Degree Thesis for M. Eng.

Load-balanced route optimization method for accident aboard a ship

Supervisor : Prof. Yang-Ick Joo

February 2018

Department of Electrical and Electronics Engineering

The Graduate School of Korea Maritime and Ocean University

Ryul Kim

Load-balanced route optimization method for accident aboard a ship February 2018 Ryul Kim

Degree Thesis for M. Eng.

Load-balanced route optimization method for accident aboard a ship

Supervisor : Prof. Yang-Ick Joo

February 2018

Department of Electrical and Electronics Engineering

The Graduate School of Korea Maritime and Ocean University

Ryul Kim

Thesis submitted by Ryul Kim in fulfillment of requirement for the degree of M. Eng.

Committee Chairman : D. Eng. Tae-In Jeon 印

Committee Member : D. Eng. Dong-Hoan Seo 印

Committee Member : D. Eng. Yang-Ick Joo 印

December 2017

Department of Electrical and Electronics Engineering

The Graduate School of Korea Maritime and Ocean University

Ryul Kim

Contents

List of Table

List of Figures

Load-balanced route optimization method for accident aboard a ship

by Ryul Kim

Department of *Electrical and Electronics Engineering*
Graduate School of Korea Maritime and Ocean University
Busan, Republic of Korea

Abstract

An emergency evacuation system is a system that helps people in the space to evacuate safely and quickly from emergencies in the event of an emergency. Such systems are essential as the size of vessels becomes larger and more complex. However, current emergency evacuation systems play only a limited role. For example, evacuation route guidance through placement of real human resources or evacuation route such as direction of emergency exit point which is pointed in one direction only in one place. Relying on human subjective judgment in a dangerous situation can be quite dangerous, and emergency lights and escape routes that always point in the same direction are not able to deal flexibly with risk factors and can expose the public to danger. Furthermore, due to the nature of the ship structure, the initial response is important as the rescue time is delayed rather than the land accident.

Therefore, emergency evacuation systems should be more intelligent in increasingly complicated and larger structures, and should be able to quickly identify information on the surrounding situation and suggest an optimal evacuation route. In particular, it is not possible to exclude the possibility that dangerous elements may spread or become dangerous areas in the route where evacuees are passing.

Therefore, there is a need for a system that predicts and responds to the near future through sufficient modeling of risk factors. Among various risk factors, risk factors such as fire, smoke, and isolation can be sufficiently collected by using sensors or image processing devices. However, in the case of bottlenecks, it is essential to model the density of the population at the current node, the direction in which people at that location will evacuate, and whether the path of the selected path will accommodate the incoming population. Therefore, we propose a bottleneck modeling method and load balancing based on disaster situation in this paper. The proposed performance is verified by computer simulation.

KEY WORDS: Search algorithm; Route optimization; Bottleneck; Load balancing

Chapter 1. Introduction

1.1 Research background

Due to the nature of the ship accident, the distance between the point of the accident and the rescue resource is long, and it takes long time to approach. As a result, accident response is relatively delayed to land accident. Therefore, the International Maritime Organization (IMO) has prepared the Maritime Safety Committee / Circulation 1238 (IMO MSC / Circ. 1238), a provision for the escape of passengers in emergency situations in the design of passenger ships To ensure the safety of passengers[1], they are required to satisfy the applicable regulations. In addition, the size of existing vessels, such as cruise vessels, is becoming larger and more complex in the event of ship accidents. Therefore, there is a need for technology to ensure passenger safety in case of ship disaster.

However, emergency evacuation systems currently applied to ships and onshore buildings are not interoperable and are not intuitive in determining the current location and evacuation route of passengers[2]. In addition, there is no proper response to the accident situation, and it depends only on manpower.

1.2 Research Trends

In China, IoT based emergency evacuation system design method and Dijkstra algorithm based evacuation system have been developed. In Japan, researchers developed an indoor evacuation guidance service system through mobile phone. In the United Kingdom and Canada, researchers have developed technologies to track the number and location of people in the building using thermal imaging technology and RFID tags in real time[3-4]. However, considering the unpredictable characteristics of the risk factors, it is difficult to simply detect the location of a evacuee or to transmit fragmentary evacuation guidance message in case of a disaster situation.

Therefore, in the field of disaster prevention and disaster prevention such as safety engineering and architectural engineering for the unpredictable risk factors, the influence and characteristics of the risk factors in the disaster situation have been analyzed and modeled. In the United States, modeling and optimization studies were conducted for emergency evacuation considering the migration characteristics of evacuees. Numerous university research teams in Korea have modeled and tested various risk factors for emergencies and ship disaster situations. These studies have analyzed the impacts and characteristics of various risk factors in disaster situations, which can lead to system requirements to respond to the risk factors of the ship disaster environment.

In the ICT field, we developed an optimal path derivation method based on sensor network and artificial intelligence algorithm. The university team in England designed a distributed decision support system based on sensor network for building evacuation and simulated the multi - layered building environment. The university team in Taiwan proposed an adaptive evacuation system based on sensor network and designed a technique to reduce the evacuation time considering the bottleneck in the evacuation route[5-6].

The study on modeling and simulation of disaster situations in disaster prevention disaster area has been conducted to verify the effects, characteristics analysis, and modeling of disaster environment, but it does not provide the necessary countermeasures in case of actual disaster situation. In addition, the artificial intelligence-based optimal evacuation route search algorithm in the ICT field does not take into account various practical risk factors due to insufficient consideration of disaster situation. Therefore, it is necessary to verify the effectiveness of the proposed method based on the performance evaluation based on actual disaster environment[7-9].

1.3 Research Necessity

Unlike a land-based accident situation, the ship is surrounded by the sea, so the safety of the external environment must be taken into consideration. Also, since the time required for rescue work is relatively long, internal evacuation should be performed more quickly. Therefore, an appropriate route search algorithm based on technical stability is needed for the specificity of the ship disaster situation.

1.4 Research Summary

Various evacuation algorithms based on artificial intelligence have been proposed in the field of ICT until now, but they have not considered various risk factors due to insufficient consideration of actual disaster situation. Therefore, we model various risk factors for route optimization in emergency evacuation, and propose a load - balancing path optimization method considering the behavior characteristics and bottleneck of evacuees.

Chapter 2. Related Theory and Research

2.1 Searching Algorithm

2.1.1 State Space and Search

A state space is a set of all states that can be reached from an initial state in a problem-solving process, or a set of all states that are likely to be a solution to a problem. The state here refers to the state of the problem world at a particular point in time, and the world collectively refers to the objects included in the problem and their situation.

In the state space, there are an initial state and a goal state. The initial state literally means a starting state of a given point in time, and the target state means a desired final state in a problem.

Search means the process of defining such a state space or finding the optimal solution in the space. **Fig. 1** Sudoku game is also an example of search, and the existing Traveling Salesman Problem (TSP) algorithm is also an example of searching for the optimal solution in the proposed state space.

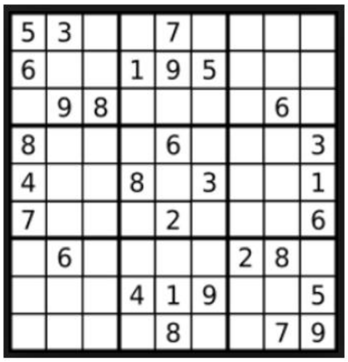


Fig 1. Example of search (sudoku & TSP algorithm)

2.1.2 Blind Search

The blind search is a method of searching the solution by gradually generating the state space graph according to a predetermined order. It is characterized by high accuracy although it takes a long calculation time by searching and calculating all the nodes.

A graph showing changes in the state space according to the search process is called a state space graph. Representative examples of blind search include a depth first search algorithm and a breadth first search algorithm.

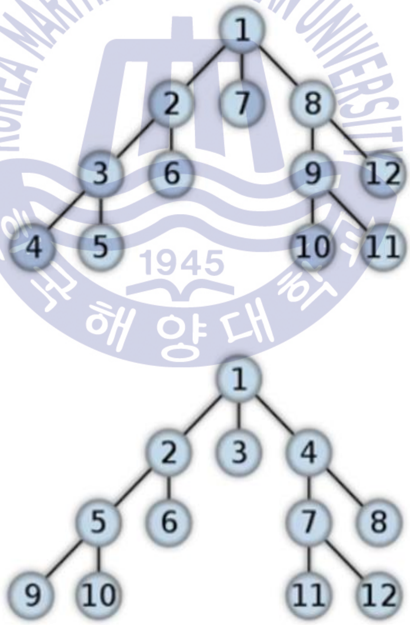


Fig 2. Depth first search and Breadth first search algorithms

The depth-first search algorithm searches for a certain vertex and then searches for a vertex with a higher priority connected to the previous vertex.And returns to the previous state when it can no longer be searched. It is an algorithm that iteratively searches all nodes through the above process. On the other hand, the width-first search algorithm searches for a certain vertex, searches for other vertices connected to the vertex, and sets the vertex in order. And, it is a method to search sequentially from the closest vertices. The retrieved results are stored according to the queue structure, and the tree structure can be designed according to the same depth / level.

According to the characteristics of finding the optimal escape route in case of disaster situation of this study, blind search has a limitation of direct application. However, the backtracking features of depth - first search and the design features of the tree structure of width - first search are very helpful in designing the proposed state space.

2.1.3 Heuristic Search

The heuristic search is a method to select the most promising path from the various possibilities to reach the goal by the evaluation function or the like when the calculation order is not determined uniquely according to the unspecific state space.

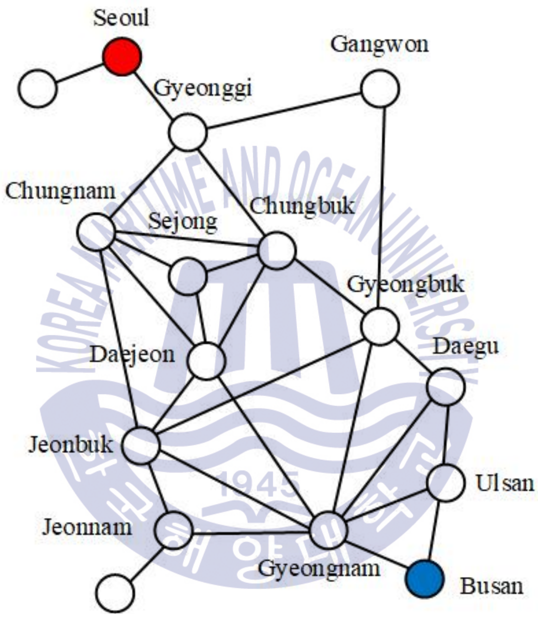


Fig 3. Example of heuristic search

Assuming that the shortest path is searched from Seoul to Busan in the example of **Fig. 3**, the blind search calculates the value for the shortest distance by directly searching each node (city), In the case of heuristic search, it has an additional evaluation function called the shortest distance from Seoul to Busan to enhance search efficiency.

2.1.4 $A*$ Algorithm

A * algorithm is a graph / tree based search algorithm for finding the shortest path from a given starting point to a target point[10]. The algorithm searches for nodes in order based on a "heuristic estimate($h(n)$)" that estimates the best path from each node to the target. This feature is similar to the breadth first search method of blind search. However, unlike the width-first search method, the search is not performed in the direction in which the estimated value increases except for a case where the search is based on the corresponding estimated value. To understand these estimates, we need to define an evaluation function for each node. The formula of the evaluation function($f(n)$) is shown below.

$$f(n) = g(n) + h(n) \tag{2.1}$$

$g(n)$: The path cost from the originating node to node n

$h(n)$: Estimated path cost from node n to target node n

$g(n)$ represents the cost from the starting point to the current node, and $h(n)$ represents the estimated value from the current node to the target node as mentioned above. However, the remaining cost of $h(n)$ defines a heuristic function h (n) corresponding to $\hat{h}(n)$, since accurate prediction is impossible. Therefore, the estimated total cost via node n is shown below.

$$\hat{f}(n) = g(n) + \hat{h}(n) \tag{2.2}$$

2.1.4.1 Operation Process

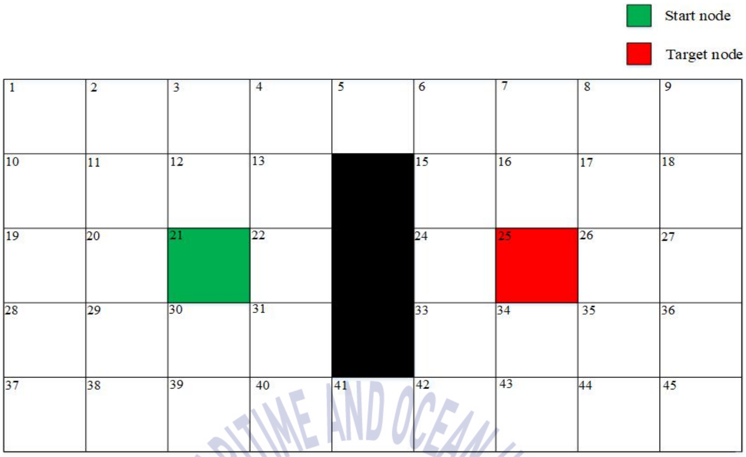


Fig 4. Operation of $A*$ algorithm in step 1

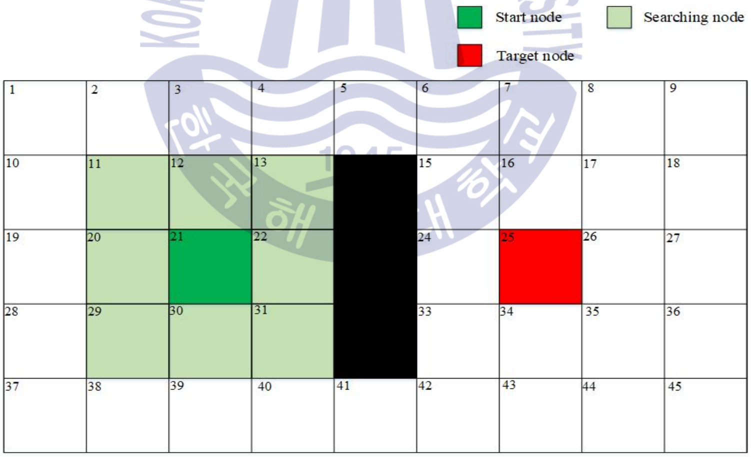


Fig 5. Operation of $A*$ algorithm in step 2

Fig. 4 shows the node set of 9 * 5 size. The distance between each node is assumed to be 10 for convenience. Nodes 14, 23, and 32 are not allowed to pass and distance values are set to infinity. When the node 21 is the start node and the node 25 is the target node, the operation principle of the algorithm is as follows.

(1) Define the evaluation function $f(21)$

(2) $f(21) = g(21) + h(21) = 0 + 40 = 40$

(3) When the evaluation function of the start node is calculated, the evaluation function of the neighbor node is searched in **Fig 5**.

(4) $f(11) = g(11) + h(11) = 10\sqrt{2} + 10\sqrt{26} \fallingdotseq 65.13$. Calculate $f(12)$, $f(13)$, $f(20)$, $f(22)$, $f(29)$, $f(30)$, $f(31)$ in ths way.

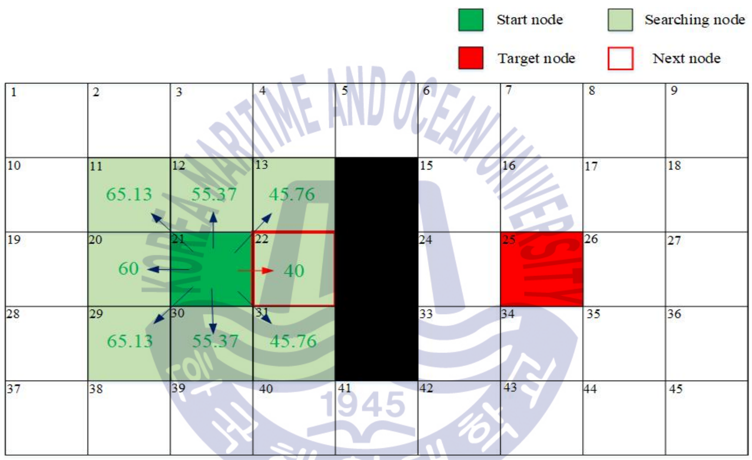


Fig 6. Operation of A^* algorithm in step 3

(5) **Fig 6.** compares each evaluation function and searches for the node with the minimum value.

(6) However, since the node that can compare the evaluation function no longer exists after the node 22, it selects the node 13 or 31 which is less in the next evaluation function. For convenience, select node 13.

(7) Repeat steps (4) and (5) above.

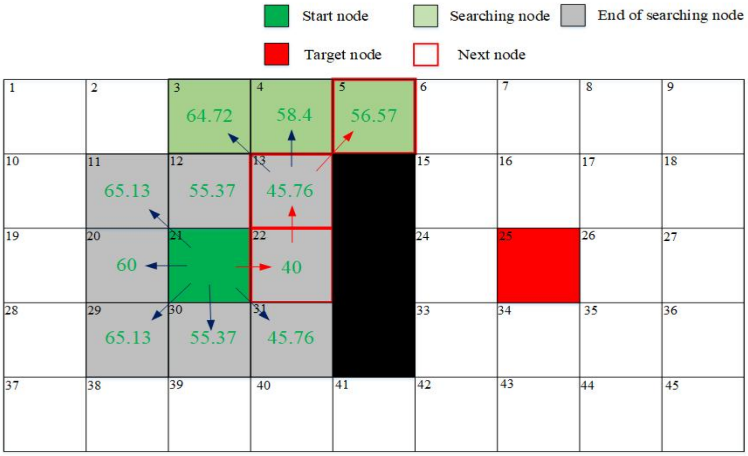


Fig 7. Operation of A^* algorithm in step 4

However, the reason why node 5 is selected instead of node 12 in node 13 is because the search order is determined according to the priority of the evaluation function of the algorithm.

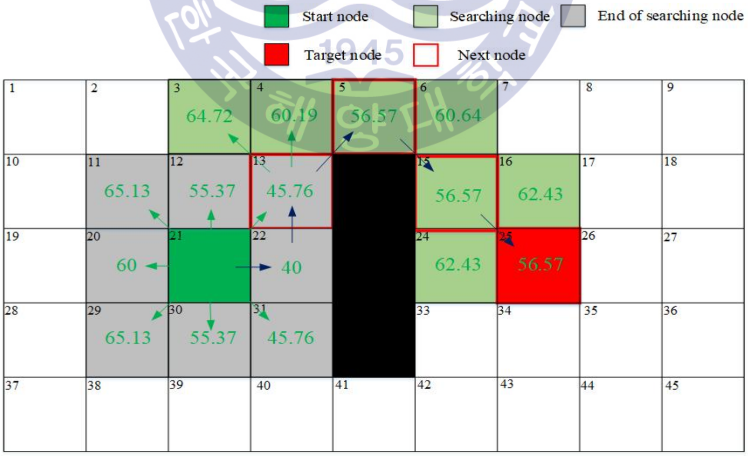


Fig 8. Operation of A^* algorithm in step 5

(8) The search order is 21 → 22 → 13 → 5 → 15 → 25, but the node 22 is omitted in the final search order because it is possible to move from

21 to 13 directly.

(9) This process continues until reaching the target node.

2.2 Searching System

2.2.1 Feeling Factor

Generally, when a disaster occurs, there will be no risk factors, and if you are familiar with the inside of the structure and only a few people escape, you will be able to escape quickly. However, it is very difficult to find an optimal route if you have a complicated structure like inside a large building, or if it is a narrow and complex passage like the inside of a ship. Considering the factors that can interfere with the path in such an environment, even if the same distance travels, the distance actually experienced by the person in consideration of the actual physical distance and the risk factors will be different. This is expressed as a feeling factor (α), which expresses the extent to which movement can be influenced by factors such as flooding, fire and obstacles in the area[11].

The weighted distance (D_w) is the distance that the evacuees actually senses. The larger the value, the more the distance travels the actual distance than the actual physical distance. This is expressed as a product of the actual distance value (D) and the feeling factor (α), as follows.

$$D_{w,r} = \sum_m d_m \cdot \alpha_m = \sum_m d_m \cdot \frac{\sum_{n=1}^{N} W_{n,m} \times O_{n,m}}{N_m}$$

$$, \ where\ N_m = \max(1, \sum_{n=1}^{N} O_{n,m}) \tag{2.3}$$

Here, $O_{n,m}$ is set to 1 when considering the n-th risk element in the m-th partial path, and set to 0 otherwise.

N_m is set to the number of risk elements in the m-th partial path or to 1. The weight (W_n) is set according to the risk level or the degree of difficulty of the risk, and each risk factor is denoted by subscript n, so that the fire can be defined as W_f, the submersion as W_w, and the slope as W_s.

The more various factors are taken into account, the more precise the feeling factor is, and how much of the risk factor is viewed by how much of each factor weighted. The weight of each element can be reflected through the actual measurement or the disaster related simulator. The measurement of the value is beyond the scope of this study. The optimum evacuation path considering weighted distance is selected as the path with the smallest value among all the paths as in Equation (2.4).

$$Optimal\, path\, I = \arg\min_{r \in R} \overline{D}_{w,r} \tag{2.4}$$

2.2.2 Risk Predicted Value

In the case of a route with risk factors, there is a time difference between the optimal route decision point and the movement condition of the opponent when the route is represented only by the feeling factor, and during the time interval. For example, it can be assumed that the evacuee has an explosion due to a risk factor near the fire point during the escape, or that the fire is spreading more and more, affecting one of the currently escaped escape routes. Therefore, the evacuation route decision using only the situation at the time of evacuation can not be considered as the optimal route selection without considering the location, speed and potential risk factors of the evacuee. The expected risk in the near future is defined as the expected risk, and it is dealt with the risk situation that can change in real time considering the optimal path induction algorithm. We divide the m-th partial path (d_m) from the node where the risk has occurred to the next node and the speed (v) at which the n-th risk element is spread to define the time to guarantee the safety of the path[11].

$$t_{s,m} = \min\left(\frac{d_m}{v_1}, \frac{d_m}{v_2}, \cdots, \frac{d_m}{v_N}\right) \tag{2.5}$$

t_s decreases with time, and the risk prediction coefficient ($\overline{Ri}$) can be obtained through t_s. If t_s is large, $\overline{Ri}$ has a value close to “0“ because there is a lot of reflection time of risk factor, and $\overline{Ri}$ has a value close to “1“ when t_s is small.

However, the scope of the $\overline{Ri}$ is not limited to “1“ or less, and it should have a value of “1“ or more if the expected degree of damage due to the risk factor is very large. A device is provided that can reflect the risk level that can be dynamically changed separately from the static weight (W_n) of the risk factors considered in a general situation. Since the method of determining the risk prediction coefficient also goes beyond the scope of this paper, we assume that each risk prediction value is a specific value. Considering all of the risk factors including the risk prediction coefficient ($\overline{Ri}$), that is, the expected risk factors, the estimated weight distance (D_w) for the r-th path is expressed as follows.

$$D_{w,r} = \sum_m d_m \cdot \frac{\sum_{n=1}^{N} W_{n,m} \times O_{n,m} \times \overline{Ri_{n,m}}}{N_m},\ where\, N_m = \max(1, \sum_{n=1}^{N} O_{n,m}) \qquad (2.6)$$

Then, the optimal path decision criterion can be defined as Eq. (7) by reflecting the sensed distance D_w considering the risk prediction coefficient.

$$Optimal\, path\, II = \arg\min_{r \in R} \overline{D}_{w,r} \qquad (2.7)$$

2.2.3 Evacuation System for accident situation

The optimum route guidance system considering the disaster situation is to detect the presence of risk factors (O_n) and the risk level (W_n) through the sensor installed in the target structure and calculate the optimal distance of the escape route Search. The search procedure of the optimal path described above can be summarized as shown in **Fig 9**. In the proposed route guidance system, the route detected through this process is installed in the form of an emergency guide light which can direct the direction to the floor or the wall inside the structure, thereby inducing escape in the event of a disaster[11].

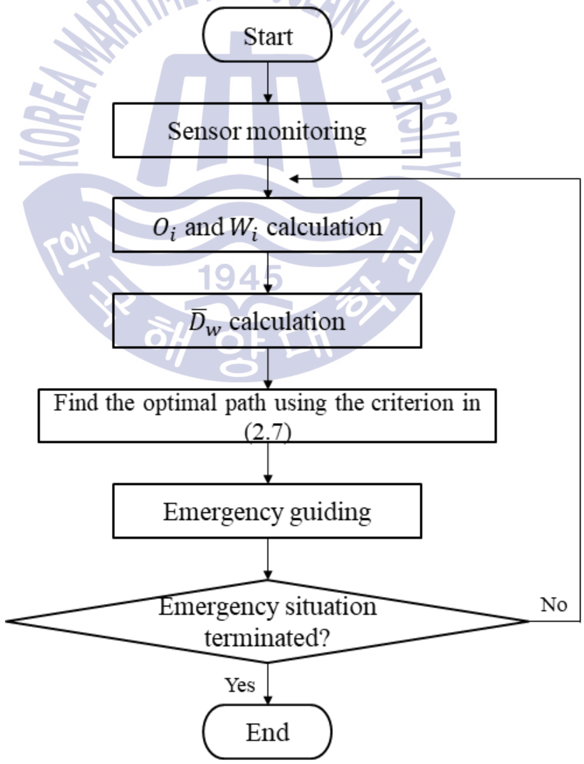
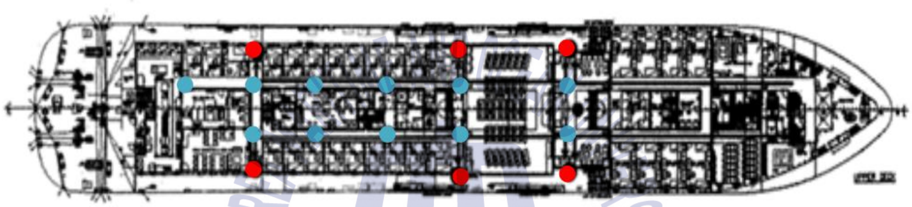


Fig 9. Optimal route discovery and leading process

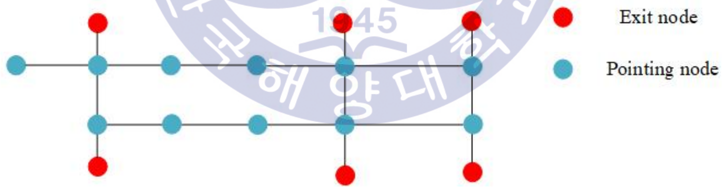
Chapter 3. Proposed Scheme

3.1 Graph Search for inside of ship

In order to apply the disaster evacuation system according to various ship structures, it is necessary to be able to identify and design the main nodes and emergency exit direction for each ship structure.



(a)



(b)

Fig 10. Graph search based on ship structure

Fig. 10 is the design drawing of the Hanbadaho, the Korea Maritime and Ocean University training ship. The exit is a red dot, and the place where a crossing point occurs is made a graph by making a main node by using a blue dot and connecting an edge between each node.

3.2 Modeling of bottleneck

In the optimal path search algorithm considering the disaster situations studied previously, the factor that is the basis of the path search is the weight distance (D_w). The optimum path is searched based on the value that minimizes the weight distance. The factors that directly affect the weight distance are the various risk factor models. It is easy to reflect factors affecting movement due to fire situation, smoke, isolation etc. based on sensor data. However, in the case of bottlenecks, depending on the population density of the surrounding nodes, the evacuation route selection may vary depending on the direction in which the people of the specific node are evacuated[12-13].

Because of this characteristic, existing route search algorithm optimizes route based on specific node without consideration of population density, so people existing in the same node are guided through the same route. If the width between the previous node and the next node can accommodate the number of people, bottleneck does not occur, which is not a problem in route selection. However, if the node accommodating rate is smaller than the population inflow rate, bottleneck occurs and it becomes difficult to apply the existing system. A value obtained by averaging the number of people who can pass through the unit time in this manner is expressed as a node capacity (Node capacity) as follows.

$$C_m = \frac{P_m}{t} \tag{3.1}$$

C_m : Node capacity

P_m : The number of people

Using the accommodating rate in the interval, it is possible to obtain the variation of the weight distance by the bottleneck by dividing the number of persons distributed in the corresponding section in each section from the section to the section accommodating rate in the next section.

$$\Delta D_m = \frac{P_m}{C_m} \cdot \ v \tag{3.2}$$

It is presupposed that the people distributed in each section when the accident occurs move according to the average moving speed toward the exit.

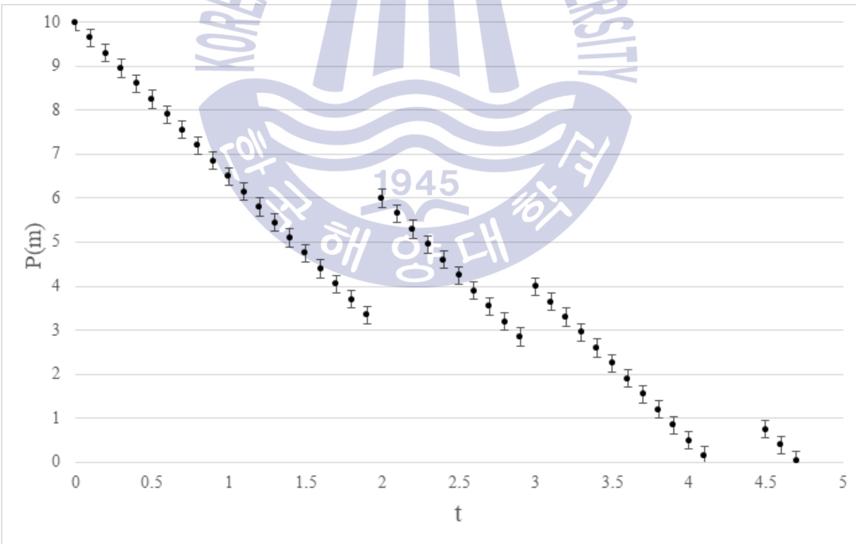


Fig 11. Simulation of bottleneck measurement

The graph shows the value of P_m for an arbitrary interval m according to the proposed method. The initial value of P_m in section m was set to 10, and the value C_m was set to $3.5\,people/s$, and the inflow rate in any three sections was repeated about 100 times reflecting the arbitrary value. As a result, it can be confirmed that bottleneck occurs when the value of P_m is larger than that of the bottleneck, and a delay of about 1.7 seconds occurs depending on the inflow rate compared to the initial number of people until the bottleneck is solved. The experimental data may be different depending on the number of human inflow sections and the inflow rate in each inflow section.

3.3 Proposed Route Optimization Algorithm

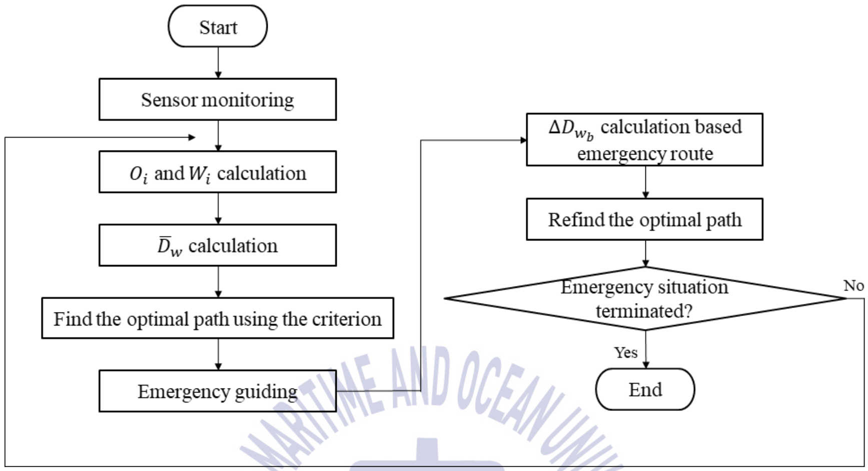


Fig 12. Proposed algorithm

The route guidance system considering the previously proposed disaster situation is the minimum cost based route search algorithm after calculating the weight distance based on the sensation factor and the risk prediction from each node to the entrance. These algorithms have the disadvantage that bottlenecks may occur on the evacuation path if there are a large number of people guided by the same path in each node, which may cause a difference in performance depending on the situation. Accordingly, each node can perform the first route search and then search and reflect the area where the bottleneck occurs again in advance to prevent or eliminate the bottleneck.

The proposed load-balanced path optimization algorithm is as follows.

(1) In the event of a disaster, the sensor node of each major node invokes the sensor data value for the risk element, calculates and stores the risk factors and the risk level.

(2) It calculates the weight distance reflecting the risk factors except the bottleneck, and performs the primary path search based on the corresponding value at each node.

(3) It checks the directionality of each node and calculates the weight distance due to the expected bottleneck when moving in that direction.

(4) The final value is derived by reflecting the weight distance value by the bottleneck and the weight distance value reflecting the existing risk factors, and the secondary path search is performed based on the value.

Chapter 4. Simulation and Analysis

4.1 Bottleneck occurrence probability

Since the existing evacuation algorithm searches for the optimal path at each node, if there are many people starting from the same node, the bottleneck may occur and the performance may be deteriorated. Therefore, we experimented how the probability of bottlenecks is decreased according to the proposed algorithm.

4.1.1 Experiment environment and result

Table 1. Parameters of simulation

Parameter	Value
Edge	$5m$
Occurrence probability of risk element	20%
Weight of risk element	10
The number of people of each nodes	0 ~ 10
Node capacity	2 ~ 4
Number of iterations per round	100
Walking speed	$0.5m/s$

As shown in **Fig 10**. (b), the main node and the trunk lines are arranged and the route optimization from each node to the exit is simulated through the graph search method of the actual ship structure. The value of each trunk line is set to 5m, and the probability of risk occurrence at each node is assumed to be 20%. Risk estimates are not reflected. Also, the number of people located in each node is arbitrary value, and an integer value between 0 and 10 is set. The node acceptance rate of each node is set to any value from 2 to 4, and after 100 simulations per round The result graph is derived based on the mean value.

Conv_1 in **Fig. 13** is the graph when only the shortest distance is searched based on the optimal path system, and Conv_2 is the graph when the existing system is applied. Prop is a bottleneck occurrence probability graph when the proposed system is applied.

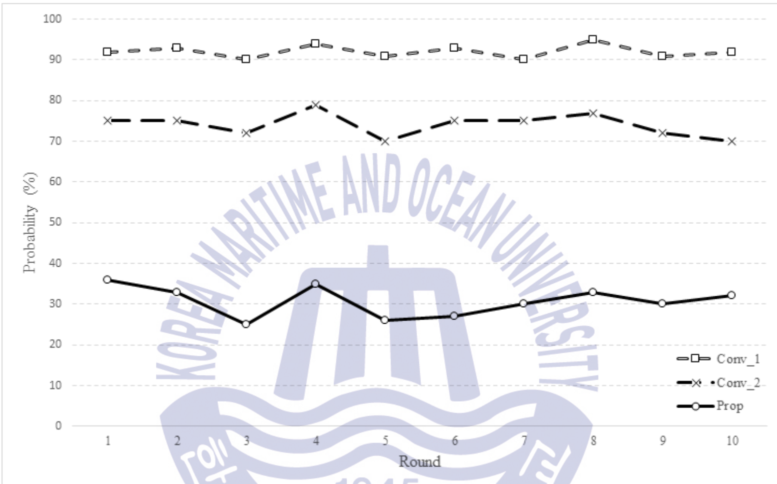


Fig 13. Bottleneck occurrence probability

It is confirmed that the probability of occurrence of bottlenecks is reduced when the proposed algorithm is applied to the existing emergency guiding algorithm. However, the cost is increased by taking the bypass route to avoid the bottleneck, which indicates an increase in the total evacuation time. However, it can be seen that the overall stability is improved by reducing the additional delay of a particular node due to the bottleneck.

4.2 Weighted distance according to proposed scheme

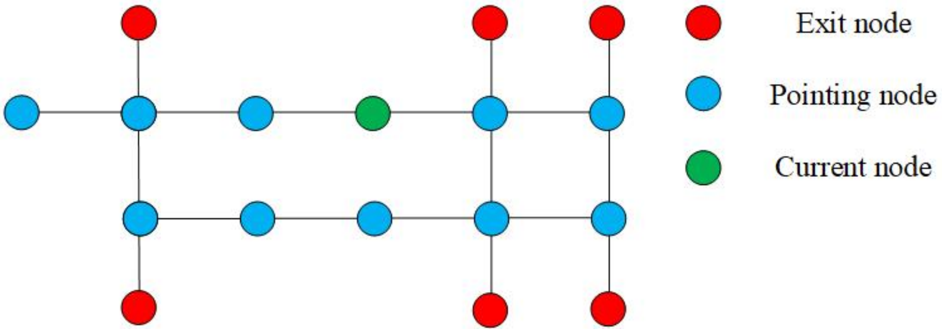


Fig 14. Virtual simulation environment

In the graph of **Fig. 14**, we have simulated the increase in the weight distance due to the bottleneck at the current node. The parameters applied to the simulation are the same as those used in the simulation.

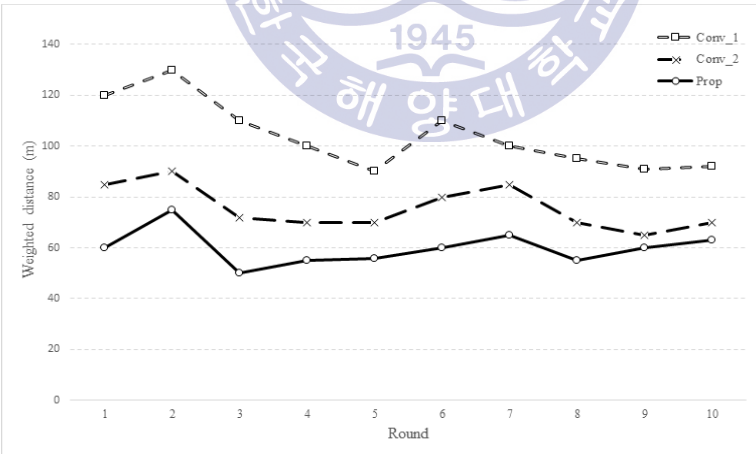


Fig 15. Weighted distance according to proposed schemes

Fig. 15 is a graph showing the value of the sensed distance according to the proposed method at the current node. In case of Conv_1, when the risk factor generation and the bottleneck occur at the same time, the value of the weight distance is accumulated and the average value is measured larger than the other graphs. In case of Conv_2, it can be confirmed that the value of the weight distance is reduced by applying the path optimization. However, in the graph of Prop, it can be seen that by applying the load-balancing to the existing route optimization algorithm, the sense distance is decreased by about 20% on the average.

4.3 Evacuation time according to proposed scheme

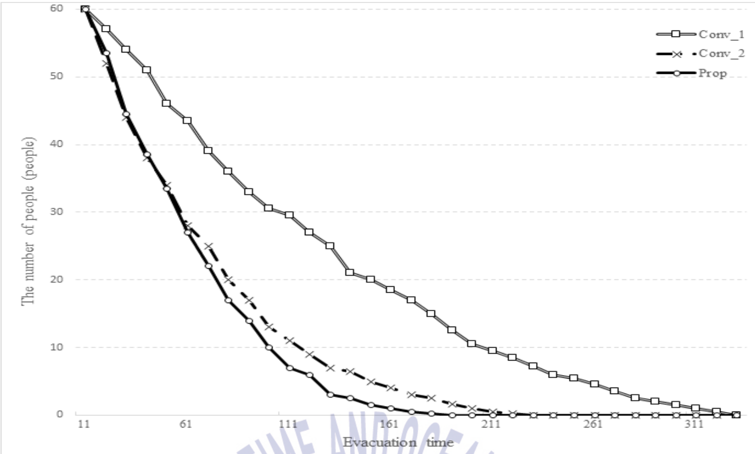


Fig 16. Evacuation time according to proposed schemes

Fig 16. measures the escape time according to the proposed method. Conv_1 represents the time taken to retrieve the route based on the shortest distance, and Conv_2 is a graph that shows the application of the proposed method when applying the existing system. It can be confirmed that the overall escape time is delayed when the route is searched based on the shortest distance. On the other hand, it can be confirmed that the performance of the proposed system is similar to that of the conventional system up to about 50 seconds after the simulation operation. However, from the time after 50 seconds, the performance of the proposed system is superior to that of the existing system. Considering that the virtual environment of the simulation reflects the one floor structure and the complexity is low, the performance of the proposed system will be further improved. In addition, through the disaster simulator such as FDS, it can be reflected as the survival rate when the escape time limit is reflected, so that a more practical environment can be considered.

Chapter 5. Conclusiond

In this paper, we analyze the bottleneck conditions, model the bottlenecks, and consider the bottleneck phenomenon in addition to the proposed system to evacuate quickly and safely in a complex ship in case of a disaster.

Existing evacuation path algorithms do not reflect bottlenecks, so evacuation guidance systems can add bottlenecks in certain situations, which can miss the golden time of evacuation and rescue. Therefore, we analyze the bottleneck conditions and apply them to the proposed algorithm and verify the performance through simulation in this paper.

Experimental results show that the proposed algorithm can reduce the bottleneck probability and solve the additional bottleneck, but it has a disadvantage that the escape time is delayed due to the increase of the travel distance due to the load distribution. However, the delay caused by the bottleneck is reduced, which can increase the stability of the overall system.

In the future, we plan to design a more reliable intelligent evacuation system by reflecting the risk factors modeled in various disaster related simulators to the proposed system.

Reference

[1] International Maritime Organization, Guidelines for evacuation analysis for new and existing passenger ship, IMO MSC/Circ.1238, 2007.

[2] K. T. Kim, S. J. Cho, and G. W. Jeon, "A study on the optimal routing planning algorithm for rescue of multiple victims in disaster area," Journal of Korea Safety Management & Science, vol. 12, no. 2, pp. 17-23, 2010 (in Korean).

[3] M. Dong, H. Li, K. Ota, "Multicloud-Based Evacuation Services for Emergency Management," IEEE Cloud Computing, vol. 1, Issue 4, pp. 50-59, Nov, 2014.

[4] Q. S. JIA, Y. GUO, "Event-based Evacuation in Outdoor Environment," Chinese Control and Decision Conference, pp. 33-38, 24th, 2012.

[5] L. Chu, "A RFID-Based Hybrid Building Fire Evacuation System on Mobile Phone," Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on, Oct, 2010.

[6] P. B. Luh, C. T. Wilkie, S.H. Chang, "Modeling and Optimization of Building Emergency Evacuation Considering Blocking Effects on Crowd Movement," IEEE Transactions on Automation Science and Engineering, Vol. 9, Issue 4, Oct. 2012.

[7] B. Kim, K. I. Hwang, "A study on prediction and improvement method of fire risk for a newly built college dormitory," Journal of the Korean Society of Marine Engineering, Vol. 40, no. 3, pp. 228-234, Mar, 2016.

[8] K. I. Hwang. "A Prediction on Indoor Contaminant Diffusion Characteristics of a Training Ship by Mechanical Ventilation System,"

Journal of the Korean Society of Marine Engineering, Vol. 35, no. 8, Nov, 2011.

[9] K. I. Hwang, D. K. Shin, Y. J. Kim, J. H. Youn, S.I. Lee, W. H. Hong, "A Case Study on the Passengers' Evacuation Times according to the Fire Smoke Density On a Ship," Journal of the Korean Society of Marine Engineering, Vol. 33, no. 2, pp. 336-343, Mar, 2009.

[10] G. Hart, P. E. Nilsson, B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybemetics SSC4, 1968.

[11] M. B. Kang, Y. I. Joo, "Intelligent evacuation systems for accidents aboard a ship," Journal of the Korean Society of Marine Engineering, vol. 40, no. 9, pp. 824-829, 2016 (in Korean).

[12] R. Kim, Y. I. Joo, "Intelligent evacuation systems considering bottleneck," Proceeding of the Korean Institute of Information and Communication Engineering Spring Conference, p. 63, 2017 (in Korean)

[13] R. Kim, Y. I. Joo, "Bottleneck modeling for disaster evacuation systems," Proceedings of the 41st Korean Society of Marine Engineering Conference, p. 192, 2017 (in Korean).