Simulation of Optimized Evacuation Processes in Complex Buildings Using Cellular Automata Model

Rong Xie
International School of Software, Wuhan University, Wuhan, China
Email: xierong@whu.edu.cn

Long Li
International School of Software, Wuhan University, Wuhan, China
Email: eric.longlee@gmail.com

***Abstract*—Emergent evacuation is particularly prominent in events of various disasters. However, most existing evacuation models are either missing some human behavioral characteristics in crowds, or are computationally complex. In the paper, a cellular automata (CA) model is proposed to verify validity of evacuation plan. In the model, route is defined by a series of sub-targets. A "distance map" concept is introduced to help in the simulation approach, by which various phenomena including obstacle avoiding etc. are handled efficiently. Occupants move along planning routes gained from optimization algorithm. The simulation results show that our new model is both efficient and potential for evacuation plan.**

***Index Terms*—evacuation plan, cellular automata (CA), evacuation simulation**

I. Introduction

Under rapid development of urbanization progress, more and more large public buildings with complex structure appear, such as shopping center, theater, stadium etc. Crowd in these areas is relatively large-scale and intensive. Thus, occupant evacuation is becoming a particularly prominent problem in events of various disasters (fire, earthquake etc.). Use of advanced computer models for simulation and analysis of evacuation problem in large and complex buildings is becoming an increasingly important research direction.

Occupant evacuation is how to determine an optimal evacuation route for occupants in any evacuation process. People wish to conduct quantitative analysis and quantitative evaluation on evacuation safety performance, which can provide with optimal escape routes and rescue -making when various types of emergency events occur. Currently, researchers and developers have been doing a lot of work to study various phenomena during evacuation process and have made a series of useful research results on the aspects of evacuation modeling, behavioral analysis, evacuation simulation, as well as applications. There have been about more than 30 kinds of evacuation models and corresponding software available (Kuligowski and Peacock 2005), applied to wide applications, such as architecture design, safety performance analysis, crowd management and rescue decision-making etc. Some researchers (Santos and Aguirre 2005, Olenick and Carpenter 2003) give the detailed literature review of the existing evacuation models, respectively. Evacuation models become important sources for understanding evacuation processes and making emergency plan (Machado 2009). There are mainly two objectives for establishment of evacuation models, i.e., (1) In process of building design, to evaluate whether safety performance of building (architecture structure, exit capacity etc.) is reasonable or not. (2) In fire and other disasters, to make an optimal evacuation route so that occupants can spend the least time to reach safety zone. However, most existing evacuation models are either missing some human behavioral characteristics in crowds, or are computationally complex, making evacuation results difficult to comprehensively reflect a more accurate and realistic evacuation process. One hand, previous evacuation models are more focused on simulation aspects, but ignoring individual characteristics, effectiveness of evacuation model is to be verified. On the other hand, some optimization models attempt to use linear programming method to solve complex network problem of evacuation path planning, but computational complexity maybe very high. Therefore, how to understand evacuation phenomenon to be modeled, how to describe evacuation behavioral characteristics and how to optimize evacuation route are the key issues to evaluate whether an evacuation model is successful or not.

To address these problems, we propose an evacuation model and simulation method in this paper to verify the validity of evacuation plan. The model is based on sub-target approach that the entire evacuation space is divided into several small sub-spaces with simple structure. Each sub-space is defined as a certain number of sub-targets, like a door or an exit etc. In process of evacuation, occupants move along the routes identified by such sub-targets. In addition, our model is introduced a "distance

doi:10.4304/jsw.9.6.1428-1434

map" concept to simulate phenomenon that occupants move to avoid obstacles during evacuation.

The rest of the paper is organized as follows. Section 2 discusses related work. An evacuation model is proposed in Section 3, including sub-target, distance map and update rules. In Section 4, the model is applied to simulate different evacuation scenarios to test whether it works as expected. Conclusion and future work are finally given in Section 5.

II. Related Work

Evacuation modeling is critical and complex in the research of occupant evacuation. As it is difficult to quantify human behavior, current research mainly focuses on qualitative analysis or statistical analysis using limited observational data. In order to represent decision-making process, researchers have proposed four solutions as follows, that is, function-based model, rule-based model, agent-based model and cellular automata model.

A function-based model assumes movement of evacuee to be a continuous flow instead of an aggregate of individual with varying physical abilities, human characteristics and movement directions. Some scholars propose to establish mechanical equation for population movement using fluid dynamics and thermodynamics. Helbing et al. (2003) present Maxwell Boltzmann Distribution Law using thermodynamics to establish distribution equation of population movement. Hoogendoom and Bovy (2004) propose a more complex gas dynamic pedestrian model and establish a relationship among density, position and desired speed. The model describes change of crowd density under continuous or non-continuous interaction. This model is microscopic in the sense, but ignoring effects of obstacles and infrastructures on crowd movement. Also, it would cost relatively more calculation time particularly for large-scale crowd. Okasaki and Matsushita (2004) propose that movement of each pedestrian is simulated by motion of a magnetized object in a magnetic field. Magnetic force which acts on a pedestrian from a magnetic pole is basically calculated by equation according to Coulomb's Law. Completely determined by function(s), occupant's movement and behavior will react in a deterministic manner to its influences, and therefore these models will be limited in the same way for each occupant.

Rule-based model assigns individual action and decision based on pre-defined set of rules which can be triggered in specific circumstance or conditions. During the previous studies, decision-making process is in simplistic style. Objects in the model are logically interconnected with each other and with process that could influence their behavior. But the same decision is taken under the same circumstance in a deterministic fashion. This is obviously disadvantage of denying possibility of natural variation. CRISP model (Fraser-Mitchell 2001) improves it and views occupant as an individual by giving occupant certain behavioral roles. These behavioral activities during evacuation are in a probabilistic fashion. buildingEXODUS (Parke et al. 2003) incorporates deterministic and stochastic approaches depending on circumstance, situation and condition.

In the recent years, it tends to use agent technology for behavioral modeling to simulate human and social behavior in emergency evacuation. An agent-based model treats each occupant as an agent and sets a certain set of attributes to the agent. Then, each occupant's movement is determined independently by its own characteristics and environment. Williams (2004) proposes Legion model which views occupant as an intelligent agent with social, physical, and behavioral characteristics that make up a profile for each occupant. Such model can accurately represent decision-making process, but with weak capacity of controlling the model, accuracy of simulation is thus unpredictable. Another agent-based application is called multi-agent, which is proposed to be sufficient to represent complex human behavior and decision making process. Pan et al. (2006) have developed a multi-agent system to model human emergent social behavior, such as competitive, queuing, herding, and to simulate crowd flow behavior through human agents at microscopic level. Murakami et al. (2002) think leader plays important role in many real-world situations. In order to study the role of leadership during evacuation process, they introduce a leadership evacuation simulator system. The system describes behavior of each leader and evacuee and represents the interaction between leader and other occupants. Pelechano and Badler (2006) also simulate a leader agent who is trained and has complete knowledge about the internal connectivity and helps other agent during the evacuation process. The leadership and other human have also been represented by Sugimoto (2005) using multi-agent simulation in a Virtual space.

Cellular automata model is a discrete model which consists of a regular grid of cells, simulating evacuation process by updating states of cells. With different updating rules, occupants show different behaviors. Yang et al. (2002) give the detailed description about cellular automata application. Besides, several works have been attained to study how to make optimization plan for evacuation. Hameacher and Tjandra (2002) give an extensive literature review of linear programming methods to find the optimization solution. Lu et al. (2005) propose a CCRP heuristic algorithm to reduce computational cost when evacuation network is large.

III. An Evacuation Model

When occupant is inside a building, he/she can only see the scope in which he/she is actually limited to the local space. So, we can divide a complex space into some smaller and simpler sub-spaces in the process of evacuation modeling. Different types of sub-space are assigned to different levels of security. For example, area far away from exit is set to a low security level; while exit or area near exit is set to a high security level. Occupant evacuates according to the ordering from a lower level to a higher level. This approach can greatly simplify complexity of evacuation environment modeling.

Some principles of occupant movement in evacuation process are generally as follows. (1) Try to move to the place near exit. (2) Try to move to the area with low crowd density. (3) Try to avoid obstacles on evacuation route.

We propose an evacuation model as follows to gain an evacuation plan including route and schedule for each occupant from the optimization algorithm. The model is a kind of CA model based on sub-target approach that the entire evacuation space is divided into many small sub-spaces with simple structure and each sub-space is defined as a certain number of sub-targets. In process of evacuation, occupant moves along sub-targets till he/she reaches exit. A "distance map" concept is also introduced to simulate the phenomenon that occupant moves to avoid obstacles on evacuation route. On the basis of the model, we can evaluate validity of evacuation plan with some simulation results under various evacuation scenarios. In the proposed CA model, evacuation space is divided into $W \times H$ square grids with the same size, where W, H represents dimension of X axis and Y axis, respectively. Each grid represents a cell. Each cell can be empty, or occupied by an occupant, or by an obstacle. Its size corresponds to $0.4 \times 0.4\text{m}^2$, which is the typical surface occupied by a person in a dense situation (Butstedde 2001). Considering that mean velocity of occupant is usually around 1.3m/s, and raising to 1.8m/s in emergency (Yang 2002; Burstedde 2001), if occupant moves 0.4m at each time-step, the cost will be $0.4/1.8 \approx 0.22$s in emergency.

A. Sub-target

Inside a building, occupant is actually limited to a local space he/she stays, not knowing the situation outside. So, in process of evacuation, the first step for occupant is to leave the space where he/she stays, and then to determine the next step of evacuation plan. At each time instant, occupant is located in a specific space. Before leaving the space, all behavior of occupant is related to this relevant space. According to layout of building, space can be divided into some small sub-spaces with simple structure. Each sub-space is an enclosed space consisting of walls and exits. As for corridors, stairways, halls and other public spaces, they can be modeled as some specific structures with virtual closure of exit to constitute several independent sub-spaces. The junction between sub-space and sub-space is defined as a sub-space exit, which is also the sub-target of the current sub-space.

Considering an example as shown in Figure 1, there are five rooms and one corridor in the floor, each room has a door connecting to a corridor, and corridor connects to two exits. Here, black line represents wall, and green line for sub-target. According to the definition of sub-target, each room is defined as a sub-space, and door as its sub-target. Corridor is divided into three independent sub-spaces by intersection, and connections are defined as sub-target. Two exits are also defined as sub-target.

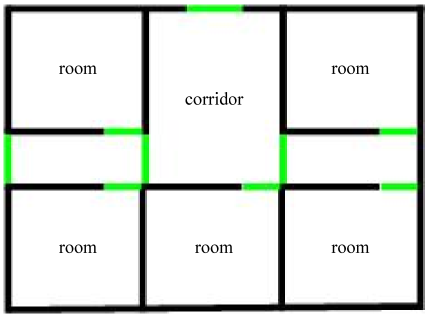


Figure 1. An example of sub-target.

Before evacuation, according to the current global environment and distribution of occupant, select exits and optimal evacuation routes making use of path planning algorithm. Evacuation routes are composed of a series of sequential sub-targets. When evacuation begins, occupant within a sub-space will move to the first pre-specified sub-target. After reaching the current sub-target, occupant will continue moving to the next sub-target until he/she reaches the final exit.

B. Distance Map

Once location of wall and exit are determined, each cell is assigned to a value which represents distance between cell and exit. We define a two-dimensional grid of cells with distance value as "distance map". The unit of distance value is time step. Distance map is generated by the following steps.

Step 1. Identify boundary of sub-space, denoted by (X_{min}, Y_{min}, X_{max}, Y_{max}). Determine each cell as a sub-target with variables (*minDistance*, *available*); where *minDistance* is the shortest distance between the current grid and sub-target, and *available* represents whether it is empty or occupied by an occupant or an obstacle, 0 is occupied; while 1 is empty. Before distance map is generated, *minDistance* of each cell is assigned to a big value, like 99999. *minDistance* of the grid occupied by exit is set to 0, otherwise is 1; *available* of the grid occupied by obstacle is set to 0, otherwise is 1.

Step 2. Establish a grid queue *Queue*<*Cell*>; push all grids occupied by sub-targets into the queue.

Step 3. Remove a cell from the head of queue *Queue*<*Cell*>, named $Cell_{cur}$, and its *minDistance* is named as $minDistance_{cur}$. Then, check all neighborhoods of $Cell_{cur}$ one by one, assuming $Cell_{adj}$ is one of neighborhoods, and *minDistance* of $Cell_{adj}$ is named as $minDistance_{adj}$. If *available* of $Cell_{adj}$ is 1 and $minDistance_{adj} > minDistance_{cur} + 1$, then $minDistance_{adj} = minDistance_{cur} + 1$, and push $minDistance_{adj}$ into the end of queue *Queue*<*Cell*>. Boundary detection is required in order to ensure only cells within the sub-space can be put into calculation.

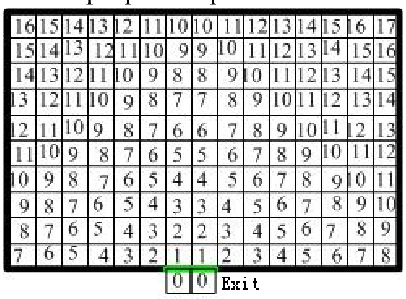
Step 4. Repeat step 3 until *Queue*<*Cell*> is empty, and distance map is finally generated.

Our approach to generating "distance map" is similar to the traditional *Dijkstra* algorithm. Cells, which are not occupied by walls or other obstacles, can be regarded as nodes, connecting each node to its available neighborhood. Weight of edge is assigned to 1. So, generation of "distance map" can be regarded as a process to calculate the shortest distance from the current cell to all other available cells. The implementation code of generation of distance map is shown as follows.

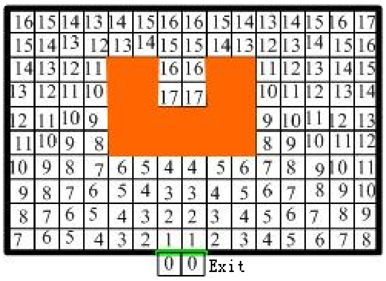
```
// Compute distance from a specified door to all grids in the room.
public void computeDoorDistance(Room room) {
for (Door door : room.getDoors()) {
  int[][] distance = new int[room.getGridHeight()][room.getGridWidth()];
  for (int i = 0; i < room.getGridHeight(); i++)
    Arrays.fill(distance[i], Integer.MAX_VALUE);
 for (int i = 0; i < door.getGridWidth(); i++)
  if (door.isHorizontal() == true)
   distance[door.getGridY() - room.getGridY()][door.getGridX() - room.getGridX() + i] = 0;
  else if (door.isHorizontal() == false)
   distance[door.getGridY() - room.getGridY() + i][door.getGridX() - room.getGridX()] = 0;
 Queue queue = new LinkedList();
 if (door.isHorizontal() == true)
  for (int i = 0; i < door.getGridWidth(); i++)
   queue.add(new Grid(door.getGridX() + i, door.getGridY()));
 else if (door.isHorizontal() == false)
  for (int i = 0; i < door.getGridWidth(); i++)
   queue.add(new Grid(door.getGridX(), door.getGridY() + i));
 while (queue.isEmpty() == false) {
  Grid grid = (Grid) queue.poll();
  for (int i = -1; i <= 1; i++) {
   for (int j = -1; j <= 1; j++) {
    if (i + j == 1 || i + j == -1) {
     if (grid.getGridX() + j >= room.getGridX() && grid.getGridX() + j < room.getGridX()
+ room.getGridWidth()) {
      if (grid.getGridY() + i >= room.getGridY() && grid.getGridY() + i < room.getGridY()
+ room.getGridHeight()) {
       int roomY = grid.getGridY() - room.getGridY();
       int roomX = grid.getGridX() - room.getGridX();
       if (grids[grid.getGridY() + i][grid.getGridX() + j] == true && distance[roomY +
i][roomX + j] > distance[roomY][roomX] + 1) {
        distance[roomY + i][roomX + j] = distance[roomY][roomX] + 1;
        queue.add(new Grid(grid.getGridX() + j, grid.getGridY() + i));}}}}}}}
 if (door.isHorizontal() == true)
  if (door.getGridY() > room.getGridY())
   door.setGridsDistanceUL(distance);
  else if (door.getGridY() <= room.getGridY())
   door.setGridsDistanceDR(distance);
 else if (door.isHorizontal() == false)
  if (door.getGridX() > room.getGridY())
   door.setGridsDistanceUL(distance);
  else if (door.getGridX() <= room.getGridX())
   door.setGridsDistanceDR(distance);
}
}
```

An example of "distance map" is given in Figure 2. Figure 2(a) shows distance map without obstacles. Figure 2(b) shows distance map with obstacles. It is clear that cells behind obstacle with longer distance to sub-target.

By the comparison of Figure 2(a) and Figure 2(b), we can see that grids at the back of concave obstacles are far away from sub-target. Occupant needs to bypass obstacles and move over a long distance to reach the sub-target. Thus, generation of distance map can simulate phenomenon that people escape around obstacles.



(a) "Distance map" without obstacles

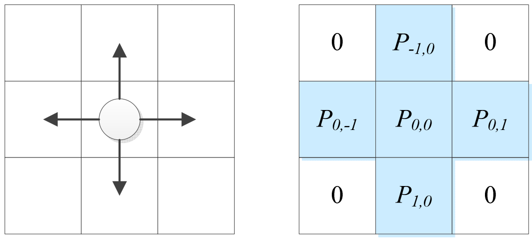


(b) "Distance map" with obstacles (orange polygon represents obstacle)

Figure 2. Examples of "distance map".

C. Update Rules

At each time step, occupant stays at the original position or moves one step. State of each cell is updated in each time step, determined by state of its neighborhoods. At each time step, movement direction of occupant may be up, down, left, right or occupant stays at the original position. As shown in Figure 3(a), it makes sure that no matter how occupant moves right, left, forward or backward, he/she moves the same distance.



(a) Movement direction (b) Cell attractiveness degree

Figure 3. Movement rules.

Cell attractiveness degree is also introduced in our model, i.e. select target cell by calculating their attractiveness degree. Different from some probability evacuation models, occupant always selects the cell which appeals to he/she so much. At the same time step, as shown in Figure 3(b), each cell responding to the five movement directions has different attractiveness degree. In the model, we define basic CA rules as follows.

Rule 1. If there is one $Grid_{adj}$ to meet the requirements among neighbors, that is, the shortest distance to the sub-target is less than the shortest distance to the current cell, i.e. $minDistance_{adj} < minDistance_{cur}$, and $available = 1$, then select $Grid_{adj}$ as target cell. If there are multiple cells to meet the above requirements, then calculate their attractiveness degree and select the most attractive cell as target cell. If there are more than one cell to meet the requirements and attractive degrees are the same, then select randomly a cell as target cell.

Rule 2. If there is no cell among neighbors to meet the requirements of the rule (1), but having cells in compliance with the following requirements, that is, the shortest distance to the sub-target is equal to the shortest distance to the current cell, i.e. $minDistance_{adj} = minDistance_{cur}$, and $available = 1$, then calculate their attractiveness degree, select the most attractive cell as target cell. If more than one cell meet the requirements and attractiveness degree is the same, then give priority to the current cell.

Rule 3. If there is no cell among neighbors to meet the requirements of the rule (1) and rule (2), and also no cell to meet the following requirements, that is, the shortest distance to the sub-target is less than the shortest distance to the current cell minus 1, i.e. $minDistance_{adj} < minDistance_{cur} - 1$, and $available = 1$, but having cells in compliance with the following conditions, that is, the shortest distance to the sub-target $minDistance_{adj}$ is greater than the shortest distance of the current cell $minDistance_{cur}$, and is currently empty. There is a neighbor inside 2 steps of neighborhood whose shortest distance is less than the shortest distance of the current cell, and the neighbor is empty. Select a cell which meets the above requirements. If there is more than one cell to meet the requirements, then randomly select one cell as the target. The rule is to avoid obstruction through the back.

Rule 4. If two or more cells have the same target cell, i.e. they conflict with each other, then select randomly one cell with equal probability to perform movement operation, remaining the others maintain the original position.

IV. SIMULATION EXAMPLES

We develop some functions of evacuation simulation under different evacuation scenarios as follows, such as single sub-space, multiple sub-spaces, space with obstacles and space without obstacles. Here, we take a single mesh size as $0.4 \times 0.4m^2$, one occupant occupying a single grid, personnel speed as 1.8m/s, and each time step as 0.22s.

A. Scenario 1. Evacuation Simulation under Single Sub-space without Obstacles

As shown in Figure 4, area of simulation space is $24 \times 17.2m^2$. There is a sub-target exit in sub-space, with width of two grids. Number of occupants is 47, randomly distributed in the sub-space. Figure 4 shows evacuation results at time instant of 0's, 3's, and 9's, respectively. It takes totally 13 seconds for the whole evacuation. Simulation results show that occupant tends to move to the grids which near to the sub-target and the grids with low population density around.

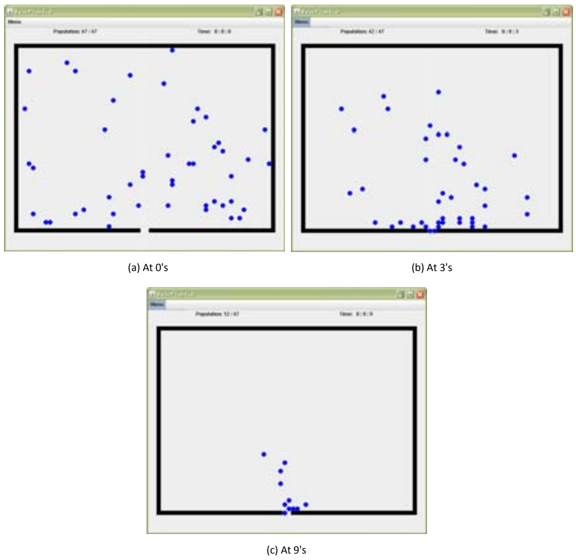


Figure 4. Evacuation simulation under single sub-space without obstacles. (Screen displays of 0's, 3's, and 9's, respectively, with 47 occupants).

As the same as evacuation environment in Figure 4, Figure 5 represents evacuation simulation with 179 occupants, who are randomly distributed in the sub-space. The results show evacuation at time instant of 0's, 4's, and 13's. It takes totally 43 seconds for the whole evacuation. Comparing with the results in Figure 4, there is a clear congestion at the exit in Figure 5 because of a large number of crowds. Occupants at the back of exit have to wait, leading to an obvious increase in spending time. This is consistent with the actual situation.

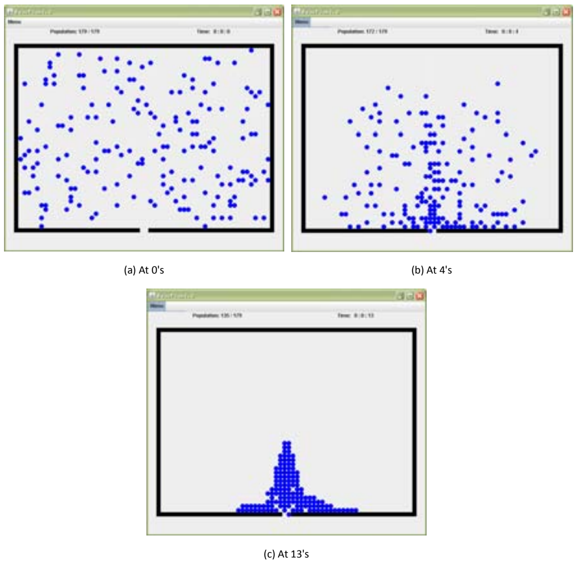
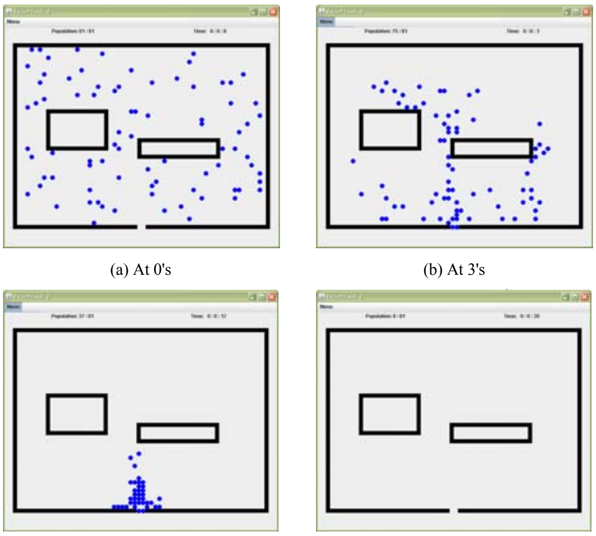


Figure 5. Evacuation simulation under single sub-space without obstacles. (Screen displays of 0's, 4's, and 13's, respectively, with 179 occupants, congestion at the exit).

B. Scenario 2. Evacuation Simulation under single Sub-space with Obstacles

Figure 6 represents evacuation simulation in single sub-space in the absence of obstacles. When there are some obstacles within sub-space, occupant can effectively avoid obstacles according to distance map and select reasonable route. Because of obstacle blocking, occupant needs to bypass obstacles on the way. As a consequence of the increasingly complicated obstacles included in the models, movement distance is increased; simulation times can be also increased. Evacuation simulation results in Figure 6 show that the algorithm can effectively simulate how occupants avoid obstacles in the movement.

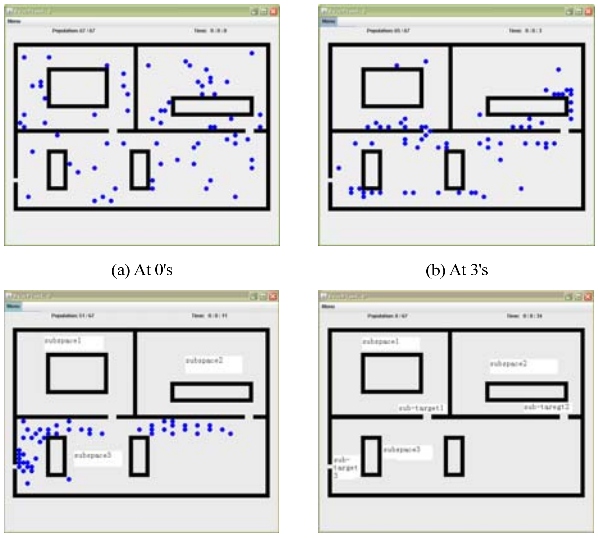


(a) At 0's (b) At 3's (c) At 12's (d) At 20's

Figure 6. Evacuation in single sub-space with obstacles for a 24×17.2m2 area with a door occupying two cells. (Screen displays of 0's, 3's, 12's, 20's, respectively, with 81 occupants).

C. Scenario 3. Evacuation Simulation under Multiple Sub-spaces with Obstacles

Before occupant reaches the final exit, he/she needs to walk through multiple sub-spaces. Path planning obtains optimal path to conduct a global guide. Figure 7 gives a case of evacuation space with three sub-spaces, three sub-targets and several obstacles. When evacuation begins, occupant inside the sub-space 1 firstly sets the exit sub-target 1 of the sub-space 1 as the current sub-target. After reaching the sub-target 1, set the current sub-target as the exit sub-target 3, and implement escape. Occupant inside the sub-space 2 has the similar escaping strategy, i.e. first move to the sub-target 2, and then to the sub-target 3. Evacuation simulation results in Figure 7 shows that the algorithm can effectively evacuation simulate in case of multiple sub-spaces.



(a) At 0's (b) At 3's (c) At 11's (d) At 34's

Figure 7. Evacuation in multiple sub-spaces with obstacles, for a 24×17.2m2 area with three sub-spaces, each sub-space has a door, the final exit occupies one cell, and others occupy two. (Screen displays of 0's, 3's, 11's, 34's, respectively, with 67 occupants).

V. CONCLUSION AND FUTURE WORK

As the existing optimization methods solve optimization path using linear planning, complexity is too high. We propose a cellular automata model to verify validity of evacuation plan in the paper. Evacuation rout is defined by a series of sub-targets, with which occupant moves till reach the final exit. Sub-targets are usually door of sub-space which is gained from dividing the entire evacuation space into some small and simple ones. A "distance map" concept is also introduced in the paper, which is a two-dimensional grid of cells with distance value, by which occupant can select the right cells to move out of the current sub-space in reasonable time. And, obstacles are efficiently handled by generating "distance map". We also develop path planning algorithm, and analyze and verify the planning results by simulating evacuation scenario. The simulation results can be used to evaluate the efficiency of the evacuation planning method. In the simulation examples, situations with single sub-space, multiple sub-spaces, and space with obstacles and space without obstacles are test, respectively. The results show that the new model can simulate the process of evacuation efficiently and has the potential to verify evacuation plan with complex space in building.

However, it is required to test the model under a real evacuation space with complex structure in building. The simulation method in the proposed model can also be improved. Another important future work is the efficiency of different evacuation planning methods should be discussed.

ACKNOWLEDGMENT

This work has been supported by National Nature Science Foundation of China under grant no. 41231171.

REFERENCES

[1] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Physica A: Statistical Mechanics and Its Applications*, vol. 295, pp. 507-525, March-April 2001.

[2] J. Fraser-Mitchell, "Simulated evacuations of an airport terminal building using the CRISP model," 2nd International Symposium in Human Behaviour in Fire, Boston, MA, pp. 89-100, 2001.

[3] H. W. Hamacher, and S. A. Tjandra, "Mathematical modeling of evacuation problems: A state of the art," Pedestrian and Evacuation Dynamics, pp. 227-266, 2002.

[4] D. Helbing, "Lattice gas simulation of experimentally studies evacuation dynamics," *Physical Review E*, vol. 67, June 2003.

[5] S. P., Hoogendoom, and P. H. L. Bovy, "Pedestrian route-choice and activity scheduling theory and models," *Transportation Research Part B*, vol. 38, pp. 169-190, 2004.

[6] E. D., Kuligowski, and R. D. Peacock, "A review of building evacuation models," Technical Note 1471, 2005.

[7] Q., Lu, B., George, and S. Shekhar, "Capacity constrained routing algorithms for evacuation planning: A summary of results," SSTD 2005, LNCS 3633, C. B. Medeiros et al. (eds.), 2005, pp. 291-307.

[8] T. R. Machado, "Evacuation processes versus evacuation models: "Quo vadimus"?," *Fire Technology*, vol. 45, pp. 419-430, April 2009.

[9] Y., Murakami, K., Minami, T., Kawasoe, and T. Ishida, "Multi-agent simulation for crisis management," Proceedings of the IEEE Workshop on Knowledge Media Networking, IEEE, Los Alamitos, pp. 135-139, 2002.

[10] S., Okasaki, and S. Matsushita, "A study of simulation model for pedestrian movement with evacuation and queuing," Engineering for Crowd Safety, pp. 271-280, 2004.

[11] S. M., Olenick, and D. J.Carpenter, "An updated international survey of computer models for fire and smoke," *Journal of Fire Protection Engineering*, vol. 13, pp. 87-110, 2003.

[12] X., Pan, C. S., Han, K., Dauber, and K. H. Law, "Human and social behavior in computational modeling and analysis of egress," *Journal of automation in construction*, vol. 15, pp. 448-461, 2006.

[13] J., Parke, S., Gwynne, E. R., Galea, and P. Lawrence, "Validating the buildingEXODUS evacuation model using data from an unannounced trial evacuation," University of Greenwich, UK: CMS Press, 2003, pp. 295-306.

[14] N., Pelechano, and N. I. Badler, "Modelling crowd and trained leader behavior during building evacuation," IEEE Computer Society, vol. 26, pp. 80-85, Nov.-Dec. 2006.

[15] G., Santos, and B. E. Aguirre, "A critical review of emergency evacuation simulation models," Proceedings of Workshop on Building Occupant Movement during Fire Emergencies, 2005, pp. 27-52.

[16] Y. Sugimoto, "Modeling action rules through participatory simulation in virtual space," Master Thesis, Kyoto University, Japan, 2005.

[17] A. Williams, "Go with the flow," *The Architects' Journal*, vol. 219, pp. 42-46, June 2004.

[18] L., Yang, W., Fang, R., Huang, and Z. Deng, "Occupant evacuation model based on cellular automata in fire," *Chinese Science Bulletin*, vol. 47, pp. 1484-1488, 2002.