Distributed Decision Making for Real-time in-Building Evacuation Guidance

Albert Y. Chen[1], and Chuan-Heng, Lin[2,*]

[1]Assistant Professor, Department of Civil Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, 10617, Taiwan; PH +886-2-33664255; FAX +886-2-23639990; email: albertchen@ntu.edu.tw
[2]Graduate Research Assistant, Department of Civil Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, 10617, Taiwan; PH +886-2-33664255; FAX +886-2-23639990; email: r02521514@ntu.edu.tw (*Corresponding Author)

ABSTRACT

The guidance in building evacuation during and after a disaster is an important and challenging problem, as the evacuees usually do not know the best route for evacuation and the static exit signals do not necessarily guide the evacuees to the right direction in dynamic situations. To find a best path in real-time within buildings is nontrivial. This paper proposes a distributed decision support system that efficiently computes the routes in real-time for building evacuation scenarios. Dijkstra's algorithm and Dynamic Programming are used for the route analysis. In the proposed approach, the physical system is composed of sensor nodes and decision nodes. The decision nodes, which are located at suitable locations for dynamic guidance, are equipped with information displays. The approach is evaluated in digital models and the authors built a 3D model with wall properties to be simulated with the Fire Dynamic Simulator (FDS). Fire spread and temperature distribution in the building under a mesh model are used as sensor input to test the proposed algorithm.

INTRODUCTION

Decision making during an emergency response must be timely in order to have an efficiently operation. In the case of evacuation, the main objective is to direct evacuees away from the origin of harm during the ongoing hazard. However, the evacuees do not always make best decisions during an evacuation without knowledge of the situation. In the case of in-building evacuation, civilians do not necessarily know the best path they should follow in order to reach an exit, as they may not be familiar enough with the complex in-door design of the building. In addition, the conditions inside the building may change rapidly due to the presence of hazard such as spread out of fire. This results to challenges of safe evacuation.

Most research take the centralized decision making approach to construct a centralized decision support system. A centralized approach is not fault tolerant, as it relies on a single center of data collection, processing and communication. When there is a failure in the processing center, the entire system is paralyzed. Furthermore, there may also be failures in the communication system during an emergency.

The objective of this paper is to design and implement a dynamic decision support system with distributed decision making. The system is to function in real-time, adapt to the changes of the environment, and provide reliable decisions to the evacuees regarding the direction of the best exiting path. The system use local communications for its self-operating local decision devices to inform evacuees, in the case when the communication infrastructure and the central processing center is lost. In a distributed approach, by using local information for decision making is more robust compared to a centralized solution.

The rest of the paper is organized as follows: Literature Review gives details of the current literature for the importance of evacuation guidance and distributed decision making. The data structure and computing logic are presented in the Methodology section. In the Validation section, evaluation of the proposed work along with some discussion are given. At the end is the Conclusions and Future Work section.

LITERATURE REVIEW

The static system has its limitations due to the dynamic situation in emergencies. For example, there are cases when some exits are blocked and the static signs cannot reflect this critical information. Tang et al. (2009) studied the affect of evacuation guidance on route selection for pedestrians using virtual reality. They found that people tend to move toward doors regardless of the doors' purpose and condition. As a result, evacuation guidance that provides clear and correct information is very important. Based on the best knowledge of the authors, dynamic evacuation guidance is research topic still under development.

There are many approaches to derive shortest paths, which in general could be categorized into two main categories - centralized and distributed. Leiserson et al. (2001) have a detailed explanation about centralized algorithms and their applications. However, these approaches have some drawbacks that negatively impact the system's functionality. For example, as the size of the building and the number of floors increases, the required number of decision nodes increases accordingly. The computing time of finding the best path towards the exit may be long and it cannot provide an optimal decision in the limited time frame.

Due to the fact that centralized approaches might not be robust enough during emergencies, shortest path finding problems using distributed algorithms are proposed. Humblet (1988) proposed Distributed algorithm that compute shortest paths by using Dijkstra's algorithm in a network with changing topology.

Most of the distributed algorithms are established for the network routing problem. Li et al. (2003) have discussed self-organizing sensor networks that can react to their environment and adapt to changes. They use a sensor network to calculate a path that leads to one exit and does not go through the hazardous area. The authors show the power of the algorithm to find the safest paths, but do not consider a spreading hazard. Dimakis et al. (2010) developed a simulation tool that addresses the needs for the simulation of emergency response conditions. The simulator operates in a distributed way to reduce the simulation time and dealt with the routing communication. Its implementation contains both decision nodes and sensor nodes. Nevertheless, it is a simulator in a digital world and the evacuation guidance might be more complicated when it implements in real-world case. The algorithm they proposed did not consider the paradox that the shortest path it may end up with a loop or a much longer path.

METODOLOGY

The authors proposed a distributed decision support system for emergency scenarios, and the system is expected to reduce evacuation time for individual civilians. The system is divided in three tiers, which are the modeling database, decision making model, and user interfaces. Below is the system framework (Figure 1). The detail for the framework of the decision support system is given as follows.

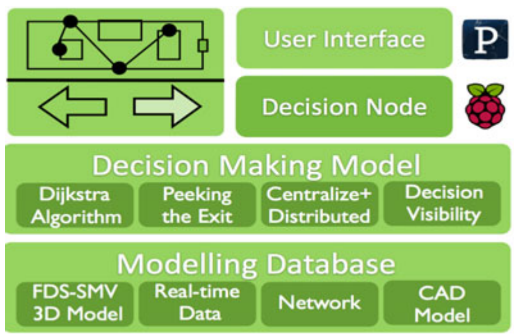

Figure 1. Decision making model.

Decision Making

As shown in Figure 2, each Decision node receives data from sensors and it shows information to both rescuers and evacuees. Under emergencies, evacuees need a quick and clear direction to avoid the hazard area and to exit. At a Decision node the authors proposed a distributed decision support system that maintains its functionality even if part of the system is nonfunctional. The proposed algorithm is based on a distributed Dijkstra's algorithm, and the efficiency of the algorithm is attempted to be enhanced by using dynamic programming and adding an ability called peeking. The definition of peeking is that the Decision node could peek for information on the nodes directly connected to the Decision node itself.

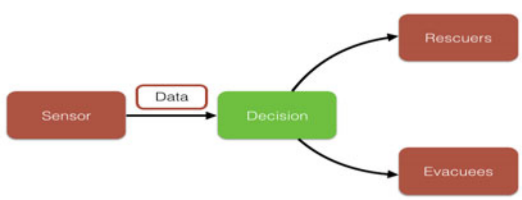

Figure 2. Workflow of decision support system.

The proposed algorithm is composed of two parts: the Update function and Calculate function. The Update function contains the work to established data structure for the decision node and the update distance table and real-time data from sensors, which only can be seen by specific neighbors. The Calculate function calculates the shortest path by the distributed Dijkstra's algorithm for evacuees and gives the direction to the next decision node towards to exit. However, the calculated path may not be shortest in dynamic situations. To solve this, the peeking ability provides an alternative opportunity as it may find the better route for evacuees with the additional information gathered from the connected nodes. More detail descriptions for the functions are given in the following.

Data Structure

In the data structure used, the Decision node contains four components as shown in Figure 3: Description, Real-Time Data, Distance Table, and Shortest Paths Table.

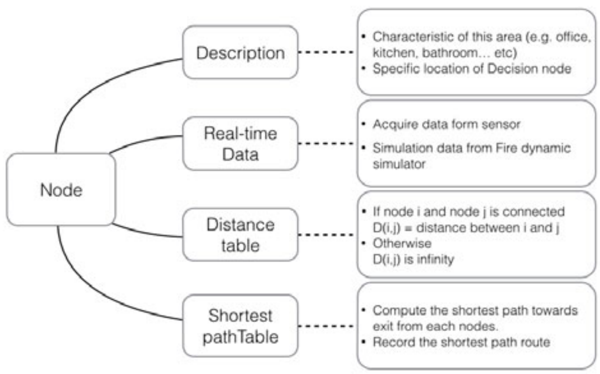

Figure 3. Data structure of decision node.

Real-time Data. As shown in Figure 3, data from sensors are received, and information for the emergency situation is updated. A binary decision is made based on the data. If it is dangerous on a network link, the distance is set to infinity. Otherwise it maintains the same distance after updated.

Description. The resulting decisions are shown to evacuees or rescuers. Included in the Description is the specific information of decision nodes. For instance, name, coordinate and which area it covers. This information can give additional guidance for both the rescuer and evacuees.

Distance Table. Distance table contains the local distance information for each node to other nodes. For example $D(I\ ,J)$ is the distance between node I and node J. The distance between two connected nodes would be the distance based on their location. In addition, updates to the distance table are carried out for different visibility level.

The visibility level is defined as one to represent in normal situation the source node can only read distance table from directly connected nodes. When visibility level is set to two, the source node will receive information from nodes within two links away from the source node. Figure 4 provides a simple network to show the visibility and its corresponding visibility table. In the table, rows represent the visibility level and each cell is the nodes' number.

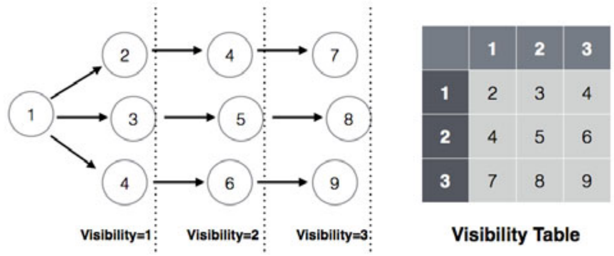

Figure 4. Simple network and Matric table to illustrate visibility.

Shortest path Table. The table contains the shortest path the distance from source node towards the exit and the path of each node calculated by the Distance Table. The information is saved to a table for reuse in the calculate function the shortest path will be updated by links in visibility range.

Computing Logic

Shown in Figure 5 is the computing logic for the proposed work. The update function transforms real-time data to binary decisions and update the distance table according to the visibility level. The calculate function computes the shortest path through Dijkstra's algorithm by updated table and give the direction to next decision node. In addition, dynamic programming is used to avoid repeat computing problem on shortest paths avoid duplicated calculations.

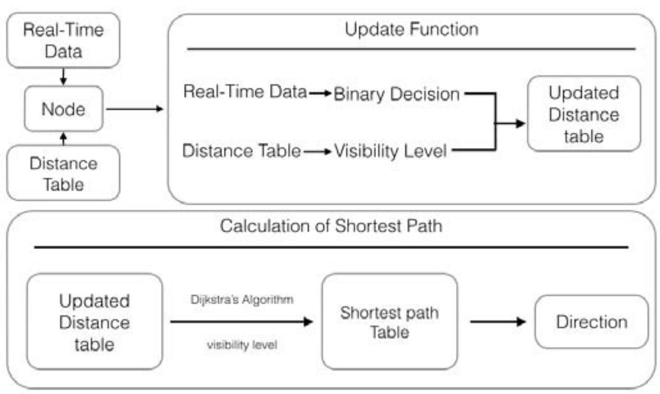

Figure 5. Computing Logic of Update function, and calculation of shortest path

In Figure 6, a simple logic is shown in the decision making after peeking. The direction from the connected node and the second node on the shortest path's direction is compared to see if they agree or not. If the direction were not the same, the second link on the path would be considered as a dangerous area and the distance is set to infinity. This way the possibility of sending people to dangerous areas is reduced and a shorter way in dynamic situation is derived.

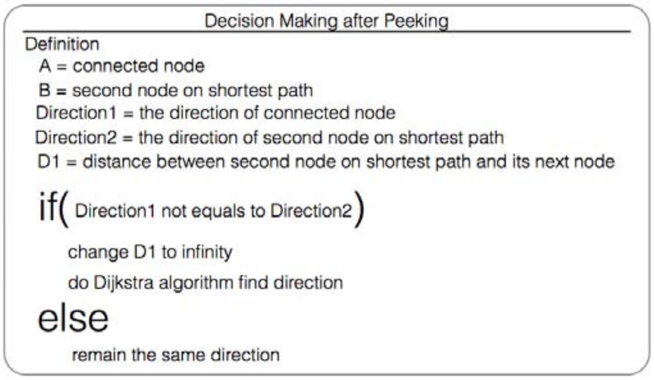

Figure 6. Decision making after peeking

VALIDATION

Matlab is used to simulate a small area to implement the proposed algorithm as shown in Figure 7. The network was constructed by the visibility graph. The algorithm correctly changes its path when there is hazard on a link. Figure 7(a) shows the shortest path when there is no fire the visibility level of Figure 7(b) is one and in Figure 7(c) two. Figure 7(b) and Figure 7(c) shows the different levels of visibility when there is a fire happened on the circled area. Compared to the distance in Figure 7(b) and Figure 7(c), we can find out that a shorter path is concluded when more information form other nodes is considered.

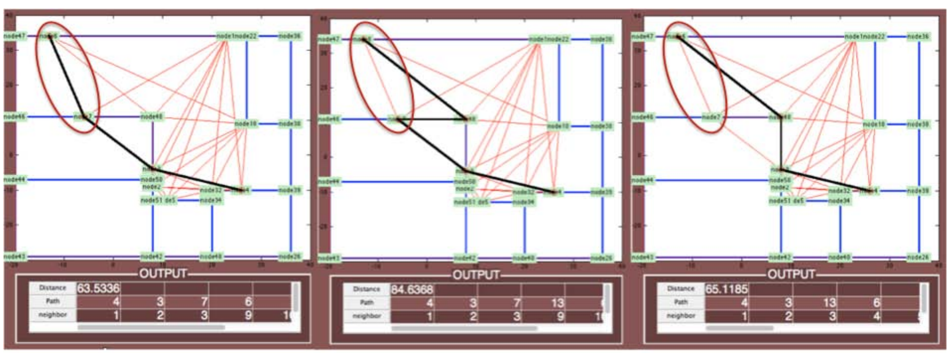

Figure 7. Visualization of shortest path (a), (b) and (c).

A random dense network for further testing shown in Figure 8(a) is conducted. The random network can test the peeking ability. A 3D model was simulated by Fire Dynamic simulator (FDS), and the information of temperature, smoke density, and distance implemented on each vertex are send decision nodes in the proposed approach. FDS is a computational fluid dynamics (CFD) model of fire-driven fluid flow and Smokeview visualizes the FDS model in a 3D environment as shown in Figure 8(b). Baum et al. (2000) showed that the numerical test for CFD model in FDS is similar to real-world cases and there are research efforts such as Floyd et al. (2005), Shi et al. (2008), and Chi et al. (2011) that utilized FDS to generate series of data as input to the proposed system for simulation.

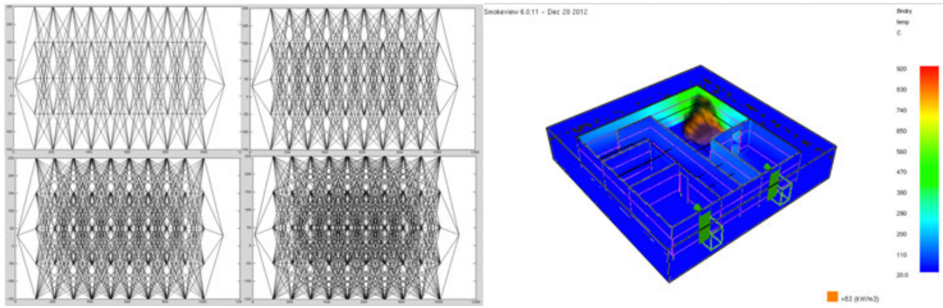

Figure 8. (a) Random dense networks and (b) Temperature in Smokeview.

In addition, comparison of the proposed approach to the original Dijkstra's algorithm under a 10 layer random network is also conducted. Figure 9(a) shows that the execution time for the Dijkstra's algorithm would spend more than the proposed algorithm. In Figure 9(b), it is shown that when visibility level is set to three, the performance is as good as a centralized Dijkstra's algorithm. The distributed algorithm can have the same performance with a gain in execution efficiency.

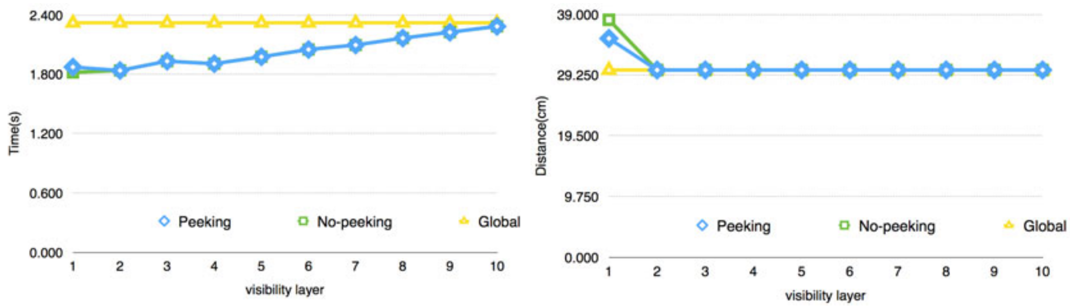

Figure 9. (a)Execution time and (b) Distance from source node to exit.

CONCLUSION & FUTURE WORK

The authors have presented a decision support system that give directions to evacuees during an emergency situation. Each decision node computes the best direction towards the exit in a distributed manner, by using only local information. The authors compared different visibility levels for the study of the efficiency and performance of the proposed approach. The peeking ability is included to the Decision nodes that can reduce the chance to have a looping cycle or go through a longer path in distributed decision making. Future work is to include risk assessment parameters such as temperature, smoke density and congestion to seek for an efficient path. In addition, implementation of the approach into a physical system is to be carried out.

ACKNOWLEDGEMENTS

The authors would like to acknowledge support by the National Science Council of Taiwan under grants NSC 102-2627-M-002-016.

REFERENCES

Baum, H. R., Rehm, R. G., Hamins, A., Forney, G. P., Floyd, J. E., & Hostikka, S. (2000). “Fire dynamics simulator: technical reference guide.” US Department of Commerce, Technology Administration, National Institute of Standards and Technology.

Chi, J. H., Wu, S. H., Shu, C. M.(2011). “Using Fire Dynamics Simulator (FDS) to Reconstruct a Hydroelectric Power Plant Fire Accident “, *Journal of Forensic Sciences*, Vol. 56, No. 6, 1639-1644.

Dimakis, N., Filippoupolitis, A., & Gelenbe, E. (2010). “Distributed building evacuation simulator for smart emergency management.” *The Computer Journal*, 53(9), 1384-1400 Floyd, J. E., Hunt, S. P., Williams, F. W., & Tatem, P. A. (2005). “A network fire model for the simulation of fire growth and smoke spread in multiple compartments with complex ventilation.” *Journal of Fire Protection Engineering*, 15(3), 199-229.

Filippoupolitis, A., & Gelenbe, E. (2009). “A distributed decision support system for building evacuation.” *In Human System Interactions, 2009. HSI'09. 2nd Conference,* IEEE, 323-330.

Humblet, P. A. (1988). “An adaptive distributed Dijkstra shortest path algorithm.” Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.

Li, Q., De Rosa, M., & Rus, D. (2003). “Distributed algorithms for guiding navigation across a sensor network.” *In Proceedings of the 9th annual international conference on Mobile computing and networking,* ACM, 313-325

Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). “Introduction to algorithms.” T. H. Cormen (Ed.). The MIT press.

Shi, J., Ren, A., & Chen, C. (2009). “Agent-based evacuation model of large public buildings under fire conditions.” *Automation in Construction*, 18(3), 338-347.

Tang, C. H., Wu, W. T., & Lin, C. Y. (2009). “Using virtual reality to determine how emergency signs facilitate way-finding.” *Applied Ergonomics*, 40(4), 722-730.

Yusoff, M., Ariffin, J., & Mohamed, A. (2008). “Optimization approaches for macroscopic emergency evacuation planning: a survey.” *Information Technology,* IEEE, Vol. 3, 1-7.