Applied Mathematical Sciences, Vol. 9, 2015, no. 31, 1531 - 1541
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2015.49681

Modification of Dijkstra's Algorithm for Safest and Shortest Path during Emergency Evacuation

Khyrina Airin Fariza Abu Samah

Faculty of Information Technology and Communication
Universiti Teknikal Malaysia Melaka (UTeM)
76100 Durian Tunggal, Melaka, Malaysia

Burairah Hussin

Faculty of Information Technology and Communication
Universiti Teknikal Malaysia Melaka (UTeM)
76100 Durian Tunggal, Melaka, Malaysia

Abd Samad Hasan Basari

Faculty of Information Technology and Communication
Universiti Teknikal Malaysia Melaka (UTeM)
76100 Durian Tunggal, Melaka, Malaysia

Abstract

Intelligent autonomous evacuation navigation (AEN) system manages to solve wayfinding problems during emergency evacuation, especially for those who are not familiar with the building. AEN system is expected to provide independent evacuation navigation through the dynamic signage, which helps occupants navigate to the safest and shortest path towards the nearest exit. It uses modified Dijkstra's algorithm in order to incorporate both aspects and considering the hazardous location. Therefore, this paper presents the methodology used for the modification of Dijkstra's algorithm to find the safest and shortest path. The first modification involved the restriction of floor plan node's direction; and the second modification was blocking the nodes that affected by fire. The example is presented to compare the result obtained using the modified algorithm. Thus, it reveals the effectiveness of the modification in determining the best path for the evacuation process.

Keywords: wayfinding, Dijkstra's algorithm, evacuation, emergency evacuation, safest and shortest, autonomous evacuation navigation

1 Introduction

Currently, people are competing to develop the modern high rise building and complex, but rarely think about the safety aspects in evacuation preparedness. Although the fire alarm system is available, but it only focuses on the detection of any abnormalities, e.g. fire incident. Once the fire indicator detected smoke and heat, the alarm triggered continuously and the occupants follow the evacuation procedure. They know the urgency to evacuate, but do not know the safest path to get out of the building especially for unfamiliar occupants. Most of them used the familiar path [1],[2]; but what will happen if they were getting into traps at the spreading hazardous location? The same condition happened to familiar occupants, although they know the safest path to go.

Wayfinding during emergency evacuation is the critical problem, especially for those who are not familiar with the building environment. How those intelligent modern building system help unfamiliar occupants navigate to the safest nearest exit? A few research study on the usage of an agent [1],[2], simulation [3],[4],[5] and robot [6] to solve the problem, but in a real emergency situation, different human behavior contributes to the failure [6],[7],[8],[9].

Therefore, we propose the autonomous evacuation navigation (AEN) system, which expected to provide independent evacuation navigation through the dynamic signage and enables to navigate occupants towards the safest and shortest path. In order to incorporate both aspect; safest and shortest, the AEN system used to modify the Dijkstra's algorithm to find the safest and shortest path from one node to another node. Hence, this paper discussed details on the modification and how it works to find the best path for the evacuation process.

This paper organized as follows: section 2 explained the design of the AEN system; section 3 reviewed Dijkstra's algorithm principle and description; section 4 discussed the methodology on modification of Dijkstra's algorithm for safest and shortest path; section 5 discussed the analysis of algorithms applied; and section 6 provides a brief conclusion of our work.

2 Design of Autonomous Evacuation Navigation System

2.1 Conceptual Framework of AEN and Methodology

The AEN system used the existing system available, such as fire alarm control panel, fire indicator and the original building floor plan.

Figure 1 shows the conceptual framework for AEN which consists of two layers; first layer is the information retrieval and the second layer is the decision making [10]. The standard model developed has the specific input, process and the output.

The first layer is divided into two stages, input and process. The input consists of two components, and process has four components. We have to complete all the components in the process stage using the input from the floor plan layout. Once the first layer completed, then we move to the second layer which has one component as output of the model.

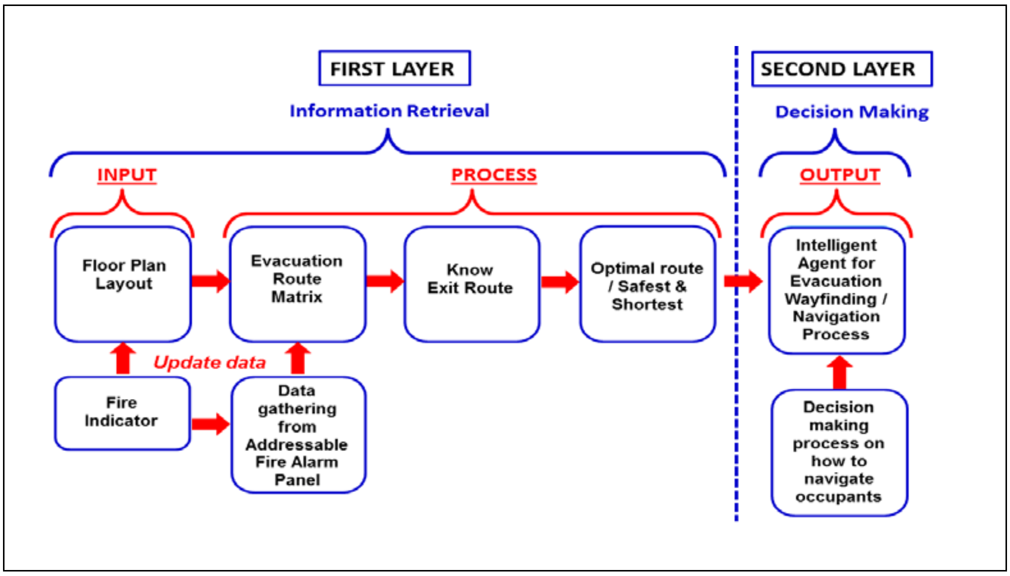

Figure 1. Conceptual Framework For Autonomous Evacuation Navigation

2.2 Methodology of AEN

A flowchart as in Figure 2 shows the implemented methodology for the first layer, where the modification involved once completed the evacuation matrix. Each of the steps plays an important role to ensure the success of the system.

The methodology begins with the selection of actual floor government office in 14th floor, due to the complex structures enclosed space, multilevel and visited by people from all ages. It is suitable to further analyze the familiar and unfamiliar occupant behavior in wayfinding selection and measure the effectiveness of evacuation preparedness.

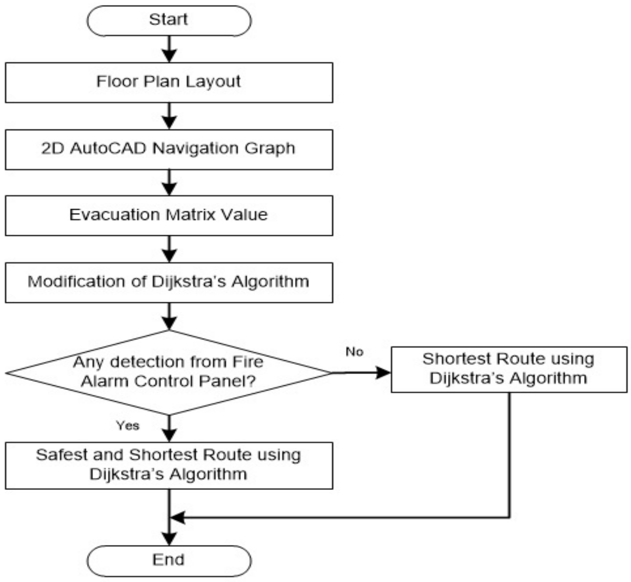

Figure 2. Flowchart for AEN First Layer Methodology

For the second flow, the navigation graph of floor plan is modelled layout using 2D AutoCAD with the weighted value from node to node, which is measured in meter (m) as in Figure 3. Active node is the fire indicator (green color) and the inactive node is the door (pink color).

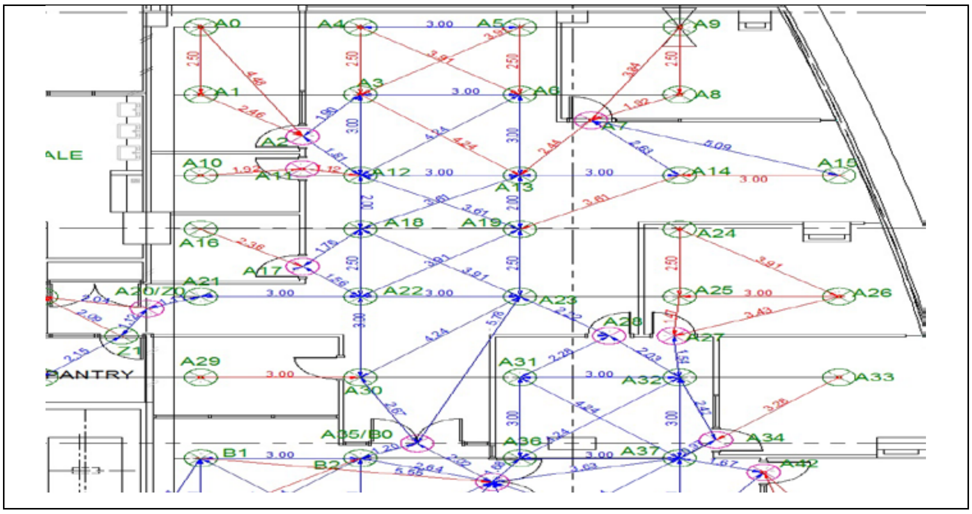

Figure 3. 2D Layout of Navigation Graph - 14th Floor Government Office

For this study, we take into consideration of the original floor plan obstacle such as wall, which is cannot be passing through. The doors react as nodes because it is the vital ways for the occupants choose during the evacuation to the safe place. In addition, since the case study building is on the 14th floor, window cannot be a node and do not play any important role. Therefore, we calculate the additional length towards the active node through the inactive node. As the study focused on evacuation planning for evacuation, we set the staircase as the endpoint and the last egress path to evacuate.

In the third flow, all the nodes and weighted values are stored using Microsoft Excel table as an evacuation path matrix and save it into the 'text' file format. This file will be used in the C++ programming later. The last step is the modification of Dijkstra's Algorithm for the implementation which will be discussed further in Section 4.

3 Dijkstra's Algorithm

There are many types of shortest path algorithms available, but for this study, we look forward also the safest path to evacuate. Dijkstra's algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs. According to [11], [12], [13], since 1959 Dijkstra's algorithm has been recognized as the best algorithm and used as method to find the shortest path. The shortest path is that of finding a path of minimum weight connecting two specified vertices, or between all pairs of vertices, or from the specified vertex to all others. In addition, the solution time required is very less comparing to dynamic programming or linear programming.

The problem of finding the shortest path from a specified vertex source point '*s*' to another vertex point '*t*', can be stated as follows:

1) A weighted connected simple graph with all weights positive, $G = \langle V, W \rangle$, all nodes in it stored in set V (v_0 , v_1, $v_2 \ldots\ldots v_n \in V$) and the weights of each neighboring nodes are stored in set W (W_0 , W_1, $W_2 \ldots\ldots W_n \in W$) and all points starting from the source point of the shortest path are stored in set S. When initializing, only the specific zone value are stored in $V - S$. $D(j)$ represents the distance between the source point 's' and the point 'e_j'. All points in the weighted graph represented by the adjacency matrix A. $A(i,j)$ means the distance between point 'v_i' and 'v_j' in matrix A and $A(i,j) = \infty$ when there is not direct path or not an edge in G.

2) Initialization, $S = \{s\}$, $V - S = \{v_0, v_1, v_2 \ldots\ldots v_n \in V\}$. Select the points '$v_j$' and make $D(j) = \min \{D(i) \mid v_j \in (V - S)\}$, $S = S \cup v_j$;

3) Change the source point '*s*' to point 'v_j'$\in (V - S)$ and $D(j) = D(i) + A(i,j)$, if $D(j) > D(i) + A(i,j)$.

4) Repeat steps (2) and step (3) until obtaining the shortest path from the source point '*s*' to the rest nodes.

4 Methodology on The Modification of Dijkstra's Algorithm for Autonomous Evacuation Navigation

The modification is considered the safest path, then followed by the shortest path. Through it, we fulfil the research gaps towards navigating wayfinding occupants towards the safest path as priority, in which it also covered guidance both occupants, familiar and unfamiliar. As in Figure 1, the data gathered from the fire alarm control panel was the fire indicator's location, which is the same node number on the floor plan layout. Therefore, the algorithm is divided into two types of emergency evacuation. If the evacuations are not involved any abnormalities of the fire indicators such as bomb threats or spills of toxic gases, the algorithm is focused on shortest path only. But, if it involves the abnormality detection of the fire indicator such as smoke, heat or flames, the algorithm will consider the safest path and then followed by the shortest path calculation. The two modifications involved are the node's direction and the Dijkstra's algorithm.

4.1 Modification of the Node's Direction

When the emergency alarm triggers, security is the most important issue need to be focused. At this stage, occupants must evacuate as soon as possible due the hazards such as fire are easily spread inside the building. Therefore, we labeled the restriction of the node direction; one way or two-ways (as shown in previous Figure 3). For further explanation, a simple example used as illustrated in Figure 4. The restriction used blue arrow for two ways direction and red arrow for one way direction only. Since the floor was divided into subsection, we assume the occupants cannot turn back to the corner nodes or any enclosed area with no way out. In addition, the calculation time can be reduced without considering the weighted value.

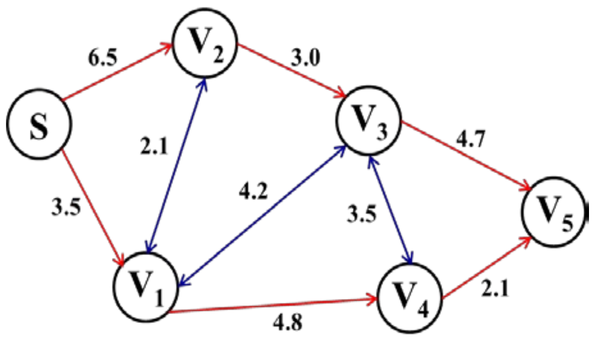

Figure 4. Graph with Nodes Diagram

The graph starts with '*s*' point to any '*v*' point in the weighted value and transform it into the matrix. The adjacency matrix value is different compared to normal matrix.

$$
A = \begin{array}{c} \\ \end{array}
\begin{array}{cccccc} \boldsymbol{S} & \boldsymbol{V_1} & \boldsymbol{V_2} & \boldsymbol{V_3} & \boldsymbol{V_4} & \boldsymbol{V_5} \end{array}
$$

$$
A = \left\{ \begin{array}{cccccc}
0 & 3.5 & 6.5 & \infty & \infty & \infty \\
\infty & 0 & 2.1 & 4.2 & 4.8 & \infty \\
\infty & 2.1 & 0 & 3.0 & \infty & \infty \\
\infty & 4.2 & \infty & 0 & 3.5 & 4.7 \\
\infty & \infty & \infty & 3.5 & 0 & 2.1 \\
\infty & \infty & \infty & \infty & \infty & 0
\end{array} \right\}
\begin{array}{c} \boldsymbol{S} \\ \boldsymbol{V_1} \\ \boldsymbol{V_2} \\ \boldsymbol{V_3} \\ \boldsymbol{V_4} \\ \boldsymbol{V_5} \end{array}
$$

4.2 Modification the Dijkstra's Algorithm

The Dijkstra's algorithm modification involved two methods; update the matrix file and modify the pseudo code for the algorithm.

i.Update the Matrix File

The related nodes (fire indicator's location), A_{ij} will be blocked and no one can use the path, although it might be the shortest path towards to the exit. The node A_{ij} value will be updated as ∞ or 999 in the temporary matrix file.

ii.Modify Pseudo Code for the Dijsktra's Algorithm

For this study, we use the C++ program to calculate the shortest path. Therefore, we have updated the algorithm pseudocode as in Figure 5 by adding new functionality; to block certain nodes and provide another path.

```
Program BlockActiveNode:

Input: The zone, the blockage quantity    //data from FACPS
Process: Updating the matrix value    //stored value = 999
Output: Graph G (V, S) - the safest and shortest route        //the route
Procedure:

Begin
Initialize qtyblock to zero;
Initialize zone to zero;
Initialize zone matrix file as ofstream file;

Input the involve zone; //fire indicator zone
Search for zone file;
Open the zone file;    //open the zone matrix file //read the total matrix no
Input the blockage quantity; //qtyblock

FOR i=0 to total matrix no
        FOR j=0, j to total matrix no
                FOR n=0, n to qtyblock
                        IF (involve zone equal to zone(i,j)
                THEN set zone (i, j) to 999;    // value to 999
                        ELSE do nothing;
          ENDIF;
                Update the zone matrix file

Close the zone matrix file
Open the Program CalculateDistance;
Open the Program OutputPrinted;
Output: Graph G (V,S)         //print the Graph
```

Figure 5. Pseudo Code for Node Blockage

5 Analysis of the Modification

5.1 Modification on the Nodes Direction

Using the same graph illustrated, the node directions for each of the graphs were identified. We analyzed the steps involved by labelling the nodes and the weighted in order to find the shortest path as in Table 1.

As a result, the shortest path from S to V_5 was 10.4 (as bold in (P)) and the nodes involved were $S \rightarrow V_1 \rightarrow V_4 \rightarrow V_5$.

Table 1. Finding the Shortest Path Label from S to V_5 (original algorithm)

S	V_1	V_2	V_3	V_4	V_5	**Description**
0(P)	∞	∞	∞	∞	∞	Starting Node S is labeled 0
0(p)	3.5	6.5	∞	∞	∞	All successors of S get labeled
0(p)	**3.5(P)**	6.5	∞	∞	∞	Smallest label becomes permanent
0(p)	3.5(p)	5.6	7.7	8.3	∞	Successors of V_1 get labeled
0(p)	3.5(p)	**5.6(P)**	7.7	8.3	∞	Smallest label becomes permanent
0(p)	3.5(p)	5.6(p)	8.6	8.3	∞	Successors of V_2 get labeled
0(p)	3.5(p)	5.6(p)	8.6	**8.3(P)**	∞	Smallest label becomes permanent
0(p)	3.5(p)	5.6(p)	11.8	8.3(p)	10.4	Successors of V_4 get labeled
0(p)	3.5(p)	5.6(p)	12.8	8.3(p)	**10.4(P)**	Smallest label becomes permanent

The execution time taken was captured and been compared the result using the Wilcoxon Rank Sum test (Mann-Whitney U test) with continuity correction as shown in Table 2. For this study, we have only two nominal variables, which is ‘no restriction’ and ‘restriction’, independent data and categorized under non-parametric [14],[15]. Non parametric test are simple, easy to understand, and not involved complicated sampling theory. The null and alternative hypotheses for the test are; H_0: No restriction time taken resulted same time taken with the restriction, and H_1: No restriction time taken resulted different time taken with the restriction. The *p-value* shows 0.00195; which *p-value* < 0.05. Therefore, we concluded that the averages of two groups are significantly different for unfamiliar occupants, and accepted the H_1: No restriction time taken resulted different time taken with the restriction.

Table 2. Wilcoxon Signed Rank Test of Two Samples

Wilcoxon signed rank test

Execution time, *s*	**No Restriction**	**Restriction**
Min	1.207	0.532
Max	1.420	0.760
Median	1.328	0.623
Mean	1.325	0.637
Var	0.004	0.057
p-value 0.00195		
*$p<0.05$		

5.2 Modification the Dijkstra's Algorithm

Once the algorithm had been modified by adding the new blockage function, we assumed node V_1 is the affected node and we have to block the node. Table 3 shows the shortest path result from S to V_5 changed to 14.2 and the nodes involved were $S \rightarrow V_2 \rightarrow V_3 \rightarrow V_5$.

Table 3. Finding Shortest Path Label from S to V_5 (modified algorithm)

S	V_2	V_3	V_4	V_5	**Description**
0(P)	∞	∞	∞	∞	Starting Node S is labeled 0
0(p)	6.5	∞	∞	∞	All successors of S get labeled
0(p)	**6.5(P)**	∞	∞	∞	Smallest label becomes permanent
0(p)	6.5(p)	9.5	∞	∞	Successors of V_2 get labeled
0(p)	6.5(p)	**9.5(P)**	∞	∞	Smallest label becomes permanent
0(p)	6.5(p)	9.5(p)	13.0	14.2	Successors of V_3 get labeled
0(p)	6.5(p)	9.5(p)	**13.0(P)**	16.3	Smallest label becomes permanent
0(p)	6.5(p)	9.5(p)	13.0(p)	16.3	Successors of V_4 get labeled
0(p)	6.5(p)	9.5(p)	13.0	**14.2(P)**	Smallest label becomes permanent

6 Conclusion

Dijkstra's algorithm has proven its advantages to solve the wayfinding problem towards the safest and shortest path in emergency evacuation. Through this study, we able to avoid the hazardous location through blocking the affected area, calculate, and provides the updated path. It contributes the significance to the AEN system in terms of wayfinding navigation, and makes the system more intelligent and automated. For future study, it is recommended to have a low computational cost using the existing sources and may be scalable to large evacuation scenarios; due to the various types of evacuation [16].

Acknowledgements. This research project is conducted by Universiti Teknikal Malaysia Melaka, Malaysia (UTeM) which funded from Malaysian government under the Fundamental Research Grant Scheme (FRGS/1/2013/ICT07/02/2/F00162).

References

[1] C. Hölscher, T. Tenbrink, and J. M. Wiener, "Would You Follow Your Own Route Description? Cognitive Strategies in Urban Route Planning," *Cognition*, vol. 121, no. 2, pp. 228–47, Nov. 2011. http://dx.doi.org/10.1016/j.cognition.2011.06.005

[2] T. Meilinger, J. Frankenstein, and H. H. Bülthoff, "Learning to Navigate: Experience Versus Maps," *Cognition*, vol. 129, no. 1, pp. 24–30, Oct. 2013. http://dx.doi.org/10.1016/j.cognition.2013.05.013

[3] Y. Liu, D. Shen, N. Okada, and S. Li, "Vitae System Based Agent Modeling and Simulation of Survivability-Critical Evacuation in Underground Flooding," in *IEEE International Conference on Systems Man and Cybernetics (SMC)*, 2010, pp. 3133–3138. http://dx.doi.org/10.1109/icsmc.2010.5641730

[4] M. D'Orazio, L. Spalazzi, E. Quagliarini, and G. Bernardini, "Agent-Based Model for Earthquake Pedestrians' Evacuation in Urban Outdoor Scenarios: Behavioural Patterns Definition and Evacuation Paths Choice," *Saf. Sci.*, vol. 62, pp. 450–465, Feb. 2014. http://dx.doi.org/10.1016/j.ssci.2013.09.014

[5] Y. Sato and Y. Osana, "Office Layout Plan Evaluation System Using Evacuation Simulation Considering Other Agents' Action," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2012, vol. 2, pp. 1911–1916. http://dx.doi.org/10.1109/icsmc.2012.6378017

[6] Y. Kim, Y. Kim, S. Lee, J. Kang, and J. An, "Portable Fire Evacuation Guide Robot System," pp. 2789–2794, 2009. http://dx.doi.org/10.1109/iros.2009.5353970

[7] N. Kim, J. Joo, L. Rothrock, R. Wysk, and Y.-J. Son, "Human Behavioral Simulation Using Affordance-Based Agent Model," in *Human Computer Interaction. Design and Development Approaches*, 2011, pp. 368–377. http://dx.doi.org/10.1007/978-3-642-21602-2_40

[8] H. Keshavarz, "Human Information Behaviour and Design, Development and Evaluation of Information Retrieval Systems," *Progr. Electron. Libr. Inf. Syst.*, vol. 42, no. 4, pp. 391–401, 2008. http://dx.doi.org/10.1108/00330330810912070

[9] M. Kobes, I. Helsloot, B. de Vries, and J. G. Post, "Building Safety and Human Behaviour In Fire: A Literature Review," *Fire Saf. J.*, vol. 45, no. 1, pp. 1–11, Jan. 2010. http://dx.doi.org/10.1016/j.firesaf.2009.08.005

[10] K. A. F. A. Samah, B. Hussin, A. S. H. Basari, and N. A. M. Sabri, "Modelling Conceptual Framework for Autonomous Intelligent Evacuation Wayfinding in Unfamiliar Building: Replacing Human with Intelligent Agent," in *International Conference on Applied and Theoretical Information Systems Research*, 2014.

[11] Y. Xu, Z. Wang, Q. Zheng, and Z. Han, "The Application of Dijkstra's Algorithm in The Intelligent Fire Evacuation System," in *4th International Conference on Intelligent Human-Machine Systems and Cybernetics The*, 2012, vol. 1, pp. 3–6. http://dx.doi.org/10.1109/ihmsc.2012.7

[12] N. R. Shankar and V. Sireesha, "Using Modified Dijkstra' s Algorithm for Critical Path Method in A Project Network," *Int. J. Comput. Appl. Math.*, vol. 5, no. 2, pp. 217–225, 2010.

[13] Y. Sharma, S. C. Saini, and M. Bhandhari, "Comparison of Dijkstra' s Shortest Path Algorithm with Genetic Algorithm for Static and Dynamic Routing Network," *Int. J. Electron. Comput. Sci. Eng.*, vol. 1, no. 2, pp. 416–425, 2012.

[14] S. Heliövaara, J.-M. Kuusinen, T. Rinne, T. Korhonen, and H. Ehtamo, "Pedestrian Behavior and Exit Selection in Evacuation of A Corridor: An Experimental Study," *Saf. Sci.*, vol. 50, no. 2, pp. 221–227, Feb. 2012. http://dx.doi.org/10.1016/j.ssci.2011.08.020

[15] S. P. Washington, M. G. Karlaftis, and F. L. Mannering, *Statistical and Econometric Methods for Transportation Data Analysis*. Chapman & Hall/CRC No, 2003.

[16] M. Yusoff, J. Ariffin, and A. Mohamed, "Optimization Approaches for Macroscopic Emergency Evacuation Planning : A Survey," in *International Symposium on Information Technology*, 2008, vol. 3, pp. 1–7. http://dx.doi.org/10.1109/itsim.2008.4631982

Received: September 15, 2014; Published: February 25, 2015