Procedia Computer Science
Volume 51, 2015, Pages 2759–2763
ICCS 2015 International Conference On Computational Science

Fire Hazard Safety Optimization

Sabin Tabirca[1]*, Laurence T.Yang[2], Tatiana Tabirca[1]

[1]Department of Computer Science, University College Cork, Cork, Ireland
[2]Department of Computer Science, St Xavier University, Antigonish, Canada

Abstract
This article provides a study for fire hazard safety in building environments. The working hypothesis is that the navigation costs and hazard spread are deterministically modeled over time. Based on the dynamic navigation costs under fire hazard, the article introduces the notion of dynamic safety in a recursive manner. Using the recursive equations, an algorithm is proposed to calculate the dynamic safety and successor matrices.

Keywords: Fire Hazard Modeling; Fire Safety; Computational Models for Safety.

1 Introduction

Building safety under fire hazard has been an important problem over the last few decades in order to make the evacuation during hazards more effective. Various evacuation schemes have been developed, which consider either how the hazard spreads on building environment (Desmet *et al.*, 2013), (Kuligowski *et al.,* 2005) or how evacuees behave during hazard evacuation (Kuligowski *et al.,* 2005), (Kuligowski*,* 2013). All these research efforts have considered some optimization criteria like shortest paths to minimize the evacuation times or maximum flows to avoid crowd congestions or various crowd congestion algorithms. We consider a building environment, which is monitored for fire hazards using some fire sensors. We suppose that the hazard starts at some locations and it spreads through the building environment. The hazard can be generated by fire, explosions or earthquake and involves a combination of fire, smoke, gases, *etc*. Under this scenario some important practical problems can be considered related to the safety of inhabitants, their evacuation and perhaps with fire fight management.

Safety concepts or safety similar notions appeared first in research connected to the topics of mobile sensors or wireless sensors for fire emergency. Li *et al.* started from the observation that the smallest numbers of hops to a sensor that triggers "Yes-Danger" represents a measure of the distance to the danger area (Li *et al.*, 2003). This idea of potentials was re-used by Barnes *et al.* to propose the first formal definition of safety in the static case (Barnes *et al.*, 2007). Barnes *et al.* considered that the safety of a path must give the maximum amount of time a person can delay at the starting node and

* Corresponding author. Tel.: +353 21 4205915; fax:+353 21 420 5367.
E-mail address: tabirca@cs.ucc.ie

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.425

still have a safe walk along the path. This was then translated into a recursive definition for path safety and node safety. Tabirca *et al.* extended the safety concept introduced by Barnes *et al.* to a dynamic safety concept for the all-to-one case (Tabirca *et al.,* 2009). In this case, the safety values depend explicitly on time to reflect the dynamic change in the network. As Tabirca *et al.* pointed the notion of dynamic safety is strongly connected to the notion of dynamic shortest paths. The dynamic shortest path problem is a classical topic in combinatorial optimization being investigated since late 60-s e.g (Cooke *et al.,* 1966). The problem became lately very popular with the emergency of Intelligent or Dynamic Transportation Systems e.g. (Chabini, 1997), (Chabini, 1998), (Ahuja *et al.*, 2002) or (Demetrescu *et al.,* 2006). Two well-established approaches have been developed to solve this problem, which are quite different in essence. The first approach considers dynamic equations for the numbers $d^{(t)}(u,v)$ to reflect the evolution in time of the shortest paths costs. The particularity of these equations is that the time variable *t* is essential in their dynamicity. These equations are then processed using dynamic programming techniques mainly in a retrospective way (Chabini, 1998). The second way to solve the problem, which has been intensely investigated lately, is to preserve the shortest paths when changes take place in the graph *e.g.* removing a node or an arc or changing an arc's cost. This approach was introduced by (Ramalingam *et al.*, 1996) and then refined successively by various contributors see (Demetrescu *et al.,* 2006) for a complete review.

2 Dynamic Safety for Fire Safety

This section presents a theoretical model for dynamic safety in building environments under fire hazard. Firstly, we assume that the building environment has a *navigation graph* $G=(V,A)$, where $V=\{s_1,s_2,...,s_n\}$ is the set of nodes representing rooms, corridor extremities *etc.* and $A=\{(u,v):\text{the nodes } u \text{ and } v \text{ are directly connected}\}$ is the set of arcs. This navigation graph represents the topology of the building that can be used for evacuation. We can also consider a *hazard graph* $G'=(V,A')$, where $A'=\{(u,v):\text{the hazard spreads from } u \text{ to } v \text{ directly}\}$. It may be possible that the hazard can spread between two nodes that are not adjacent for navigation *e.g.* from a room to the neighbor room. Suppose that there are multiple exit nodes $\{e_1,e_2,...,e_p\}$ in the navigation graph. The following functions are considered to be known for the graphs *G* and *G*' (for simplicity we work with the same notations as in (Barnes *et al.*, 2007) and (Tabirca *et al.*, 2009):

1. the hazard function $F:A'\to[0,\infty)$, $F(u,v)$ = the time taken for the fire to spread from *u* to *v,*
2. the navigation function $R:A\to[0,\infty)$, $R(u,v)$ = the time taken for an evacuee from *u* to *v.*

We consider that the navigation and hazard functions are deterministic and predetermined by some simulation or by some experiments as suggested in (Olenick *et al.,* 2003).

Suppose that the hazard is detected initially on the nodes $\{u_1,u_2,...,u_q\}$ and is analyzed over a time interval $Time=\{0,1,2,...,t_{\max}\}$. From the information given by the hazard graph and function, we can estimate the time each node is caught by fire. For that we use the function $fire:V\to\{0,1,2,...\}$ with $fire(u)=t$, when the node *u* is caught by fire at the time *t*. This function can be calculated by the following recursive equations:

$$fire(u)=0 \text{ for all the nodes } \{u_1,u_2,...,u_q\} \text{ where the fire started} \tag{1}$$

$$fire(u)=\min\{fire(v)+F(v,u):(v,u)\in A \text{ and } fire(v)<fire(u)\}. \tag{2}$$

The second rule gives an estimation for $fire(u)$ as the earliest time when fire catches *u* from a neighbor node *v,* which was already in fire before *u*. Consider $Nodes^{(t)}=\{u\in V: fire(u)=t\}$ as the set of all the nodes, which are in fire at the time *t*. The computation of *fire* is based on a graph traversal over the time sequence $Time = \{0, 1, 2, ..., t_{\max}\}$.

Definition 1. The dynamic hazard function is defined as follows

$$H^{(t)} : V \to R_+, H^{(t)}(u) = fire(u) - t, \tag{3}$$

and it represents the estimated time left for the node *u* to become hazardous at the time *t*. The node *u* is hazard free when $H^{(t)}(u)$ is positive; when the node *u* is hazardous then the value of $H^{(t)}(u)$ is negative. The navigation under the fire hazard changes so that some navigation arcs are no longer usable if the fire is present on them or the navigation is still possible but it takes longer navigation time in order to cope with the hazard. Hence, the navigation graph is now dynamic given by $G^{(t)} = \left((V, A), c^{(t)}\right)$, which has *G*=(*V*, *A*) as the underlying graph and $c^{(t)}$ as dynamic costs for the arcs. The dynamic cost function $c^{(t)} : A \to \overline{R}_+$ can be defined as follows:

$$c^{(t)}(u,v) = \begin{cases} R(u,v) \text{ when } t < t_1 \\ R(u,v) + f(u,v;t-t_1),\ t_1 \le t \le t_1 + time(t_1,t_2) \\ \infty,\ t > t_1 + time(t_1,t_2) \end{cases}, \tag{4}$$

where $t_1 = \min\{fire(u), fire(v)\}$ is the time when (*u,v*) becomes hazardous (Tabirca *et al.*, 2009).

2.1 Dynamic Maximum Safety

We can now introduce the concept of maximum dynamic safety starting from the dynamic graph $G^{(t)} = \left((V, A), c^{(t)}\right)$ in the All-To-All case. This dynamic safety concept measures the amount of time after which there is no longer possible to have safe navigation between any two nodes. For that we use the path safety concept which is similar to the approach presented by Barnes *et al.* (2007) for the static case or by Tabirca *et al.* (2009) for the All-To-One case.

Definition 2. The safety of $path = (u = v_0, v_1, ..., v_p = v)$ starting from *u* at the time *t* towards *v* is defined as

$$S^{(t)}\left(path\right) = H^{(t)}\left(u\right) \text{ if } path = (u) \text{ contains only one node} \tag{5}$$

$$S^{(t)}(path) = \min\left\{S^{(t+c^{(t)}(u,v_1))}(path_1) - c^{(t)}(u,v_1), H^{(t)}(u)\right\}, \text{ where } path_1 = (v_1, ..., v_p = v) \tag{6}$$

We can now extend the safety concept for the All-to-All case as follows.

Definition 3. The safety function $S^{(t)} : V \times V \to \overline{R}$ is defined by

$$S^{(t)}(u,v) = H^{(t)}(v), \text{ when } u = v \tag{7}$$

$$S^{(t)}(u,v) = \max\left\{S^{(t)}(path) : path \text{ is a path between } u \text{ and } v\right\}, \text{when } u \ne v \tag{8}$$

for each pair of nodes *u, v*. Based on some theoretical considerations, it can be proven that the dynamic safety values satisfy the following theorem.

Theorem 4. The safety values $S^{(t)}(u,v)$ satisfy the following recursive equation

$$S^{(t)}(u,v) = \min\left\{H^{(t)}(u), \max\left\{S^{(t+c^{(t)}(u,w))}(w,v) - c^{(t)}(u,w) : (u,w) \in A\right\}\right\} \tag{9}$$

where $u \ne v$.

The theorem itself does not provide useful practical insights, however it gives a recursive equation to calculate the safety values. The safety value $S^{(t)}(u,v)$ can be calculated from the safety values $S^{(t+c^{(t)}(u,w))}(w,v)$ from which the costs $c^{(t)}(u,w)$ are subtracted for any $(u,w) \in A$. The maximum safety path from *u* to *v* at the time *t* is defined as the path with the maximum safety value $S^{(t)}(u,v)$. In order to generate the safety paths associated with the safety values $S^{(t)}(u,v)$, $Successor^{(t)}(u,v)=w$ is defined as the node w succeeding u on the safety path between u and v. By convention $Successor^{(t)}(u,v)=-1$ when $u=v$. According to Theorem 4, the equations of the dynamic safety matrices are given by

$$S^{(t)}(u,v)=H^{(t)}(u) \text{ when } u=v \tag{10}$$

$$S^{(t)}(u,v)=\min\left\{H^{(t)}(u),\max\left\{S^{(t+c^{(t)}(u,w))}(w,v)-c^{(t)}(u,w):(u,w)\in A\right\}\right\} \text{ when } u\neq v\,. \tag{11}$$

```
function DynamicSafety(tmax, n, C, S, Succesor){
    for t=0 to Tmax-1 do
      for each u in V do
           S^(t)(u,u) = 0 ; Successor^(t)(u,u) = -1;
           for each v in V \{u} do
                S^(t)(u,v) = -∞ ; Successor^(t)(u,v) = u;
    // find the shortest path at time T using a static solution
    StaticSafety( S^(Tmax), Succesor^(Tmax) );
    // iterate backward
    for t = Tmax-1 downto 0 do
       for each node u in V do and for each node v in V do
          for each node w adjacent to u do
                if S^(t)(u,v) < S^(t+c^(t)(u,w))(w,v) - c^(t)(u,w) then
                          S^(t)(u,v) = S^(t+c^(t)(u,w))(w,v) - c^(t)(u,w) ;
                          Successor^(t)(u,v) = w;
                S^(t)(u,v) = min{ S^(t)(u,v), H^(t)(u) };
}
```

Figure 1: Algorithm to calculate the matrices $S^{(t)}, Successor^{(t)}$.

These equations give a retrospective way to calculate both matrices $S^{(t)}$ and $Succesor^{(t)}$. Firstly, the matrix $S^{(t_{\max})}$ can be calculated with the static equations

$$S^{(t_{\max})}(u,v)=H^{(t_{\max})}(u) \text{ when } u=v$$

$$S^{(t_{\max})}(u,v)=\min\left\{H^{(t_{\max})}(u),\max\left\{S^{(t_{\max})}(w,v)-c^{(t_{\max})}(u,w):(u,w)\in A\right\}\right\} \text{ when } u\neq v$$

based on a simple Dijkstra-like computation. This solution should also generate the matrix *Successor,* which gives $Succesor^{(t_{\max})}(u,v)=w$ when w is the successor of u of the maximal safety path. Assume this computation is achieved by the function $StaticSafety\left(S^{(t_{\max})}, Succesor^{(t_{\max})}\right)$. Then, we can retrospectively calculate $S^{(t)}$ and $Succesor^{(t)}$ from the sequence of matrices $S^{(t_{\max})}$,…, $S^{(t+1)}$ and $Succesor^{(t_{\max})}$, …, $Succesor^{(t)}$ based on Equations (10, 11). For that we need to consider all the neighbors w of u; when the maximum $\max\left\{S^{(t+c^{(t)}(u,w))}(w,v)-c^{(t)}(u,w):(u,w)\in A\right\}$ is given by the node w then the value of $Successor^{(t)}(u,v)$ becomes w. The details of the dynamic safety computation are presented in Figure 1. It can be seen that the overall complexity of the dynamic safety computation is given by $O\left(|V|^2\cdot\log|V|+|V|\cdot|A|+t_{\max}\cdot\left(|V|^2+|V|\cdot|A|\right)\right)$.

Theorem 5. The safety value $S^{(t)}(u,v)$ represents the maximum amount of time one can delay safely at the node *u* heading to the node *v* at the time *t*.

This theorem gives that $S^{(t)}(u,v)$ is the maximum time to delay at the node *u* and still have a safe navigation to the node *v*. Suppose that a fire-fighter is at the location *u* to assess the hazard or to help the injured people and then to move to the location *v*. When the value $S^{(t)}(u,v)$ is positive then the fire fighter knows that there will be a hazard-free path to *v*. The value $S^{(t)}(u,v)$ can be $-\infty$ and in this case the fire-fighter has no safety path to move along *v*. Even when $S^{(t)}(u,v)$ is negative but not $-\infty$, the fire-fighter can still move to *v* however he will encounter some hazardous arcs along the safety path.

3 Final Conclusions

This work proposed a model for the dynamic safety notion, which is related with building hazard emergencies. Several results were proposed in order to calculate the dynamic safety and to prove that the safety is the maximum amount of time to delay safely in nodes. An efficient algorithm was also proposed to generate the dynamic safety matrix as well as the successor matrix.

References

Ahuja, R.K., Orlin, J.B., Pallottino, S. and Scutella, M.G. (2002). *Dynamic Shortest Paths Minimizing Travel Times and Costs.* MIT Sloan Working Paper No. 4390-02.

Barnes, M., Leather, H. and Arvind., D.K. (2007). *Emergency Evacuation using Wireless Sensor Networks*. Proceedings of the 32nd IEEE Conference on Local Computer Networks, pp 851-857.

Chabini, I. (1997). *A new shortest path algorithm for discrete dynamic networks*. Proceedings of the 8th IFAC Symposium on Transport Systems, Chania, Greece, pp.551-556.

Chabini, I. (1998). *Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run tim*e. Transportation Research Record 1645, pp.170-175.

Cooke, K.L. and Halsey, E. (1966). *The Shortest Route Through a Network with Time-Dependent Internodal Transit Times*. J. Math. Anal. Appl., 14, pp.493-498.

Demetrescu, C and Italiano, G. F. (2006). *Dynamic shortest paths and transitive closure: Algorithmic techniques and data structures*, Journal of Discrete Algorithms, 4(3) pp. 353-383.

Desmet, A. and Gelenbe, E. (2013). *Graph and Analytical Models for Emergency Evacuation*, Future Internet, 5(1), pp.46-55.

Kuligowski, E.D. and Peacock R.D. (2005). *A Review of Building Evacuation Models*, NIST Technical Note, 2005.

Kuligowski, E.D. (2013). *Predicting Human Behaviour During Fires*, Fire Technology, 49(1), pp. 101-120

Li, Q., DeRosa, M. and Rus, D. (2003) *Distributed algorithm for guiding navigation across a sensor network*. Proceedings of ACM Int'l Symp. on Mobile Ad Hoc Networking and Computing (MobiHOC 2003), ACM Press, pp. 313–325.

Olenick, S. M. and Carpenter, D.J.. (2003). *An updated international survey of computer models for fire and smoke*, SFPE Journal of Fire Protection Engineering, 13(2), pp. 87–110.

Ramalingam, G. and Reps, T.W. (1996). *An incremental algorithm for a generalization of the shortest-path problem*, J. Algorithms, 21(2), pp. 267-305.

Tabirca, T., Brown, K. N. and Sreenan, C.J. (2009). *A Dynamic Model for Fire Emergency Evacuation Based on Wireless Sensor Networks*, Proceedings of the International Symposium on Parallel and Distributed Computing, Lisbon, pp. 29-36.