A Stage-Wise Path Planning Approach for Crowd Evacuation in Buildings

Xuechao Chen
Sichuan University
Chengdu, China

Wenjing Li
Sichuan University
Chengdu, China

Yu Nie
Sichuan University
Chengdu, China

Rui Chen
Sichuan University
Chengdu, China

Kangben He
Sichuan University
Chengdu, China

Yanli Liu
Sichuan University
Chengdu, China

Yanci Zhang*
yczhang@scu.edu.cn
Sichuan University
Chengdu, China

ABSTRACT

We propose a new path planning approach for crowd evacuation in buildings. Based on the crowd distribution at current moment, our approach is capable of predicting congestion at intersections in the near future so that a more reasonable path can be achieved than previous method. We also introduce a stage-wise path planning mechanism to adjust routes to respond to the dynamic changes of crowd distribution. Experimental results on several evacuation scenarios indicate that our method can reduce congestion comparing with previous method so that more reasonable evacuation routes can be achieved and the evacuation time is shortened.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent planning**; **Agent / discrete models**; *Animation.*

KEYWORDS

crowd evacuation, path planning, crowd distribution

ACM Reference Format:
Xuechao Chen, Wenjing Li, Yu Nie, Rui Chen, Kangben He, Yanli Liu, and Yanci Zhang. 2019. A Stage-Wise Path Planning Approach for Crowd Evacuation in Buildings. In *Computer Animation and Social Agents (CASA '19), July 1–3, 2019, PARIS, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3328756.3328757

1 INTRODUCTION

Concerts, schools, shopping malls, etc., tend to gather high density crowds. When an emergency situation occurs, crowds need to be evacuated safely and efficiently. Designing and validating an evacuation plan via computer simulation has attracted much attention of researchers, such as cellular automata model, social force model, etc.[10, 15].

The major goal of designing a crowd evacuation plan is to avoid congestion so that crowd can be evacuated as soon as possible. Using a fixed evacuation plan is apparently not a good idea because it ignores the crowd distribution which keeps changing during the whole evacuation process. Thanks to the development of sensor techniques, now it is possible to retrieve the information of crowd distribution in real time [12]. Thus, the main challenge of designing an efficient evacuation plan is to achieve a congestion avoidance path planning based on dynamic crowd distribution.

Recent works treat the route planning as a global optimization problem. For example, [9] applies improved artificial bee colony algorithm to plan a global optimized evacuation route based on the congestion of different exits. [16] refines the routes based on multiple crowd simulation in an iterative manner. Unfortunately, a global optimization path planning algorithm could be affected by the uncertainty of scenario and crowds.

This paper proposes a congestion avoidance path planning algorithm to efficiently evacuate a crowd in a complex indoor space. Our method is capable of predicting the crowd distribution in the near future to avoid potential congestion at intersections. We also note an important fact that the prediction accuracy will drop rapidly for the moment far ahead of present moment because too many unpredictable factors may influence the behaviour of crowd. In order to address this issue, we present a stage-wise strategy to update the planned route several times during the process of crowd evacuation so that it is not necessary to predict congestion in the far future.

2 RELATED WORK

Crowd Evacuation [3, 17, 18] under emergency situation has been studied over the last decades, which aims to evacuate people efficiently and safely. Many existing models can be used in the crowd evacuation problems, like dynamic flows [13], maximum dynamic flows [5], universal maximum flows [1], quickest path and flows [4] and simulation [14].

CASA '19, July 1–3, 2019, PARIS, France

ACM ISBN 978-1-4503-7159-9/19/07…$15.00
https://doi.org/10.1145/3328756.3328757

Path planning is one of the most important factors in crowd evacuation. Mubbasir et al. [7] presented a real-time planning framework for multi-character navigation that enables the use of multiple heterogeneous problem domains of differing complexities for navigation in large, complex, dynamic virtual environments. Qingsong et al. [11] proposed a capacity constrained method (CCRP), which models capacity as a time series and uses a capacity constrained routing approach to incorporate route capacity constraints. Based on CCRP, Gopinath Mishra et al. [8] explored evacuation optimization and planning of evacuation routes. Vania Campos et al. [2] proposed a heuristics algorithm which applies analytical techniques to define a set of optimal routes and evaluate performance measures simultaneously. To gain an optimized evacuation route, San-Keung Wong et al. [16] initially set the evacuation direction based on the shortest path algorithm and iteratively updated their predefined division points and distribution ratio by repeating simulation processes.

3 ALGORITHM OVERVIEW

3.1 Problem Formulation

As illustrated in Fig.2, our method models the environment as a graph $G = (V, E)$, where vertex $v \in V$ is defined as the intersection point of multiple paths (green triangles in Fig.2) and E is the set of edges. Additionally, all exits of building (green boxes in Fig.2) are also defined as vertices of G. Supposing that t^i is the time of individual i moving from his initial position p^i to exit $\mathcal{T}^i$, the goal of our method is to compute routes for all individuals in crowd C so that the largest t^i is minimized, as formulated in Eq.1.

$$t(\mathrm{C}) = \arg\min \max_{i \in \mathrm{C}} t^i(p^i, v_0^i, v_1^i, ..., \mathcal{T}^i) \tag{1}$$

where $(p^i, v_0^i, v_1^i, ..., \mathcal{T}^i)$ defines the path of i, and $v_k^i \in V$ is the k^{th} vertex on the route.

3.2 Basic Idea of Our Method

As illustrated in Fig.1a, shortest path strategy is often not optimal for the large-scale crowd evacuation problem. This is because shortest path strategy may cause congestion in the evacuation process.

Our approach tries to find a sub-optimal solution of $t(\mathrm{C})$ by predicting and avoiding congestion in the evacuation process. As illustrated in Fig.1b, in order to plan a proper route for individual a, the future crowd distribution on path P_1 and P_2 should be estimated. On the other hand, predicting crowd distribution at any future moment based on the crowd distribution at current moment is very hard because of too many uncertainties. A reasonable solution is to only estimate the crowd distribution in the near future, which causes another problem that is how to determine a proper threshold ξ to define the near future. Comparing the situations illustrated in Fig.1b and Fig.1c, Fig.1c requires a bigger ξ than Fig.1c to estimate the future crowd distribution on P_2.

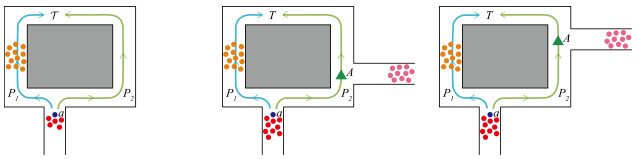


Figure 1: Impact of crowd distribution in path planning.

Instead of finding a global ξ, we assume that the major cause of congestion is the intersection of multiple streams of people which only happens at the intersections of different paths. Thus, we only consider the congestion occurring at $v \in V$. Based on this idea, the problems of estimating future crowd distribution on P_2 in Fig.1b and Fig.1c are unified to predict crowd distribution at point A at the moment when a reaches A.

In this paper, the next vertex in V that agent i will reach is called as Next Target Point(NTP). The definition of NTP makes each individual have his own ξ to define the near future. Another important function of NTP is to introduce a stage-wise mechanism to plan the whole route.

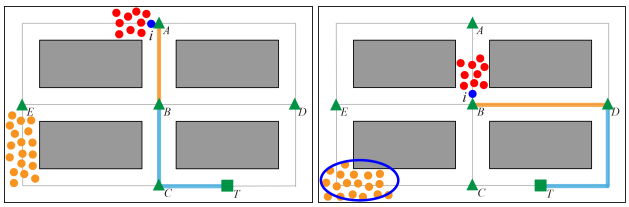


Figure 2: The basic idea of our method.

As shown in Fig.2, we take agent i as an example to make the above idea clear:

(1) In the first stage, i moves in the initial direction. At the beginning moment, the shortest path is the best path for i. B is NTP of the route.
(2) In the remaining stages, each time i reaches its NTP, a re-planning process is triggered to decide the new NTP. As illustrated in Fig.2b, when i reaches B, there are three possible vertices(C,D,E) to be the new NTP for i. Based on the dynamic crowd distribution, our algorithm is capable of predicting the congestion at these three intersections in the near future. By taking into account the congestion at intersections and distance, D is chosen as the new NTP and a new route $B \rightarrow D \rightarrow T$ is generated.

4 IMPLEMENTATION DETAILS

4.1 Estimation of Time Spent on Path

As formulated in Eq.2, the time $t(v_{cr}, \mathcal{T}, \mathbf{t})$ spent on the path from vertex v_{cr} to target $\mathcal{T}$ is divided into two parts by NTP.

$$t(v_{cr}, \mathcal{T}, \mathbf{t}) = t_1(v_{cr}, v_{NTP}, \mathbf{t}) + t_2(v_{NTP}, \mathcal{T}, \mathbf{t}) \tag{2}$$

where v_{NTP} is the vertex corresponding to NTP, $t_1(v_{cr}, v_{NTP}, \mathbf{t})$ and $t_2(v_{NTP}, \mathcal{T}, \mathbf{t})$ represent the time spent on moving from v_{cr} to v_{NTP} and from v_{NTP} to $\mathcal{T}$ respectively. Note that path from v_{NTP} to $\mathcal{T}$ is normally composed of more than one edge.

The formulas to compute $t_1(v_{cr}, v_{NTP}, \mathbf{t})$ and $t_2(v_{NTP}, \mathcal{T}, \mathbf{t})$ are as represented in Eq.3 and Eq.4 respectively.

$$t_1(v_{cr}, v_{NTP}, \mathbf{t}) = \omega(\mathbf{CDF}(v_{cr}, \mathbf{t}))\cdot \tau(v_{cr}, v_{NTP}) \tag{3}$$

$$t_2(v_{NTP}, \mathcal{T}, \mathbf{t}) = \omega(\mathbf{CDF}(v_{NTP}, \mathbf{t}'))\cdot \tau(v_{NTP}, \mathcal{T}) \tag{4}$$

where $\mathbf{CDF}(v, \mathbf{t})$ describes the crowd distribution of the neighborhood of vertex v at moment $\mathbf{t}$, $\mathbf{t}' = \mathbf{t} + t_1(v_{cr}, v_{NTP}, \mathbf{t})$ is the estimated moment that individual i arrives at NTP, $\tau(v_1, v_2)$ is the estimated time that i moves from vertex v_1 to v_2 without consideration of the crowd distribution around v_1, $\omega()$ is a function mapping the value of CDF to a coefficient greater than or equal to 1.

Both Eq.3 and Eq.4 can be explained as enlarging the time spent on the path by a weight determined by the CDF at the start point of the route. Fig.3 demonstrates this idea. The crowd distribution on the edge AB demonstrated in Fig.3a and Fig.3b is similar, but it is clear that i in Fig.3b needs more time to reach B than Fig.3a because point A in Fig.3b is much more crowded.

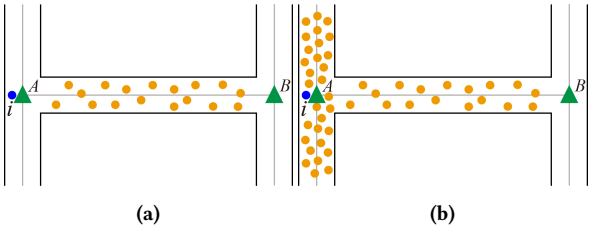


Figure 3: Time cost on edge *AB* influenced by the crowd distribution at *A*.

4.2 Estimation of CDF

According to Eq.2, Eq.3 and Eq.4, the key of our method is to predict $\mathbf{CDF}(v_{NTP}, \mathbf{t}')$ which describes the crowd distribution of the neighborhood of NTP at a future moment $\mathbf{t}'$.

Without loss of generality, let's take vertex A in Fig.4 and a set of individuals $C_A = \{i_1, i_2, ..., i_8\}$ whose paths contain point A as an example to demonstrate the prediction of $\mathbf{CDF}(A, \mathbf{t}')$ at the present moment $\mathbf{t}$. In order to predict $\mathbf{CDF}(A, \mathbf{t}')$, the arrival moments of all individuals in C_A need to be estimated. Due to the fact that it is hard to accurately estimate the arrival moment of $i_k (k = 1, 2, ..., 8)$, our method treats it as a random variable x which satisfies the Gaussian distribution $\mathbb{G}(\mathbf{t}'_{i_k}, \sigma^2_{i_k})$ with probability density function $f_{i_k}(x)$, which is as illustrated in Eq.5.

$$f_{i_k}(x) = \frac{1}{\sqrt{2\pi}\sigma_{i_k}} e^{-\frac{(x-\mathbf{t}'_{i_k})^2}{2\sigma^2_{i_k}}} \tag{5}$$

where $\mathbf{t}'_{i_k}$ is the expected value of arrival moment at A of i_k (see Sec.4.3 for more details on the computation of $\mathbf{t}'_{i_k}$). σ_{i_k} is the standard deviation of x, which is in positive correlation with $\mathbf{t}'_{i_k} - \mathbf{t}$. In our method, σ_{i_k} is defined as Eq. 6.

$$\sigma_{i_k} = \frac{\ln(\mathbf{t}'_{i_k} - \mathbf{t} + 1)}{3} \tag{6}$$

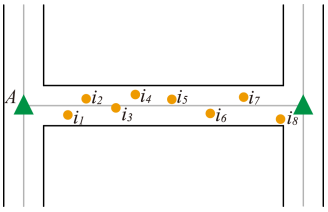


Figure 4: Individuals $i_1, i_2, ..., i_8$ who will reach *A* in the near future.

Fig.5 shows the corresponding Gaussian distributions for all individuals in C_A. It can be seen that for individuals who are closer to A, the probabilities of arriving at A at moment $\mathbf{t}'_{i_k}$ are higher.

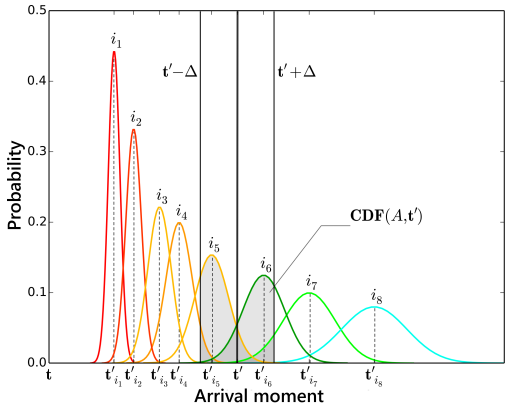


Figure 5: Estimation of $\mathbf{CDF}(A, \mathbf{t}')$.

Based on the above discussion, $\mathbf{CDF}(A, \mathbf{t}')$ is defined as Eq.7:

$$\mathbf{CDF}(A, \mathbf{t}') = \sum_{i_k \in C_A} \int_{\mathbf{t}'-\Delta}^{\mathbf{t}'+\Delta} f_{i_k}(x)dx \tag{7}$$

Note that according to Eq.7, $\mathbf{CDF}(A, \mathbf{t}')$ actually defines the crowd distribution at the neighborhood of A instead of the exact position at A. Δ in Eq.7 is used to control the size of neighborhood and a bigger Δ indicates larger neighborhood. In our experiments, we assume the diameter of the size of neighborhood is equal to the width of the road and set Δ to 30.

$\mathbf{CDF}(A, \mathbf{t}')$ will be converted to a coefficient greater than or equal to 1 by a mapping function $\omega()$, as illustrated in Eq.8.

$$\omega(\mathbf{CDF}(A, \mathbf{t}')) = \lambda^{\frac{\mathbf{CDF}(A,\mathbf{t}')}{N}} \tag{8}$$

where N is the normalization constant required to ensure that the exponent term is between 0 and 1, and λ defines the upper bound on the coefficient. In the latter experiments, we set $N = 60$ and $\lambda = 2.5$.

4.3 Computation of $\tau(v_1, v_2)$

A* algorithm [6] is employed in our method to compute the path that incurs the minimum time cost, recorded as $\tau(v_1, v_2)$, from vertex v_1 to v_2. Since A* algorithm works on a weighted graph, the weight for edge $e \in E$ is defined as a time cost c_e which is formulated in Eq. 9:

$$c_e = \begin{cases} \frac{|e|}{s_{avg}} & m > 0 \\ \\ \frac{|e|}{s_{max}} & m = 0 \end{cases} \tag{9}$$

where $|e|$ is the length of e, m is the number of individuals on e at current moment and s_{avg} is the average speed of these individuals, s_{max} is the maximum speed of an individual.

Supposing $C_e = \{i_1, i_2, ..., i_m\}$ are m individuals on the edge e at current moment, s_{avg} can be computed by Eq.10:

$$s_{avg} = \frac{\sum_{k=1}^{m} (\vec{v_{i_k}} \cdot \vec{e})}{m} \tag{10}$$

where $\vec{v_{i_k}}$ is the velocity of individual $i_k \in C_e$, $\vec{e}$ is the normalized direction of edge e.

4.4 Replanning Stage

In the replanning stages, when individuals reach their own NTPs, our replanning computation will be triggered. As illustrated in Fig.2b, when i reaches B, the path will be recomputed, which can be viewed as a process of determining a new NTP so that Eq.2 is minimized.

Algorithm 1 Path replanning.

```
1:  procedure ReplanPath(v_cr, t, T)
2:      v ← v_cr
3:      t_min ← FLT_MAX
4:      NTP ← NULL
5:      for each v' connected to v do
6:          τ(v, v') ← EstimateTime(v, v')
7:          τ(v', T) ← EstimateTime(v', T)
8:          w_1 ← ω(CompCDF(v, t))
9:          t_1(v, v', t) ← w_1 · τ(v, v')
10:         w_2 ← ω(CompCDF(v', t + t(v, v', t)))
11:         t_2(v', T, t) ← w_2 · τ(v', T)
12:         t(v, T, t) ← t_1(v, v', t) + t_2(v', T, t)
13:         if t(v, T, t) < t_min then
14:             t_min ← t(v, T, t)
15:             NTP ← v'
16:         end if
17:     end for
18: end procedure
```

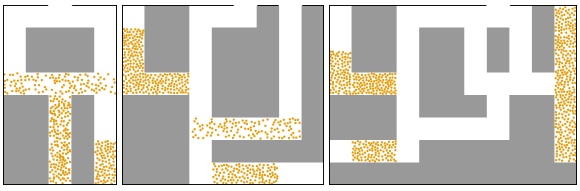
When individual i arrives at vertex v_{cr} at moment $\mathbf{t}$, the pseudo-code of the replanning process is as illustrated in Alg.1. For each vertex v' connecting to the current vertex v, line 5 to 17 evaluate the time cost of traversing from v to target $\mathcal{T}$ through v'. **EstimateTime**(v_1, v_2) computes the minimum time cost from vertex v_1 to v_2 by the method described in Sec.4.3. **CompCDF**$(v, \mathbf{t})$ is used to estimate crowd distribution of vertex v at moment $\mathbf{t}$, which is specified in Sec.4.2. Finally, the new route is updated as the path with the smallest overall time cost from v_{cr} to target could be acquired and a new NTP is generated as well.

5 EXPERIMENTAL RESULTS

The algorithm proposed in this paper has been implemented on a computer with a 3.07GHz Intel Core i7-950 processor and 12GB RAM. We also implement another non-dynamic crowd evacuation algorithm recently proposed in [16] as well as the traditional Dijkstra's shortest path algorithm. [16] is a typical non-dynamic optimization method. The basic idea of [16] is to update a set of predefined division points and distribution ratios in an iterative manner according to simulation result until the convergence state is achieved.

5.1 Overall Results

Three different scenes are employed (as illustrated in Fig.6) to compare the three algorithms mentioned above. The time costs defined in Eq.1 achieved by these three algorithms are as shown in Table.1. Thanks to the ability to predict congestion in the near future and the stage-wise mechanism, our path planning method can save approximately 9.5% of time comparing with [16].



(a) Scene with single exit. **(b) Scene with multi-ple exits.** **(c) Scene with multiple intersections and streams of crowds.**

Figure 6: Three test scenes employed to compare our algorithm with [16] and the shortest path algorithm. The number of agents in the scenes is 390, 500 and 650, respectively.

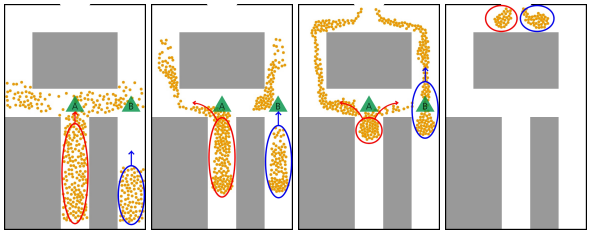
Method	Scene(a)	Scene(b)	Scene(c)
Shortest	1427	1548	1651
Non-dynamic	1153	1439	1651
Ours	**1075**	**1338**	**1453**

Table 1: Comparison between the overall time cost achieved by our method, [16] and the shortest path algorithm.

5.2 Scene with Single Exit

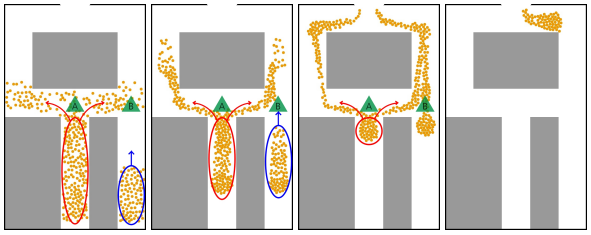
As illustrated in Fig.7a, two groups of agents, circled in red and blue, are about to escape through the same exit. When t = 120, with the prediction that agents in the blue circle will reach *B* soon, our algorithm guides agents arriving at *A* to the left path to avoid congestion. When t = 350, most of the agents circled in blue are moving away from *B*, so that *B* will be uncrowded soon. Our algorithm guides part of agents arriving at *A* to the right path. When t = 960, with the effort of avoiding potential congestion, two groups of agents circled in red and blue reach the exit at almost the same moment.

Fig.8 shows the crowd evacuation process of algorithm [16]. As stated above, [16] optimizes division points and distribution ratios. This implies a fact that once division points and distribution ratios are determined, they never get updated during the whole evacuation process. As demonstrated in Fig.8a,8b,8c, No matter the moment that agents in the blue circle arrive at *B*, agents in the red circle are guided to different routes at a fixed distribution ratio. Lacking the ability of dynamic path guidance, [16] leads to the situation that agents on the right path arrive at the exit more slowly than agents on the left path.



(a) t=20 (b) t=120 (c) t=350 (d) t=960

Figure 7: Crowd evacuation process of our algorithm in scene(a).



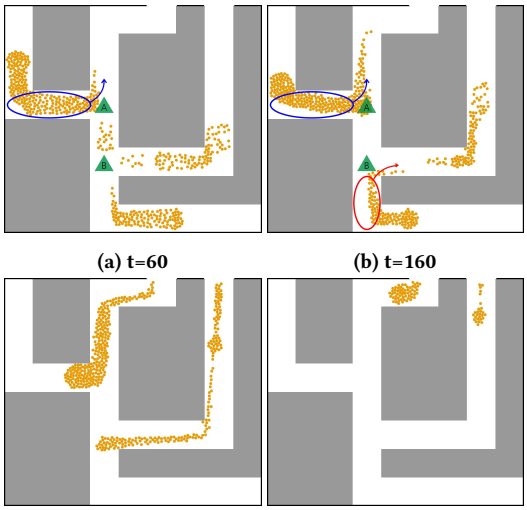
(a) t=20 (b) t=120 (c) t=350 (d) t=960

Figure 8: Crowd evacuation process of [16] in scene(a).

5.3 Scene with Multiple Exits

We also conduct experiments on evacuation scene with multiple exits. As shown in Fig.9b, when agents in the red circle reach *B*, our algorithm detects that a number of agents, circled in blue, will reach *A* in the near future. As a result, our algorithm chooses another exit for agents reaching *B* though path toward this exit is longer. Thus, the capacity of the two exits is fully utilized.

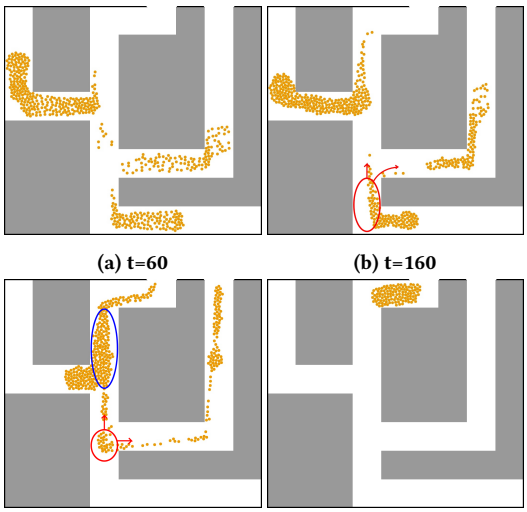
Similar to scene(b), [16] divides agents in the red circle into two exits at a fixed distribution ratio, as shown in Fig.10. But agents circled in blue block the path toward the left exit and slow down the movement.



(a) t=60 (b) t=160

(c) t=510 (d) t=1200

Figure 9: Crowd evacuation process of our algorithm in scene(b).



(a) t=60 (b) t=160

(c) t=510 (d) t=1200

Figure 10: Crowd evacuation process of [16] in scene(b).

5.4 Complex Scene

Scene(c) shows a more complex case with multiple intersections and streams of agents. Our stage-wise replanning mechanism plays a crucial role in dealing with such a complex scene. As illustrated in Fig.11, when t = 130, aware of congestion at *A* in the near future,

agents circled in red turn to the right path to avoid congestion. When t = 450, those agents reach the next intersection and our stage-wise replanning mechanism is triggered. Detecting that the right path from B to exit is crowded at the current moment and even in the near future, our algorithm guides them to the left path. This complex scene shows that our algorithm is equipped with the ability to adjust routes during the dynamic evacuation process.

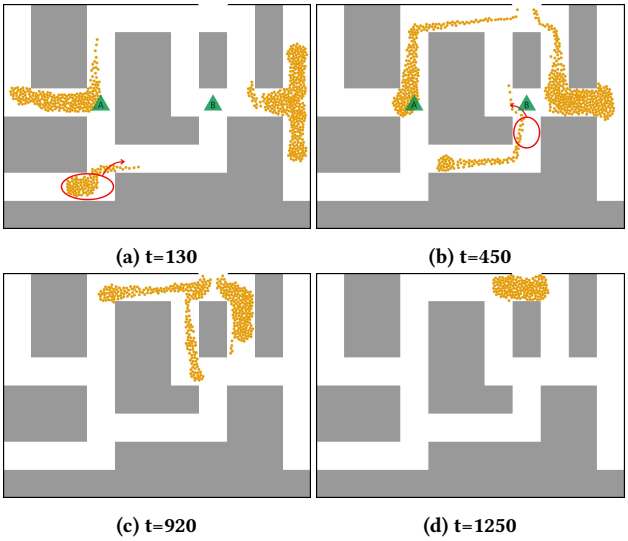


Figure 11: Crowd evacuation process of the our algorithm in scene(c).

[16] fails to handle this kind of situation and degenerates into the shortest path algorithm. Because it computes distribution ratio based on evacuation time of agents on outward edges connected to intersections. In this scene, most of the edges are empty so that

Figure 12: Crowd evacuation process of the [16] in scene(c).

poor information is obtained for updating the routes in the iterative simulation process. As shown in Fig.12, algorithm [16] plans the shortest path for agents and incur longer evacuation time.

6 CONCLUSION

In this paper, we focus on dynamic route guidance for crowds when an evacuation occurs. A stage-wise path planning algorithm for crowd evacuation is proposed to solve this problem. The goal is to avoid congestion and minimize the overall time cost for evacuees escape from building. By predicting the congestion at intersections in the near future, our algorithm guides crowds to more reasonable paths. We also note the fact that the congestion in the far future is hard to be accurately estimated. A stage-wise replanning mechanism is proposed to update the routes when agents reach their NTPs. Based on the above mechanisms, a congestion-avoiding path can be produced for each agent.

REFERENCES

[1] X Cai, D Sha, and CK Wong. 2001. Time-varying universal maximum flow problems. *Mathematical and Computer Modelling* 33, 4-5 (2001), 407–430.
[2] Vania Campos, Renata Bandeira, and Adriano Bandeira. 2012. A method for evacuation route planning in disaster situations. *Procedia-Social and Behavioral Sciences* 54 (2012), 503–512.
[3] Xiaodong Che, Yu Niu, Bin Shui, Jianbo Fu, Guangzheng Fei, Prashant Goswami, and Yanci Zhang. 2015. A novel simulation framework based on information asymmetry to evaluate evacuation plan. *The Visual Computer* 31, 6-8 (2015), 853–861.
[4] Lisa Fleischer and Martin Skutella. 2007. Quickest flows over time. *SIAM J. Comput.* 36, 6 (2007), 1600–1630.
[5] Lisa K Fleischer. 2001. Faster algorithms for the quickest transshipment problem. *SIAM journal on Optimization* 12, 1 (2001), 18–35.
[6] Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.
[7] Mubbasir Kapadia, Alejandro Beacco, Francisco Garcia, Vivek Reddy, Nuria Pelechano, and Norman I Badler. 2013. Multi-domain real-time planning in dynamic environments. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics symposium on computer animation*. ACM, 115–124.
[8] Guo Li, Lijun Zhang, and Zhaohua Wang. 2014. Optimization and planning of emergency evacuation routes considering traffic control. *The Scientific World Journal* 2014 (2014).
[9] Hong Liu, Bin Xu, Dianjie Lu, and Guijuan Zhang. 2018. A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm. *Applied Soft Computing* 68 (2018), 360–376.
[10] Lili Lu, Ching-Yao Chan, Jian Wang, and Wei Wang. 2017. A study of pedestrian group behaviors in crowd evacuation based on an extended floor field cellular automaton model. *Transportation research part C: emerging technologies* 81 (2017), 317–329.
[11] Qingsong Lu, Yan Huang, and Shashi Shekhar. 2003. Evacuation planning: a capacity constrained routing approach. *Intelligence and Security Informatics* (2003), 959–959.
[12] Marin Lujak, Holger Billhardt, Jürgen Dunkel, Alberto Fernández, Ramón Hermoso, and Sascha Ossowski. 2017. A distributed architecture for real-time evacuation guidance in large smart buildings. *Comput. Sci. Inf. Syst.* 14, 1 (2017), 257–282.
[13] Martin Skutella. 2009. An introduction to network flows over time. In *Research trends in combinatorial optimization*. Springer, 451–482.
[14] Daniel Thalmann. 2016. Crowd simulation. *Encyclopedia of Computer Graphics and Games* (2016), 1–8.
[15] Hendrik Vermuyten, Jeroen Beliën, Liesje De Boeck, Genserik Reniers, and Tony Wauters. 2016. A review of optimisation models for pedestrian evacuation and design problems. *Safety science* 87 (2016), 167–178.
[16] Sai-Keung Wong, Yu-Shuen Wang, Pao-Kun Tang, and Tsung-Yu Tsai. 2017. Optimized evacuation route based on crowd simulation. *Computational Visual Media* 3, 3 (2017), 243–261.
[17] Ruilin Xie, Zhicheng Yang, Yu Niu, and Yanci Zhang. 2016. Simulation of Small Social Group Behaviors in Emergency Evacuation. In *Proceedings of the 29th International Conference on Computer Animation and Social Agents*. ACM, 71–77.
[18] Xiaoping Zheng, Tingkuan Zhong, and Mengting Liu. 2009. Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment* 44, 3 (2009), 437–445.