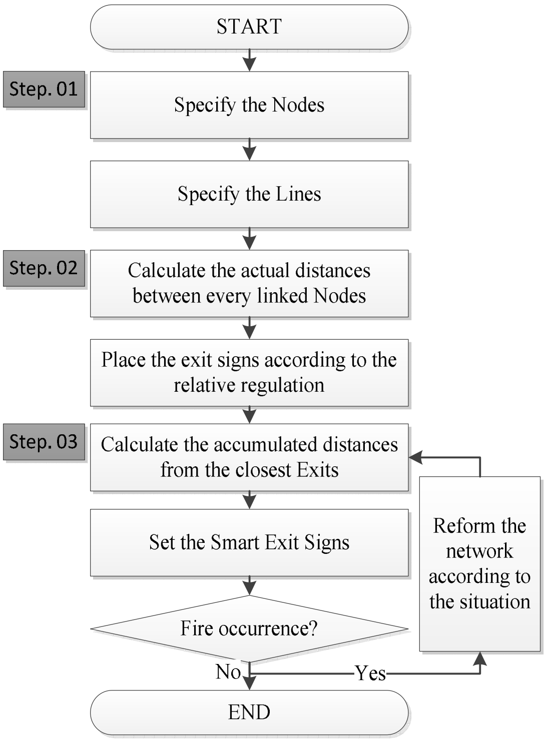

START
Step. 01
Specify the Nodes
Specify the Lines
Step. 02
Calculate the actual distances between every linked Nodes
Place the exit signs according to the relative regulation
Step. 03
Calculate the accumulated distances from the closest Exits
Set the Smart Exit Signs
Reform the network according to the situation
Fire occurrence?
No
Yes
END

Figure 6. A 3-Step Operation Process of the Smart Exit Sign Simulator

The simulation program was developed using C# computer programming language. Table 1 displays the icons used in the simulator.

Table 1. Simulator Icons

Menu		Function	Result
Node	Exit	Specify 'Exit's	●
	Intersection	Specify 'Intersection's	T
	Dead-end	Specify 'Dead-end's	🟥
Line	Corridor	Specify 'Line's	—
	Parameter	Save the actual length	value
AP		Arrange the Signs	➔
Fire		Simulate the fire	🔥

4 Simulation Results

We ran simulation tests to check the applicability and validity of ADSA through two test cases

The first case was a hypothetical maze map which was a good sample to test, debug, and refine the algorithm (Figure 7).

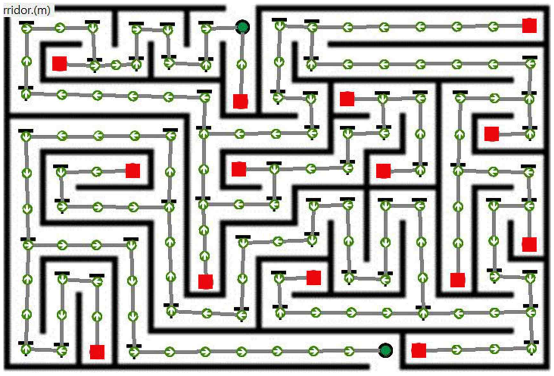
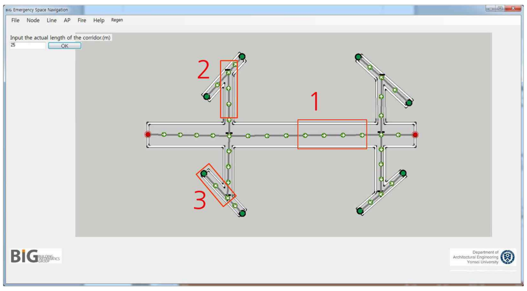


Figure 7. Arrangement of exit signs in a maze

. The second test case was a subway station in Seoul, South Korea. The plan had 15 separated corridors, 6 intersections, and 8 exits.



Input the actual length of the corridor.(m) 25

Figure 8. Three test case scenarios

We manually checked whether the direction of each exit sign in the simulator pointed towards the nearest and safest exit.
First, it was confirmed that the simulator automatically set the initial directions of the exit signs towards the nearest and safest exit. Then, we tested three different fire scenarios (fire on a corridor, fire on an intersection, and fire on an exit) and checked to see if the signs were correctly redirected to the new safe path (Figure 8).

Figures 9 to 11 show a zoomed-in view of three cases and redirection results: fire on a corridor (Figure 9), fire at an intersection (Figure 10), and fire at an exit (Figure 11), respectively. All three cases showed the correct changes.

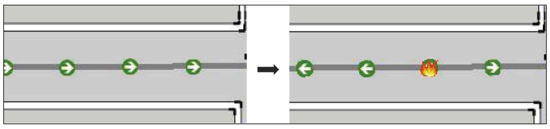


Figure 9. Fire on a corridor and the redirection result

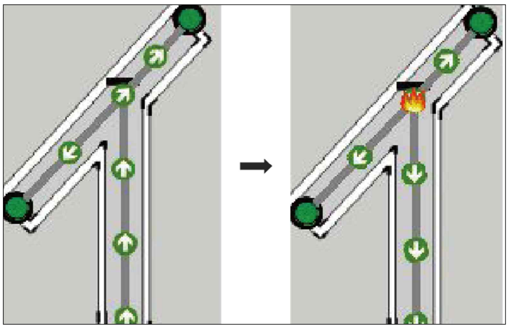


Figure 10. Fire at the intersection and the redirection result

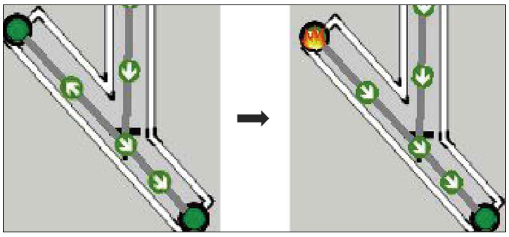


Figure 11. Fire at an exit and the redirection result

Through the tests, we confirmed that the simulator could correctly show the shortest safe path and redirect the exit signs avoiding impassable corridors.

5 Conclusions

This paper presents the ADSA, which was developed as a part of the Smart Exit Sign system based on Dijkstra`s algorithm. ADSA finds and leads evacuees at any point in a building to safe exits. To simplify the calculation of the direction from multiple starting points to multiple exits, ADSA set the exits as the starting point unlike the traditional application of Dijkstra's algorithm. ADSA could find the direction from a point to the nearest exit simply by comparing the accumulated distance values of neighbouring points. The algorithm directs the exit sign towards the neighbouring point with a smallest accumulated distance value. If a fire occurs, the direction of exit signs is dynamically recalculated in real time. Through test cases, we confirmed that ADSA could be applied in an actual Smart Exit Sign system. However, our current system only considered distances. The system can be improved by considering other influential factors in a real fire situation, such as spreading direction of fire, smoke, and crowdedness. We intend to reflect these factors to our system in later studies. Also, through additional test cases, we plan to rigorously test whether they are any exceptional cases and check the validity and applicability of the Smart Exit Sign system. Development and test of an actual Smart Exit Sign system in a real building is another future intent.

6 Acknowledgement

This research has been supported by the Ministry of Knowledge Economy, Korea, under the national HRD support program for convergence information technology supervised by the National IT Industry Promotion Agency (NIPA-2010-C6150-1001-0013).

7 References

[1] Kobes M., Helsloot I., de Vries B., Post J. G., Oberijé N., and Groenewegen K., Way finding during fire evacuation; an analysis of unannounced fire drills in a hotel at night, Building and Environment, 45 (3):537-548, 2010.

[2] Choi J., "A study of the intelligent evacuee guidance of pictogram," Master, Graduate School of Urban Sciences, University of Seoul Seoul, South Korea, 2013.

[3] Jang J.-S., Kong I.-C., and Rie D.-H., A study for optimal evacuation simulation by artificial intelligence evacuation guidance application, *Journal of the Korean Society of Safety*, 28 (3):118-122, 2013.

[4] Kim D.-O., Mun H.-W., Lee K.-Y., Kim D.-W., Gil H.-J., Kim H.-K., and Chung Y.-S., The development of the escape light control system, *Journal of the Korean Institute of Illuminating and Electrical Installation Engineers*, 23 (6):52-58, 2009.

[5] Kim Y.-W., Boo J.-P., and Kim D.-H., A scheme of active refuge guide based on Dijkstra`s algorithm, *Korean Multimedia Society Spring Conference*, pages 341-345, Daegu, South Korea, 2008.