# Routing Diverse Evacuees with Cognitive Packets

Huibo Bi and Erol Gelenbe
Imperial College London
Department of Electrical and Electronic Engineering
Intelligent Systems and Networks Group
Email: {huibo.bi12, e.gelenbe}@imperial.ac.uk

*Abstract*—**This paper explores the idea of smart building evacuation when evacuees can belong to different categories with respect to their ability to move and their health conditions. This leads to new algorithms that use the Cognitive Packet Network concept to tailor different quality of service needs to different evacuees. These ideas are implemented in a simulated environment and evaluated with regard to their effectiveness.**

## I. Introduction

Emergency navigation [1], [2], [3] is the process of guiding evacuees to exits when an emergency occurs, and this paper presents an multipath routing algorithm that attempts to satisfy the requirements of diverse classes of evacuees. The approach is based on the Cognitive Packet Network (CPN) which searches [4], [5], [6] for optimised routes rapidly and adaptively with regard to user-defined Quality of Service (QoS) goals to evaluate the quality of paths and choose appropriate routes for different groups of evacuees. In order to minimise the congestion encountered by each evacuee, a congestion-aware algorithm that can predict the future clogging with respect to the current observed congestion is proposed. "Direction oscillation problem" [7] that occurs in adaptive routing with sensitive metrics [8] due to delays in the available information is discussed and "movement depth" is introduced to facilitate the evacuees to update their routes periodically to avoid being directed into hazards.

The remainder of the paper is organized as follows: Section II presents the related studies of emergency navigation algorithms. In Section III we recall the concept of CPN. Then we introduce the QoS metrics designed for diverse categories of evacuees in Section IV, and discuss the issue of oscillations that arise in adaptive networks and their alleviation in Section V. The simulation model and assumptions are introduced in Section VI. Finally, the results of experiments are discussed in Section VII.

## II. Related Work

Much research in emergency navigation focuses on "normal" individuals with identical physical attributes such as mobility and health level, and conventional models such as flow-based models [9], [10], [11], [12], [13], [14] hypothesise evacuees as continuous homogeneous flows. Potential-maintenance approaches [15], [16], [17], [18] concentrate on the navigation algorithms that do not take the physical attributes of evacuees into account. The "magnetic model" in [19] sets a walking velocity for each evacuee but ignores personal requirements as well as the effect of social interaction. The advent of multi-agent models makes it more convenient to customise physical attributes for individuals. However, previous work has focused on incorporating sociological factors due to the easiness of describing social behaviours such as coordination or stampede. Although in [20] physical, psychological and moving attributes are considered for each evacuee, health factors such as initial health values and resistance to hazard which are influenced by gender, age, etc. are not considered. Queueing models can also be used to evaluate congestion in such systems [21][22].

Navigation algorithms proposed previously search for the shortest or the safest path and in [15] a self-organizing sensor network is proposed to guide a user (robots, people, unmanned aerial vehicle, etc.) through the safest path by using "artificial potential fields" [4]. An attractive force pulls users to the destination while repulsive forces from dangerous zones push them away. In [16] a temporally ordered routing algorithm [17] routes evacuees to exits through safer paths. A navigation map is manually defined to avoid impractical paths and each sensor is assigned with an altitude with respect to its hops to the nearest exit. Combining the definition of effective length with Dijkstra's algorithm, [23] presents a decentralized evacuation system with decision nodes (DNs) and sensor nodes (SNs) to compute the shortest routes in real time, while opportunistic communications [24] based emergency evacuation systems [25] have the advantage of being more robust to network attacks [26] that often accompany emergencies caused by malicious acts.

## III. The Cognitive Packet Network

The Cognitive Packet Network (CPN) introduced in [27], [28], [29] was originally proposed for large scale and fast-changing networks. As the conditions of networks typically change frequently, the protocols which periodically update the network status as well as the routing schemes that require the convergence of an entire network can always result in a constant lag and impede the performance. Therefore, the concept of CPN is proposed to solve this problem by sending Cognitive Packets to monitor the network conditions and discover new routes sequentially. In the CPN, "Cognitive packets" play a dominating role in routing and flow control other than routers and protocols. Cognitive packets can adapt to pursue their predefined goals and learn from their own investigations and experience from other packets. Each node in CPN hosts recurrent random neural networks (RNN) with a neuron per neighbour node, and maintains a routing list which reserves a fixed number of QoS-oriented routes in a descending order. CPN carries three types of packets: smart packets (SPs), acknowledgements (ACKs) and dumb packets (DPs). SPs are used for information collection and routing discovery with respect to a specific QoS goal. They can either choose the most excited neuron as next hop or drift randomly to explore and measure undiscovered or not recently measured paths. When a SP reaches the destination, an ACK which stores all the gathered information will be generated and sent back to the source node through the reverse path. Because loops may be produced during route exploration, a loop remove algorithm is acted on the reverse path to take out any sequences of nodes with the same origin and destination. When an ACK arrives at a node, it will update the routing list and train a random neural network [30] by performing Reinforcement Learning (RL) with neural networks [31], [32]. DPs are the packets which actually carry the payloads. It always selects the best path at the top of the routing list as the next hop.

In the context of emergency evacuation, because the environment is highly dynamic, SPs are used to search available routes and collect information such as the hazard intensity of a node. ACKs backtrack with collected information and update the excitation level of neurons at each traversed node. Evacuees are considered as DPs and always follow the top ranked path in the routing list. To apply the concept of CPN to emergency evacuation, the following requirements must be satisfied: a predefined graph-based layout of a confined place with information such as length of edges and node capacity is available; The edges of the graph depict the paths and the vertices represent the physical areas where sensors are installed. The sensors could communicate with its neighbour nodes and detect the typical hazards such as fire, smoke, water, etc. Additionally, in order to perform the congestion-aware algorithm, we also assume that the sensors can sense the arrival of a civilian and record the number of evacuees congested in proximity.

## IV. QoS Metrics

QoS metrics are expressed as goal functions which provide input data for RNN and are minimized by CPN. Details of the RNN based algorithm are available in [33]. It is not included in this paper due to space constraints. The simplest metric used in the CPN based algorithm in [34] is distance based, and it pursues the shortest effective path, rather than the path with the smallest number of segments or hops to traverse, from a source to a destination. Other novel metrics considered in this paper are the time-oriented goal function, the energy efficiency oriented goal function and the safety-oriented goal function. However our simulations will be limited to the time oriented goal. Previous work in [35] has shown that congestion will affect the performance of evacuation routing algorithms significantly. Meanwhile, certain strategic nodes such as stairs can become bottlenecks and cause long lasting congestion during an evacuation process. In order to ease congestion and bypass bottleneck nodes appropriately, we include congestion in the QoS metric.

Unlike previous congestion-aware algorithms, the algorithm that we proposed evaluates if the current congestion located at a certain node will affect an evacuee by comparing the duration of the congestion and the arrival instant of the evacuee. The congestion situation at a node when an evacuee reaches depends on the current situation of the node and the total number of arrivals and departures before the evacuee arrives. To estimate the total number of arrivals and departures, we use the average arrival rate and departure rate recorded by each node. The details are given in **Algorithm 1** and we assume that if the queue length of a node is larger than one, then congestion will occur.

**Algorithm 1** Predict the potential amount of congestion encountered by an evacuee during the evacuation process

**Data:** A path $\pi$ explored by SPs from a node $\pi(0)$
**Result:** The potential number of congestion encountered by an evacuee traversing across the path $\pi$

```
Set the total number of congestion $C_{total}$ to 0
Set the total travel time $T_{total}$ to 0
for all the edges $e(\pi(i), \pi(i+1)) \in$ path $\pi$ do
    /* Calculate the time cost on the edge $e(\pi(i), \pi(i+1))$ */
    $t_{edge} \leftarrow \frac{E^e(\pi(i), \pi(i+1))}{V_{speed}}$
    where $E^e(\pi(i), \pi(i+1))$ is the effective length of $e(\pi(i), \pi(i+1))$, $\pi(i)$ is the source node of the edge, $\pi(i+1)$ is the destination node of the edge, $V_{speed}$ is the average speed of certain type of evacuees.
    /* Compute the time cost at the node $\pi(i)$ */
    if $q_0^i + \bar{\lambda} \cdot T_{\pi(i)} - \bar{\mu} \cdot T_{\pi(i)} > 0$ then
        /* If the congestion will remain there when the evacuee reaches the node $\pi(i)$, then plus the time delay caused by congestion */
        $t_{node} \leftarrow \frac{q_0^i + \bar{\lambda} \cdot T_{\pi(i)} - \bar{\mu} \cdot T_{\pi(i)}}{\bar{\lambda}}$
        where $q_0^i$ is the current queue length of the node $\pi(i)$, $\bar{\lambda}$ is the average arrival rate of the node $\pi(i)$, $\bar{\mu}$ is the departure rate of the node $\pi(i)$, $T_{\pi(i)}$ is the potential time cost to reach the node $\pi(i)$, which is the current value of $T_{total}$. Term $\frac{q_0 + \bar{\lambda} \cdot T_{\pi(i)} - \bar{\mu} \cdot T_{\pi(i)}}{\bar{\lambda}}$ is the average time calculated by Little's formula.
        /* If the predicted queue length when the evacuee reaches it is larger than zero, the arrival of the evacuee will trigger congestion */
        $C_{total} \leftarrow C_{total} + 1$
    end if
    $T_{total} \leftarrow T_{total} + t_{edge} + t_{node}$
end for
Return $C_{total}$
```

The time-oriented goal function pursues the shortest time cost to reach an exit, and satisfies the QoS need of normal evacuees. The path traversal time is predicted as follows:

$$\begin{aligned} G_t = \sum_{i=1}^{n-1}\{ & \frac{E^e(\pi(i),\pi(i+1))}{V_{speed}} \\ & + K[\frac{q_0^i + \bar{\lambda}\cdot T_{\pi(i)} - \bar{\mu}\cdot T_{\pi(i)}}{\bar{\lambda}}]\} \end{aligned} \tag{1}$$

where $\pi$ represents a particular path, $n$ is the number of nodes on the path $\pi$, and $\pi(i)$ is the $i$-th node on the path $\pi$. $E^e(\pi(i), \pi(i+1))$ is the effective length of the edge between node $\pi(i)$ and node $\pi(i+1)$. $V_{speed}$ is the average speed of certain category of civilians. $K[X]$ is a function that takes the value 0 if X is smaller than zero or X if X is larger than or equals to zero respectively. Term $q_0^i$ is the current queue length of the node $\pi(i)$, $\bar{\lambda}$ is the average arrival rate of the node $\pi(i)$, $\bar{\mu}$ is the departure rate of the node $\pi(i)$, $T_{\pi(i)}$ is the potential time cost to reach the node $\pi(i)$.

Energy efficiency oriented goals will search for the path with minimum energy consumption. This function can be used for wheelchairs or robots due to the battery power limitations.

$$\begin{aligned} & G_e = c_b \cdot C_{total} + \sum_{i=1}^{n-1} c_s \cdot E^e(\pi(i),\pi(i+1)) + \\ & \sum_{i=2}^{n-1} c_t \cdot \theta(\pi(i-1),\pi(i),\pi(i+1)) \end{aligned} \tag{2}$$

If $C_{total}$ is the total congestion calculated by **Algorithm 1**, then $c_b$ is the energy consumption of a wheelchair when a braking event happens, and $c_s$ is the energy consumption per centimetre when moving straight on an edge. $E^e(\pi(i), \pi(i+1))$ is the effective length of the edge between node $\pi(i)$ and node $\pi(i+1)$. Term $c_t$ is the energy consumption per degree when a turning event happens, $\theta(\pi(i-1), \pi(i), \pi(i+1))$ is the rotation angle between edge $E(\pi(i-1), \pi(i))$ and $E(\pi(i), \pi(i+1))$.

The safety-oriented goal function resolves the safest path. It can be used for sick people or children as they are more likely to faint due to the impact of poisonous smoke in a fire. To estimate the hazard intensity of a node, we assume the hazard spread rate is $a$ (cm/s) and the hazard growth rate at a node is $b$.

$$\begin{aligned} G_s = & \sum_{i=1}^{n-1} 1[t_{evacuee}^{\pi(i+1)} + t_{current} < t_{hr}^{\pi(i+1)}] \cdot b \cdot \\ & (t_{evacuee}^{\pi(i+1)} + t_{current} - t_{hr}^{\pi(i+1)}) + E^s(\pi(i),\pi(i+1)) \end{aligned} \tag{3}$$

where $1[X]$ is a function that takes the value 1 or 0 if X is false or true respectively. The term $t_{evacuee}^{\pi(i+1)}$ is the time for an evacuee to reach node $\pi(i+1)$ and $t_{current}$ is the current simulation time. The term $t_{hr}^{\pi(i+1)}$ is the time for the hazard to reach node $\pi(i+1)$, it can be estimated with the hazard spread rate $a$. $E^s(\pi(i), \pi(i+1))$ is then the effective safety value between node $\pi(i)$ and node $\pi(i+1)$. We employ $E^s(\pi(i), \pi(i+1))$ to ensure the value of $G_s$ is not nil if the hazard does not reach the path during a civilian's evacuation process.

The details of this approach is shown in **Algorithm 2**.

**Algorithm 2** Predict the potential hazard of a path

```
Data: A path $\pi$ explored by SPs from a node
Result: The potential hazard of the path $\pi$

for all the nodes $n$ in the environment do
    Find the shortest path $D$ from $n$ to the fire source by using the built-in map
    $t_{hr} \leftarrow D/a$
    where $t_{hr}$ is the time cost for the hazard to reach $n$
end for
Set the total potential hazard $H_{total}$ of path $\pi$ to 0
for all the nodes $\pi(i) \in$ path $P$ do
    /* Calculate the potential time cost $t_{evacuee}^{\pi(i)}$ for an evacuee to reach $\pi(i)$ from the source node of $\pi$ */
    $t_{evacuee}^{\pi(i)}$ can be calculated by Algorithm 1
    /* Calculate the potential hazard $h_{node}$ when an evacuee arrives $\pi(i)$ from the source node of $\pi$ */
    $h_{node} \leftarrow b \cdot (t_{evacuee}^{\pi(i)} + t_{current} - t_{hr}) + E^s(\pi(i), \pi(i+1))$
    where $E^s(\pi(i), \pi(i+1))$ is the effective safety value of the edge between node $\pi(i)$ and $\pi(i+1)$
    $H_{total} \leftarrow H_{total} + h_{node}$
end for
Return $H_{total}$
```

## V. Alleviation of Oscillations

Path oscillations will occur when a sensitive metric [8] is used or the process of updating sensor readings and arrival of ACKs are asynchronous when the hazard spreads quickly. This phenomenon does not affect evacuees with portable devices as they will receive a whole path but confuses civilians using visual indicators because the guiding direction oscillates. In [29] several ways to mitigate oscillations in packet networks are discussed such as using a path switch probability, or a minimum QoS gain threshold to switch routes, or ensuring the usage of a path by a minimum number of packets before a path switch can occur though this last choice would not make sense if "packets" are human beings and the path switch is being caused by a deterioration of the safety of certain paths. In our system we set a path switch probability to ease the direction oscillation on the visual indicators.

Moveover, because CPN is a source routed algorithm, each evacuee with a portable device will obtain a route to exits when they begin to evacuate. However, some evacuees cannot stick to the original route as the path condition varies (hazard, congestion, etc.). A simple method for the evacuees to use the up-to-date suggestion is to adopt the top ranked route at each hop. However, this may cause oscillations when successive decisions vary frequently. Hence, we use "movement depth" to ensure that evacuees only accept a new path choice suggestion after traversing a certain number of nodes. To prevent that movement depth hinders the flexibility of the evacuation routing and directs civilians to hazards, the sequential decisions will be checked periodically to verify if the hazard has reached a particular node. If the hazard has reached any hop or edge related to these decisions, they will be discarded immediately and an alternate path will be used. However this approach can also be very dangerous since evacuees may get close to a hazard and then have no remaining time or possibility to change to a safe path. Thus such approaches need to be evaluated in the context of varying degrees of risk that include random rates of movement of the hazard (fire, smoke, gases, etc.) itself.

Ten randomized iterations have been conducted for movement depth from 1 to 10 in a scenario with normal fire spreading rate and 120 evacuees. We use this scenario because the effect of oscillation is more obvious in high density environments due to the larger variation of congestion. Fig. 1 shows that the performance of the system reaches the peak mean value when the movement depth is 3 and then decreases gradually with the increase of the movement depth. This is because when movement depth is small (1 or 2), though it uses the up-to-date suggestion, some evacuees may be temporarily trapped by the puzzled suggestion. Meanwhile, CPN evaluates the QoS metric of a whole path other than a single edge. In other words, the evacuees should finish the whole path to satisfy their QoS needs if the condition does not change. If they continuously change paths, they will not follow the optimal path but a collection of the first edge of optimal paths. If movement depth is too large, evacuees will use the outdated decisions and therefore increase the possibility of guiding evacuees near hazard and backtracking. Hence, we set the movement depth to 3 in our experiments.

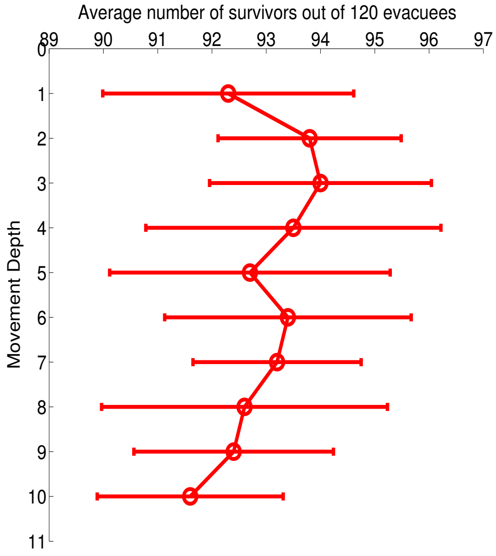

Fig. 1. Average number of survivors out of 120 evacuees with different movement depths. The error bars show the standard deviation of 10 experiments.

## VI. Simulation and Experiments

The Distributed Building Evacuation Simulator (DBES) [1] which is an agent based discrete event simulator where each entity is modelled as a process, and the physical world is represented by a set of "Points of Interest" (PoI) which are significant locations such as doorways and corridors, and links which are physical paths.

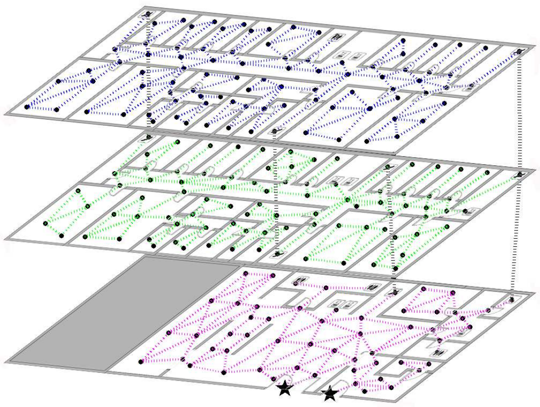

Fig. 2. The graph based layout of the building. The black stars represent the exits on the ground floor.

The environment is the lower floors of Imperial College's EEE building shown in Fig. 2. SNs and DNs are deployed in proximity to PoI. SNs collect hazard information and DNs execute the decision support algorithms. Real time decisions can be transformed to evacuees with visual indicators that are installed at POI or portable devices carried by the civilians. To evaluate the performance and adaptability of decision algorithms, we assume a fire outbreak near the eastern staircase on the ground floor that blocks a main channel. Initially, evacuees are randomly scattered in the building and that they hear or see fire alarms and evacuate immediately. Three routing algorithms are employed: Dijkstra's shortest path algorithm, CPN based algorithm with distance metric and CPN with time metric. In first scenario, when the fire spreads, global hazard information will be synchronized in the network and Dijkstra's shortest path algorithm will be executed at each decision node to obtain the up-to-date shortest path. The second and the third scenario are similar to the first one but using the CPN based algorithm to collect information and make decisions.

## VII. Results and Discussion

The experiments are conducted on scenarios with 30, 60, 90, 120 evacuees, respectively. The Dijkstra's shortest path algorithm and original distance-oriented goal function based experiments are performed for comparison purpose. To exclude the interferences of other factors, the evacuees in the simulation are homogeneous with identical velocity and initial health value.

The results of ten randomised experiments are presented in Fig 3. The error bars show the highest and lowest number of survivors in the ten simulations. In low occupancy rate (30 evacuees), both CPN with time metric and CPN with distance metric reach the performance of Dijkstra's algorithm which can be considered as an optimal algorithm because it has full knowledge of the environment and the level of congestion is low. As the population density increases, congestion occurs more and more frequently and CPN with time metric becomes the best algorithm because it takes the potential congestion into account and searches the path with shortest evacuation time. While Dijkstra's algorithm and CPN with distance metric insist on one main channel to the exits, CPN with time metric can distribute evacuees to several channels and effectively ease congestion, as can be seen from Table I.

Fig. 4 shows the average evacuation time in the above scenarios and the error bars represent the maximum and minimum egress time of ten experiments. Results confirm that CPN with time metric achieve the average shortest evacuation time among the three metrics. Furthermore, both CPN based algorithms can accomplish an evacuation process faster than Dijkstra's algorithm in low occupancy rates (30 and 60 evacuees) while gaining similar number of survivors. However, the three metrics achieve comparable results in high population densities (90 and 120 evacuees), with a slightly advantage to CPN with time metric. This is mainly caused by congestion formed at the bottlenecks of the network. A large amount of evacuees have to queue for a long time at the central staircase between the ground floor and the first floor before they can reach the ground floor. The maximum queue length at this staircase can attain 25 for Dijkstra's algorithm and 20 for CPN with distance metric. Interestingly, although the value for CPN with time metric is 16, alternate congestion will form near exits as the egresses become new bottlenecks. It proves that in high density environment, the bottlenecks in the system can significantly affect the average evacuation time. In other words, though the CPN with time metric can accelerate the process for civilians to arrive in the vicinity of exits, it can not significantly reduce the average evacuation time under the equivalent departure rate of the exits.

Fig. 5 - 7 show the number of visits of each edge in a scenario with 120 evacuees. The thickness of solid line is proportional to the number of visits. The edges that have not been traversed are shown as dotted lines. Fig. 5 and Fig. 6 which use the shortest path routing metric shows that the shortest egress path on the ground floor are overused by evacuees. Most evacuees select the identical main channel to egress and cause continuous congestion. In both Dijkstra's algorithm and CPN with distance metric, 96% of the evacuees use the main exit in the middle and only 4% of the evacuees escape from the alternate exit. Fig. 7 which employs the shortest time routing metric shows that several channels are formed on the ground floor to transmit evacuees. This phenomenon does not appear on the other floors because the corridors on the first and second storey are narrow. Meanwhile, the distribution of evacuees are more balanced between the two exits, with 79% of the evacuees depart from the main exits and 21% of the civilians choose the other exit. Therefore, the probability of congestion is reduced and the latency in the evacuation process is decreased.

| Number of Evacuees | Dijkstra | CPN with SP | CPN with ST |
|---|---|---|---|
| 30 | 97.8 | 97.6 | 80.8 |
| 60 | 382.8 | 363.5 | 332.7 |
| 90 | 695.9 | 676.8 | 627.7 |
| 120 | 943.5 | 935.7 | 850.3 |

TABLE I

AVERAGE NUMBER OF CONGESTION APPEARS IN EACH SCENARIO FOR DIFFERENT PATH FINDING ALGORITHMS. CONGESTION IS ASSUMED TO HAPPEN WHEN AN EVACUEE REACHES A NODE WITH A NON-ZERO QUEUE LENGTH. "CPN WITH SP" PRESENTS CPN BASED ALGORITHM WITH DISTANCE METRIC AND "CPN WITH ST" DEPICTS CPN BASED ALGORITHM WITH TIME METRIC.

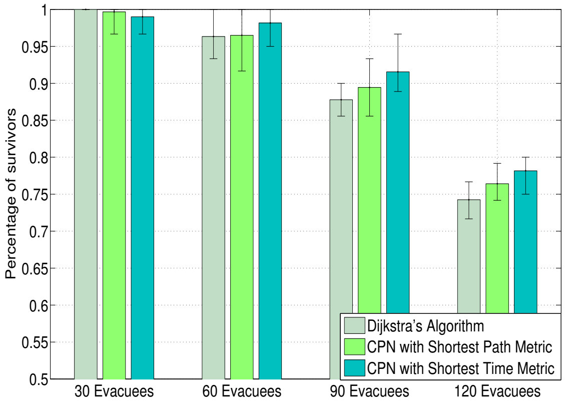

Fig. 3. Percentage of survivors for each scenario. The results are the average of 10 randomized simulation runs, and error bars shows the min/max result in any of the 10 simulation runs.

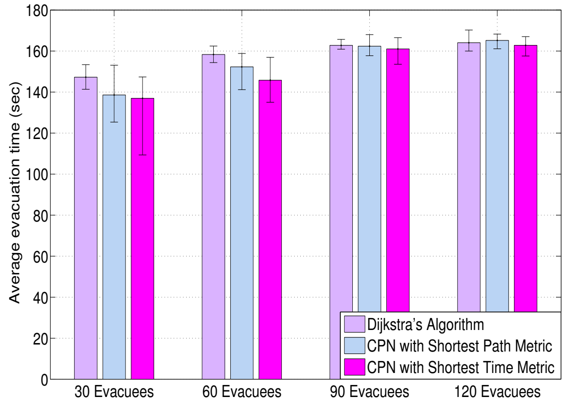

Fig. 4. Average evacuation times of ten iterations in arbitrary time units. The error bars represent the min/max values found in the ten simulations.

## REFERENCES

[1] N. Dimakis, A. Filippoupolitis, and E. Gelenbe, "Distributed building evacuation simulator for smart emergency management," *The Computer Journal*, vol. 53, no. 9, pp. 1384–1400, 2010.

[2] E. Gelenbe and F.-J. Wu, "Large scale simulation for human evacuation and rescue," *Computers and Mathematics with Applications*, vol. 64, no. 12, pp. 3869–3880, December 2012.

[3] ——, "Future research on cyber-physical emergency management systems," *Future Internet*, vol. 5, no. 3, pp. 336–354, 2013.

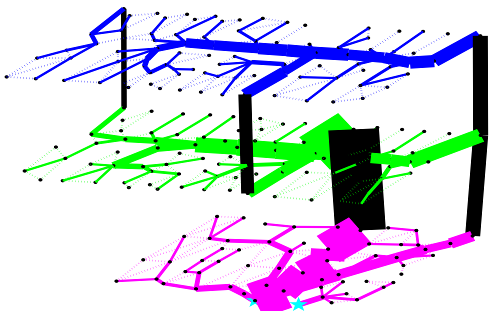

Fig. 5. Dijkstra's algorithm routing

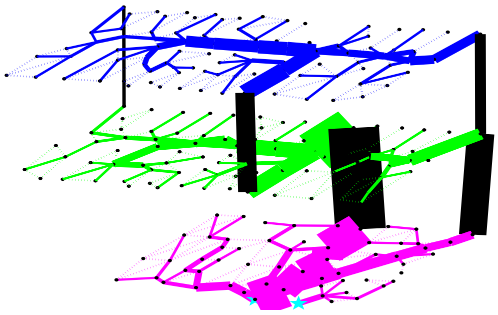

Fig. 6. Shortest-path metric routing

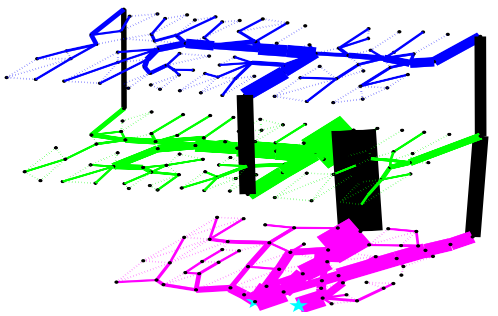

Fig. 7. Shortest-time metric routing

[4] E. Gelenbe, N. Schmajuk, J. Reif, and J. Staddon, "Autonomous search by robots and animals: A survey," vol. 22, no. 1, pp. 23 – 34, 1997.

[5] E. Gelenbe and Y. Cao, "Autonomous search for mines," *Eur. J. Oper. Res.*, vol. 108, no. 2, pp. 319–333, Jul. 1998.

[6] E. Gelenbe, "Search in unknown random environments," *Physical Review E*, vol. 82, no. 6, 2010.

[7] W.-T. Chen, P.-Y. Chen, C.-H. Wu, and C.-F. Huang, "A load-balanced guiding navigation protocol in wireless sensor networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE, 2008, pp. 1–6.

[8] E. Gelenbe, "Sensible decisions based on qos," *Computational management science*, vol. 1, no. 1, pp. 1–14, 2003.

[9] L. Chalmet, R. Francis, and P. Saunders, "Network models for building evacuation," *Fire Technology*, vol. 18, no. 1, pp. 90–113, 1982.

[10] R. Francis, *A negative exponential solution to an evacuation problem*. National Bureau of Standards, Center for Fire Research, 1984.

[11] T. M. Kisko and R. L. Francis, "Evacnet+: a computer program to determine optimal building evacuation plans," *Fire Safety Journal*, vol. 9, no. 2, pp. 211–220, 1985.

[12] B. Hoppe and É. Tardos, "Polynomial time algorithms for some evacuation problems," in *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1994, pp. 433–441.

[13] ——, "The quickest transshipment problem," *Mathematics of Operations Research*, vol. 25, no. 1, pp. 36–62, 2000.

[14] Q. Lu, Y. Huang, and S. Shekhar, "Evacuation planning: a capacity constrained routing approach," in *Intelligence and Security Informatics*. Springer, 2003, pp. 111–125.

[15] Q. Li, M. De Rosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," in *Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM, 2003, pp. 313–325.

[16] Y.-C. Tseng, M.-S. Pan, and Y.-Y. Tsai, "Wireless sensor networks for emergency navigation," *Computer*, vol. 39, no. 7, pp. 55–62, 2006.

[17] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. IEEE, 1997, pp. 1405–1413.

[18] M.-S. Pan, C.-H. Tsai, and Y.-C. Tseng, "Emergency guiding and monitoring applications in indoor 3d environments by wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 1, pp. 2–10, 2006.

[19] S. Okazaki and S. Matsushita, "A study of simulation model for pedestrian movement with evacuation and queuing," in *International Conference on Engineering for Crowd Safety*, 1993, pp. 271–280.

[20] Y. Bo, W. Yong-gang, and W. Cheng, "A multi-agent and gis based simulation for emergency evacuation in park and public square," in *Computational Intelligence and Security, 2009. CIS'09. International Conference on*, vol. 1. IEEE, 2009, pp. 211–216.

[21] E. Gelenbe and R. R. Muntz, "Probabilistic models of computer systems. part I: Exact results," *Acta Informatica*, vol. 7, no. 1, pp. 35–60, 1979.

[22] E. Gelenbe, "Probabilistic models of computer systems. part II: Diffusion approximations, waiting times and batch arrivals," *Acta Informatica*, vol. 12, no. 4, pp. 285–303, 1979.

[23] A. Filippoupolitis and E. Gelenbe, "A distributed decision support system for building evacuation," in *Human System Interactions, 2009. HSI'09. 2nd Conference on*. IEEE, 2009, pp. 323–330.

[24] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, 2006.

[25] E. Gelenbe and G. Görbil, "Opportunistic communications for emergency support systems," *Procedia Computer Science*, vol. 5, pp. 39–47, 2011.

[26] E. Gelenbe and G. Loukas, "A self-aware approach to denial of service defence," *Computer Networks*, vol. 51, no. 5, pp. 1299–1314, 2007.

[27] E. Gelenbe, R. Lent, and Z. Xu, "Towards networks with cognitive packets," in *Performance and QoS of next generation networking*. Springer London, 2001, pp. 3–17.

[28] E. Gelenbe, R. Lent, and A. Nunez, "Self-aware networks and qos," *Proceedings of the IEEE*, vol. 92, no. 9, pp. 1478–1489, 2004.

[29] E. Gelenbe, "Steps toward self-aware networks," *Communications of the ACM*, vol. 52, no. 7, pp. 66–75, 2009.

[30] E. Gelenbe and S. Timotheou, "Random neural networks with synchronized interactions," *Neural Computation*, vol. 20, no. 9, pp. 2308–2324, 2008.

[31] E. Gelenbe, E. Seref, and Z. Xu, "Simulation with learning agents," *Proceedings of the IEEE*, vol. 89, no. 2, pp. 148–157, 2001.

[32] E. Gelenbe, "Natural computation," *Computer J.*, vol. 55, no. 7, pp. 848–851, 2012.

[33] E. Gelenbe, Z. Xu, and E. Seref, "Cognitive packet networks," in *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*. IEEE, 1999, pp. 47–54.

[34] H. Bi, A. Desmet, and E. Gelenbe, "Routing emergency evacuees with cognitive packet networks," in *Proceedings of the 28th International Symposium on Computer and Information Sciences (ISCIS'13)*. Springer, London, October 2013.

[35] A. Desmet and E. Gelenbe, "Graph and analytical models for emergency evacuation," *Future Internet*, vol. 5, no. 1, pp. 46–55, 2013.