

Article

Routing Diverse Evacuees with the Cognitive Packet Network Algorithm

Huibo Bi

Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2BT, UK; E-Mail: huibo.bi12@imperial.ac.uk; Tel.: +44-207-594-7913

Received: 19 February 2014; in revised form: 2 April 2014 / Accepted: 3 April 2014 /

Published: 17 April 2014

Abstract: Regarding mobility, health conditions and personal preferences, evacuees can be categorized into different classes in realistic environments. Previous emergency navigation algorithms that direct evacuees with a single decision rule cannot fulfil civilians' distinct service requirements and increase the likelihood of inducing destructive crowd behaviours, such as clogging, pushing and trampling, due to diverse mobility. This paper explores a distributed emergency navigation algorithm that employs the cognitive packet network concept to tailor different quality of service needs to different categories of evacuees. In addition, a congestion-aware algorithm is presented to predict the future congestion degree of a path with respect to the observed population density, arrival rate and service rate of each route segment. Experiments are implemented in a simulated environment populated with autonomous agents. Results show that our algorithm can increase the number of survivors while providing improved quality of service.

Keywords: emergency navigation; diverse evacuees; quality of service; congestion management

1. Introduction

Nowadays, traditional emergency evacuation conducted by incident commanders is being superseded by cyber-physical-human systems (CPHS) [1], due to the mass of information exchange and decision-making involved, as well as the advent of low-cost microelectronic devices. As the core of emergency response systems, related emergency navigation algorithms [2–4] have motivated considerable research.

However, previous algorithms usually employed a single metric and did not consider personalised service requirements reflected by the evacuees' own capabilities, such as mobility, resistance to hazards and physical strength. This aggravates “nonadaptive crowd behaviours” (such as herding, crushing, knocking, *etc.*), which contribute considerably to the fatalities and injuries in densely-populated disasters [5]. In this paper, we present a novel routing algorithm that uses diverse goal functions to choose appropriate routes for the associated class of evacuees. The approach is based on the cognitive packet network (CPN) framework, which uses cognitive packets to search [6–9] for optimised routes rapidly and adaptively with regard to user-defined quality of service (QoS) goals. Congestion has various negative impacts on an evacuation process. For example, it can temporarily block evacuee flows and cause long-time latency. Evacuees in electric powered wheelchairs may create an extra consumption of energy due to the slowing down and acceleration effects caused by jamming. To minimise the total number of congestion situations encountered by civilians, a congestion-aware algorithm, which combines the real-time congestion level and the average arrival and service rate, is embedded in goal functions to distribute evacuees to unoccupied routes. However, because sensitive metrics [10], such as congestion, are involved in the routing protocol, the “direction oscillation problem” [11] may occur, due to delays in the available information. This may confuse evacuees and reduce their confidence in the emergency response system. Hence, we set a path switch probability to ease direction oscillations. Although evacuees with portable devices retrieve a whole path to a suitable exit each time they communicate with the system, the suggested route should be updated periodically, due to the fast-changing path situation. Therefore, “movement depth” is introduced to facilitate the evacuees to renew their routes after traversing certain hops.

The rest of the paper is organized as follows. Section 2 presents the related studies of emergency navigation algorithms and congestion management. In Section 3, we recall the concept of CPN. Then, we introduce the congestion-ease algorithm and the QoS metrics designed for diverse categories of evacuees in Section 4 and discuss the issue of oscillations that arise in adaptive networks and their alleviation in Section 5. The simulation model and assumptions are introduced in Section 6, and the results of experiments are discussed in Section 7. Finally, Section 8 presents the conclusion and future work.

2. Related Work

2.1. Emergency Navigation

Because evacuation drills are high-priced, time-consuming and cannot be reproduced, most research employs macroscopic or microscopic simulation models to characterise real evacuations. However, the diversity in the physical attributes of occupants is rarely taken into account. For example, cellular automata models, such as [12], discretize given structures into uniform “cells” that can hold one person. This approach can precisely model the influence of an individual's physical dimensions, but is ineffective in depicting the movement speed and direction, due to the discrete spatial structure [13]. Moreover, evacuees are assumed to have identical physical conditions, and the movement rules are determined by cells. Flow-based models [14–19] hypothesise evacuee movement as continuous flows, and civilians are assumed to have identical physical attributes. Potential maintenance approaches [20–22] concentrate

on establishing safe escape paths and do not take individual ability into consideration. Particle-based algorithms, such as the “magnetic model” in [23], make use of the analogy between particle motions and evacuee movements, but ignore personal requirements, as well as the effect of crowd behaviours. The multi-agent models can customise physical attributes for individuals and imitate complex phenomena efficiently. However, a majority of previous work has focused on studying crowd dynamics. Although simulations in [24] are based on a multi-agent framework that employs evacuation agents with physical, psychological and moving attributes, health factors, such as initial health values and resistance to hazard, which are influenced by gender, age, *etc.*, are not considered.

Current emergency navigation algorithms explore paths with a single characteristic, such as safest paths, shortest paths or quickest paths. Reference [20] uses “artificial potential fields” to establish a self-organizing sensor network with artificial forces and directs users to the safest path. Exits and dangerous zones generate attractive forces and repulsive forces, respectively, and evacuees move under the actuation of the resultant force. In [22], a multi-path routing protocol is proposed based on the temporally ordered routing algorithm [25] to guide evacuees to egresses through safer paths. Each actuator is assigned with an altitude in accordance with its hops to the nearest exit, and egress routes are formed along actuators with a higher altitude to actuators with a lower altitude. In addition, a manual navigation map is defined to eliminate the impractical paths caused by the radio’s penetration capacity. Combining the definition of effective length with Dijkstra’s algorithm, [26] presents a decentralized evacuation system with decision nodes (DNs) and sensor nodes (SNs) to compute the shortest routes in real time, while opportunistic communications-based [27] emergency evacuation systems [28] have the advantage of being more robust to network attacks [29] that often accompany emergencies caused by malicious acts. By considering the evacuation planning problem as a minimum cost network flow problem [30,31], EVACNET+[14–16] converts the original evacuation network to a time-expended network to obtain the quickest routes. The optimal solution is produced by using linear programming methods. To reduce the computational cost of this kind of approach, [32] replaces the linear programming algorithm with a heuristic A*algorithm.

2.2. Congestion Management

Current congestion-aware algorithms in emergency evacuation can be divided into congestion-free algorithms and congestion-ease algorithms. Congestion-free algorithms, such as [19], take node and edge capacity into account and assign a time and routes schedule to evacuees with regard to capacity constraints. This approach can efficiently avoid congestion, but it is not very realistic, because some civilians may have to wait for a certain time before evacuating. Congestion-ease algorithms employ mathematical models to estimate the time delay on a path and to redirect evacuees accordingly. For example, [33] proposes a distributed navigation protocol based on the artificial potential field model to guide evacuees to exits with minimal congestion. Each sensor holds a potential, which is determined by flow velocity, distances to all exits and the weights of exit capacity. Exits are assigned with the maximal value, and paths are constructed from lower potential values to higher potential values. However, the spread of hazards is not considered, and therefore, evacuees can actually be directed to a hazardous area. An egress dynamic feedback control model is proposed in [34] to balance the crowd among exits.

Actuators at each egress build a graph, which is classified as safe, alert and danger regions. If an egress evolves in a danger region, a control algorithm will be performed to reduce the number of evacuees that choose the related exit to regress the state to the safe/alert region. However, some parameters to delimitate the graph are difficult to know in reality, such as the maximum available time for a safe evacuation. Furthermore, this approach only considers the congestion around the exits, but does not take the jamming on the path into account. Reference [35] establishes a macroscopic probabilistic model to mimic the chaotic collective motion with respect to fire status, passage capacity, psychological factors and crowd flow rates. Simulations are performed on a traditional network-flow model, which incorporates the above crowd model, and fire propagation is imitated by a cellular automation model. The evacuation planning problem is then translated to a Markov decision problem, which aims to maximise the total number of evacuees and minimise the relevant risks. To solve this optimisation problem, a divide-and-conquer approach is used to divide evacuees into different groups, and multiple groups are coordinated within a Lagrangian relaxation framework. Queueing models can also be used to evaluate congestion in such systems [36,37]. A M/M/1 queueing model [38] is employed in [1] to analyse the time needed to exit with respect to congestion. Compared with closed queueing network models, this open model-based approach is more computationally efficient and can avoid long time simulations. The “time to exit” is over-estimated due to the assumption of the highest arrival rate and deterministic service times. However, this analytical model is useful in emergency procedure design and densely-populated evacuations.

In summary, regarding the aforementioned literature, most experiments in the previous work employ simulated evacuees with identical physical attributes. Hence, the analysis based on those experiments may not reflect the real situation, where evacuees actually have different velocities and health conditions. Furthermore, previous emergency navigation algorithms provide routes with the same property (shortest, safest or quickest) to civilians and ignore their specific service requirements. Therefore, evacuees tend to congregate in the main channels and cause a stampede or congestion, due to the limited space and diverse mobility. Moreover, injuries and fatalities are likely to be increased owing to the dissatisfaction of the unique service needs of evacuees. In addition, although jamming has a significant impact on the performance of emergency navigation algorithms, the previous congestion management algorithms or models are either not very realistic or too computational complex.

3. The Cognitive Packet Network

The cognitive packet network (CPN) introduced in [39–41] was originally proposed for multimedia networks with diverse end users [42]. As different classes of users may require different QoS, the traditionally best effort Internet, which satisfies a single QoS need, is difficult to adapt and can impede the performance. Therefore, CPN is presented to provide QoS-driven routing and realise self-improvement by using cognitive packets. Contrary to conventional routing protocols, in the CPN, “cognitive packets” play a predominant role in routing and flow control, other than routers and protocols. For example, cognitive packets can discover optimised routes with their predefined goals and improve QoS by learning from their own investigations and experience from other packets.

Each node within a CPN hosts a recurrent random neural network, which has been applied to many domains [43] to determine the cognitive packets' next hop among neighbour nodes, and it maintains a routing list to reserve the routes discovered by cognitive packets. CPN carries three types of packets: smart packets (SPs), acknowledgements (ACKs) and dumb packets (DPs). SPs are responsible for information collection and routing discovery with respect to a specific QoS goal. They can either choose the neighbour node decided by random neural networks as the next hop or drift randomly to explore and update routes. 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 enacted on the reverse path to take out any sequences of nodes with the same origin and destination. When an ACK reaches a node, it will update the routing list and train the random neural network [44] by performing reinforcement learning (RL) with the neural networks [45,46]. DPs are the packets that 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 resolve the "optimised QoS routes" and collect information, such as the hazard intensity or congestion level of a node. ACKs backtrack with collected information and update the excitation level of neurons at the traversed nodes. 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 the length of edges and node capacity, are 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 their neighbour nodes and detect 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.

4. QoS Metrics

The simplest metric used in the CPN-based algorithm in [47] is distance based, and it resolves the path with the shortest effective length, which is a compound metric with respect to path length and safety. Other novel metrics considered in this paper are the time metric, the energy efficiency metric and the safety metric. Previous work in [1] has proven that congestion formed at bottlenecks can temporally block evacuee fluids and has a severe impact on the performance of evacuation routing algorithms in high occupancy rates. Furthermore, congestion can impede QoS implementation by causing undesired time latency and extra energy consumption for electric powered wheelchairs. Hence, we include congestion in the new QoS metrics.

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 potential arrival instant of the evacuee. The congestion situation at a node when an evacuee reaches it 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 Predicting the potential amount of congestion encountered by an evacuee during the evacuation process.

Data: A path π explored by SPs from a node, $\pi(0)$

Result: The potential number of congestion situations encountered by an evacuee traversing across the path, π

- 1: Set the total number of congestion situations, C_{total} , to zero
 - 2: Set the total travel time, T_{total} , to zero
 - 3: **for all** the edges, $e(\pi(i), \pi(i + 1)) \in \text{path } \pi$ **do**
 - 4: /* Calculate the time cost on the edge, $e(\pi(i), \pi(i + 1))$ */
 - 5: $t_{edge} \leftarrow \frac{E^e(\pi(i), \pi(i+1))}{V_{speed}}$,
 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 a class of civilians.
 - 6: /* Compute the time cost at the node, $\pi(i)$ */
 - 7: **if** $q_0^i + \bar{\lambda} \cdot T_{\pi(i)} - \bar{\mu} \cdot T_{\pi(i)} > 0$ **then**
 - 8: /* If the congestion will remain there when the evacuee reaches the node, $\pi(i)$, then plus the time delay caused by congestion */
 - 9: $t_{node} \leftarrow \frac{q_0^i + \bar{\lambda} \cdot T_{\pi(i)} - \bar{\mu} \cdot T_{\pi(i)}}{\bar{\lambda}}$,
 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, $\pi(i)$. Term $\frac{q_0^i + \bar{\lambda} \cdot T_{\pi(i)} - \bar{\mu} \cdot T_{\pi(i)}}{\bar{\lambda}}$ is the average time calculated by Little's formula.
 - 10: /* If the predicted queue length when the evacuee reaches it is larger than zero, the arrival of the evacuee will trigger congestion */
 - 11: $C_{total} \leftarrow C_{total} + 1$
 - 12: **end if**
 - 13: $T_{total} \leftarrow T_{total} + t_{edge} + t_{node}$
 - 14: **end for**
 - 15: **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:

$$G_t = \sum_{i=1}^{n-1} \left\{ \frac{E^e(\pi(i), \pi(i + 1))}{V_{speed}} + K \left[\frac{q_0^i + \bar{\lambda} \cdot T_{\pi(i)} - \bar{\mu} \cdot T_{\pi(i)}}{\bar{\lambda}} \right] \right\} \quad (1)$$

where π represents a particular path; n is the number of nodes on the path, π ; and $\pi(i)$ is the i -th node on the path, π . $E^e(\pi(i), \pi(i + 1))$ is the effective length of the edge between node $\pi(i)$ and node $\pi(i + 1)$. V_{speed} is the average speed of this category of civilians. $K[X]$ is a function that takes the value zero if X is smaller than zero or X if X is larger than or equals 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, because the depletion of power for specific evacuees, such as disabled people in electric-powered wheelchairs, can be fatal. The maximum theoretical range with regard to different models of wheelchairs is approximately 28.98 kilometres, with an average energy consumption of 2.27 ampere-hours [48]. However, due to the unpredictability of disasters and daily driving behaviours [49], electric-powered wheelchairs can hardly be full-charged when an evacuation starts and, therefore, increase the likelihood for a disabled civilian to get stranded. The congestion ease algorithm is embedded in the energy efficiency-oriented goal function, as the acceleration and deceleration process can cause significant energy consumption when a braking event occurs. This function can be applied to electric powered wheelchairs or robots with battery power limitation.

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

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, and $\theta(\pi(i-1), \pi(i), \pi(i+1))$ is the rotation angle between edge $E(\pi(i-1), \pi(i))$ and $E(\pi(i), \pi(i+1))$.

The safety-oriented goal function discovers 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 .

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

where $1[X]$ is a function that takes a value of one or two 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 time elapsed since the hazard occurred. 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 that 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 are shown in Algorithm 2.

Algorithm 2 Predicting the potential hazard of a path.

Data: A path π explored by SPs from a node

Result: The potential hazard of the path, π

```

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

```

5. Alleviation of Oscillations

Path oscillations will occur when a sensitive metric [10] is used or the process of updating sensor readings, and the 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 it confuses civilians using visual indicators, because the guiding direction oscillates. In [41], 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.

Moreover, 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 movement depth from hindering the flexibility of the evacuation routing and from directing 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, which include random rates of movement of the hazard (fire, smoke, gases, etc.) itself.

Ten randomized iterations have been conducted for movement depth from one to 10 in a scenario with a 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. As QoS metrics can be divided into sensitive metrics and insensitive metrics [10], we employ the time metric and the distance metric as a representative of the sensitive metric and insensitive metric, respectively, to examine the influence of movement depth. Figure 1 shows that the performance of the time metric-based system reaches the peak mean value when the movement depth is three and then decreases gradually with the increase of the movement depth. This is because when movement depth is small (one or two), 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 the movement depth is too large, evacuees will use the outdated decisions and, therefore, increase the possibility of guiding evacuees near the hazard and backtracking. Figure 2 presents the influence of movement depth in the distance metric-based system. Comparing with the time metric, the distance metric is less sensitive to the variation of movement depth and takes on a fluctuant state when the movement depth is smaller than or equals six. However, the system performance decreases evidently when the movement depth is larger than six. This is because a relatively larger movement depth value will hinder the flexibility of CPN and may direct an evacuee near the spreading hazard. Therefore, victims may have no remaining time to egress with an alternate route. According to the results of the above experiments, we set the movement depth to three in our system.

Figure 1. Average number of survivors out of 120 evacuees with different movement depths in the time metric-based system. The error bars show the standard deviation of 10 experiments.

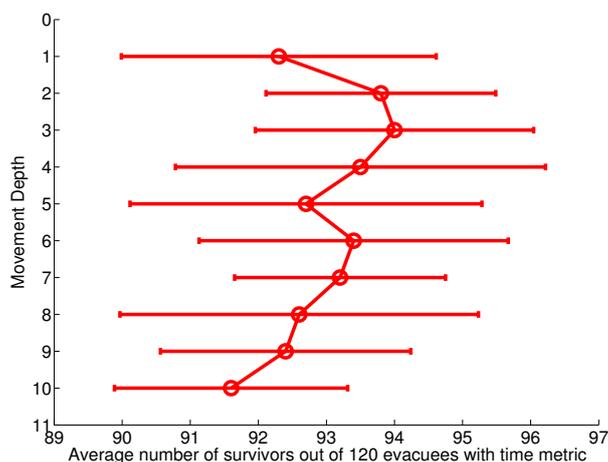
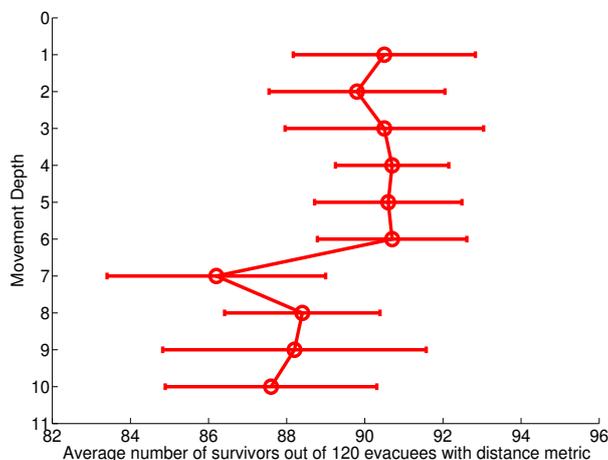


Figure 2. The average number of survivors out of 120 evacuees with different movement depths in the distance-based system. The error bars show the standard deviation of 10 experiments.

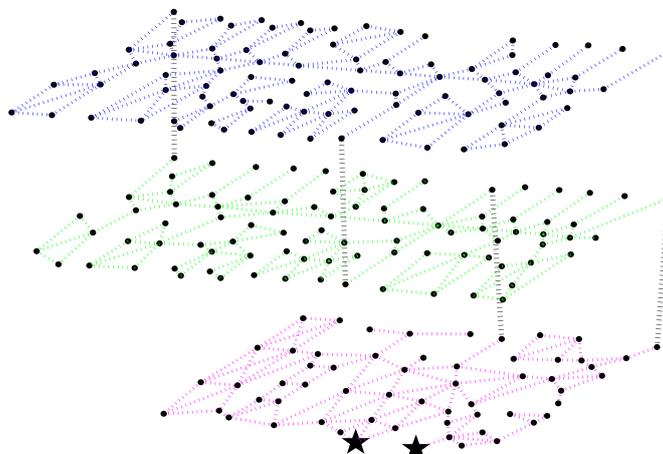


6. Simulation and Experiments

The Distributed Building Evacuation Simulator (DBES) [2] that we use 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. To improve the level of realism, we employ the congestion model introduced in [50,51] to simulate the chaotic collective motion. Each PoI where evacuees can congregate is modelled as a single server with one queue. This mimics a narrow space where only one civilian can depart at a specific time. The service time represents the time cost for a civilian to obtain the suggestion from portable devices or visual indicators. In the simulations, we set the service time to one second. When an evacuee arrives a the PoI, it will not be served until all the evacuees that came earlier have departed.

The environment is the lower floors of Imperial College’s EEEbuilding shown in Figure 3. SNs and DNs are deployed in proximity to the 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 the POI or portable devices carried by the civilians. To evaluate the performance and adaptability of decision algorithms, we assume a fire outbreaks near the eastern staircase on the ground floor that blocks a main channel. Initially, evacuees are randomly scattered in the building, and they can hear or see fire alarms and evacuate immediately. Five routing algorithms are employed: Dijkstra’s shortest path algorithm, the CPN-based algorithm with the distance metric, CPN with the time metric, CPN with the safety metric and CPN with the energy 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 other scenarios are similar to the first one, but use the CPN based algorithm to collect information and make decisions.

Figure 3. The graph-based layout of the building. The black stars represent the exits on the ground floor.



7. Results and Discussion

The experiments are conducted on scenarios with 30, 60, 90 and 120 evacuees, respectively. Dijkstra's shortest path algorithm and original distance-oriented goal function-based algorithm are performed for comparison purposes. In the first experiment, we evaluate the performance of three novel metrics, while in the second experiment, we route diverse evacuees with the relevant metric. To exclude the interferences of other factors, the evacuees in the first experiment are homogeneous with an identical velocity and initial health value. In the second experiment, we employ four types of simulated agents with different speeds, initial health and resistance to hazards.

The results of ten randomised experiments are presented in Figure 4. The error bars show the highest and lowest percentage of survivors in the ten simulations. In low occupancy rate (30 evacuees), all the CPN-based algorithms 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 gradually becomes a predominant success factor for evacuation. In this case, CPN-based algorithms outperform Dijkstra's algorithm for different reasons: CPN with the distance metric overcomes Dijkstra's algorithm, due to the nature of a decentralized system in which each decision node acts as a sub-system and updates its path recommendation at a different time. This tends to distribute evacuees to different routes and slightly ease congestion. CPN with the time metric and the energy metric embed the congestion ease algorithm, because congestion can induce a time delay and extra energy consumption (braking). CPN with the safety metric outperforms Dijkstra's algorithm mainly because it directs evacuees to the future safest path and reduces the probability of re-routing. Although CPN with the safety metric does not seek to ease congestion, it predicts the time to reach a certain location regarding the potential congestion that will be encountered. Figure 4 also indicates that CPN with the safety metric performs best in relatively low population densities (30 and 60 evacuees), while the CPN with the time metric has the largest mean percentage of survivors in high occupancy rates (90 and 120 evacuees).

Figure 4. The percentage of survivors for diverse path-finding algorithms in scenarios with homogeneous evacuees. The results are the average of 10 randomized simulation runs, and the error bars show the min/max result in any of the 10 simulations.

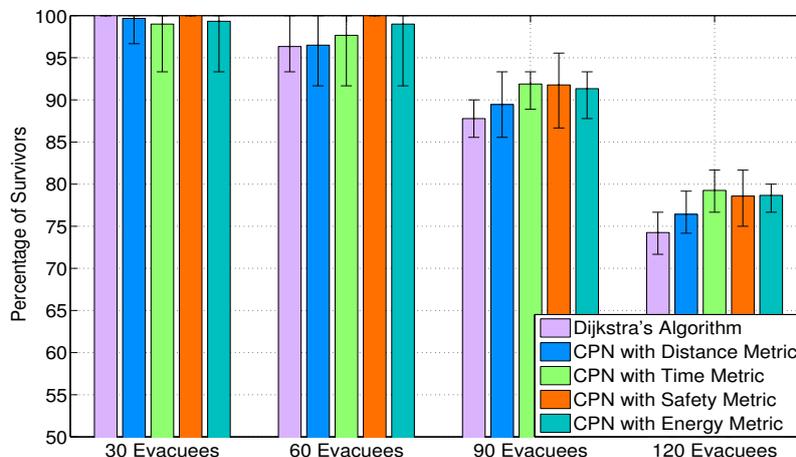
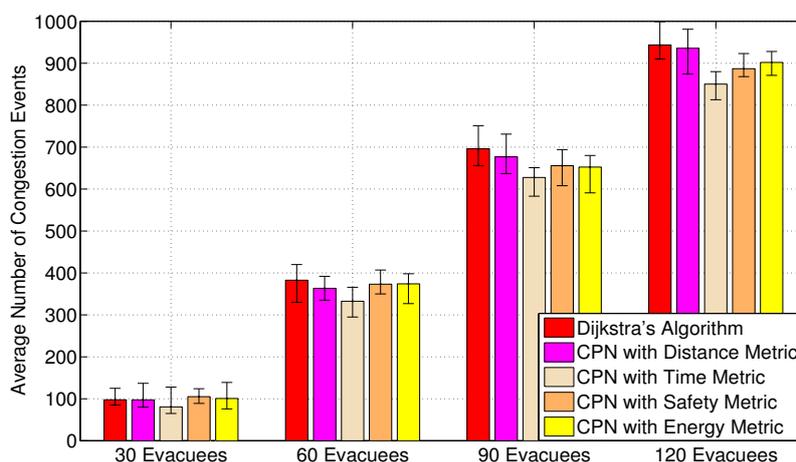


Figure 5 shows the average number of congestion situations encountered by civilians in the above evacuation processes. The results imply that CPN with the time metric can effectively alleviate congestion in comparison with other algorithms. This is because CPN with the time metric can disperse evacuees to several channels, while other algorithms tend to insist on one main channel to the exits. Although CPN with the energy metric also adopts the same congestion aware algorithm as CPN with the time metric, it produces more congestion, owing to the energy utilisation on a path largely depending on the length and sinuosity of the path. Hence, electric powered wheelchairs tends to stick to an energy efficiency route, unless a large amount of jamming occurs on the path or the hazard reaches them. CPN with the safety metric causes comparable congestion as the energy metric. This is because this algorithm not only considers a node’s current fire intensity, but the situation when it reaches that node. Actually, the safety-oriented goal function is a tradeoff between the number of congestion situations encountered and the probability of being blocked by the hazard and switching to a detour path.

Figure 5. The average number of congestion situations 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.



To evaluate if the novel goal functions improve the QoS for different categories of evacuees, we compare the results with Dijkstra’s algorithm and the original CPN algorithm with the distance metric. Figure 6 shows the average evacuation time of the three algorithms, and the error bars represent the maximum and minimum egress time of ten experiments. The results confirm that CPN with the time metric achieves 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 a similar number of survivors. However, the three metrics achieve comparable results in high population densities (90 and 120 evacuees), with a slight advantage to CPN with the time metric. This is mainly caused by the 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 the distance metric. Interestingly, although the value for CPN with the time metric is 16, alternate congestion will form near the exits as the egress becomes a new bottleneck. It indicates that in a high density environment, the bottlenecks in the system can significantly affect the average evacuation time. In other words, though the CPN with the time metric can accelerate the process for civilians to arrive in the vicinity of exits, it cannot significantly reduce the average evacuation time under the equivalent departure rate of the exits. Figure 7 shows the average health value of survivors among Dijkstra’s algorithm, CPN with the distance metric and CPN with the safety metric. The initial health value of each evacuee is 100 units, and if a civilian perishes, its health value will become zero. The results show that CPN with the safety metric achieves the highest mean health value among the three algorithms, while gaining the greatest number of survivors. The average energy consumption of each electric powered wheelchair is shown in Figure 8. Compared with Dijkstra’s algorithm and CPN with the distance metric, CPN with the energy metric consumes the least energy on average. This metric is useful for evacuation in both emergency and standard conditions.

Figure 6. The average evacuation time of ten iterations in arbitrary time units. The error bars represent the min/max values found in the 10 simulations.

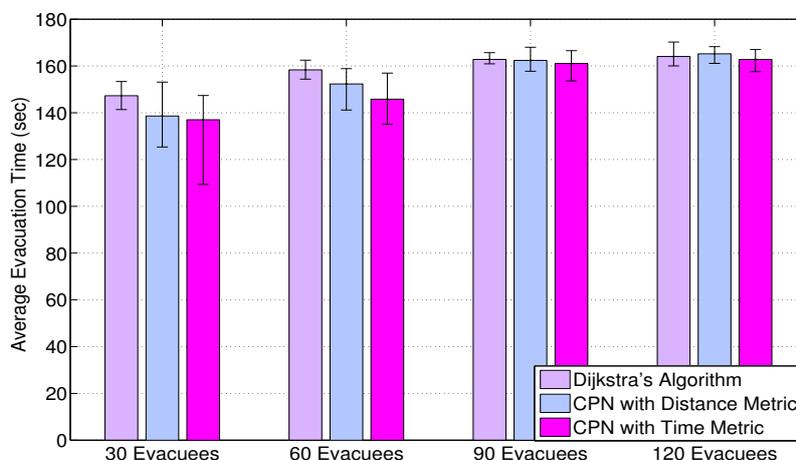


Figure 7. The average health value of survivors in ten iterations. The error bars represent the min/max values found in the 10 simulations.

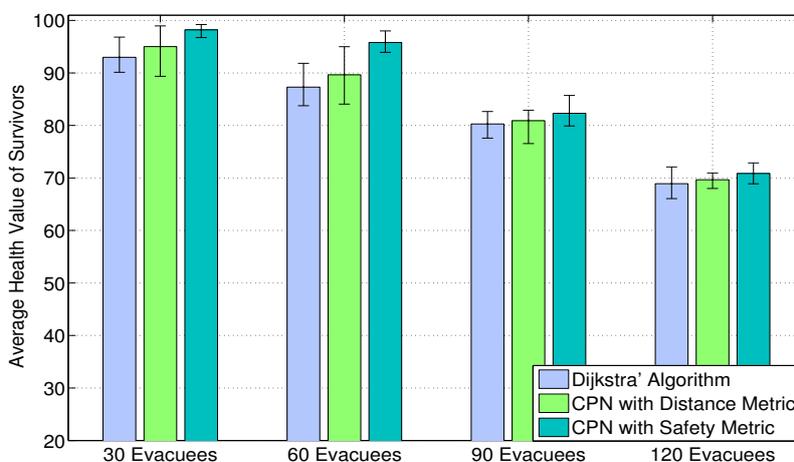
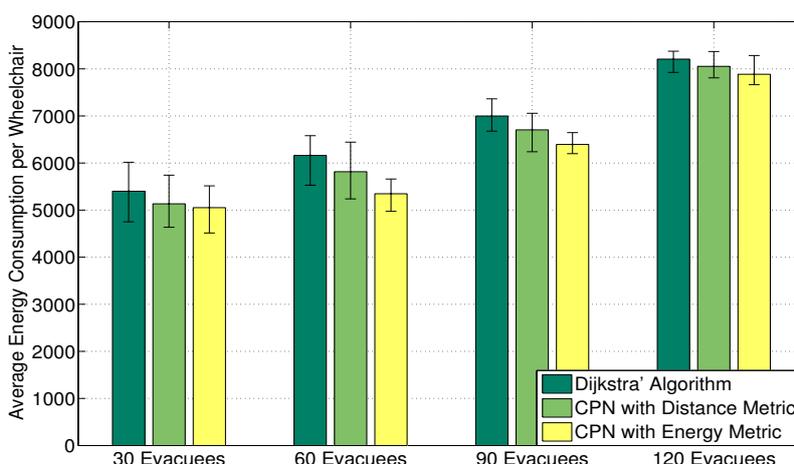


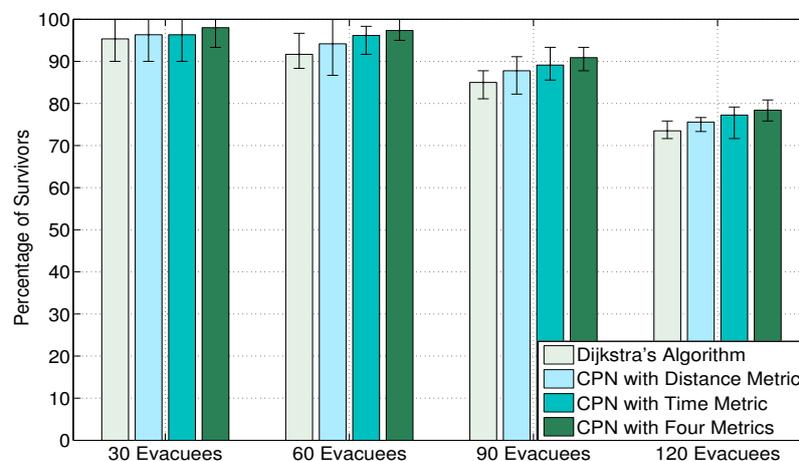
Figure 8. The average energy consumption for each electric powered wheelchair of ten iterations in energy units. The error bars represent the min/max values found in the 10 simulations.



In order to evaluate the performance of routing diverse categories of evacuees with different metrics, we develop a scenario that contains four types of evacuees: aged people, young people, children or ill people and disabled people in electric powered wheelchairs. The population ratio of the four categories of civilians is 2:5:2:1, and the speed ratio is 0.8:1:0.8:1.5. The aged people are set to a distance threshold to mimic their limited physical strength. If the displacement exceeds the distance threshold, the moving speed will reduce to one half of the initial velocity. Disabled people in wheelchairs are set at an analogous battery power limitation. If the battery power depletes, the wheelchairs will stop. In addition, the initial health value of children or ill people is 80 percent of the health level of other categories. Besides Dijkstra's algorithm and CPN with the distance metric, CPN with the time metric is also employed as a comparison for its good performance in the first experiment. The results in Figure 9 show that CPN with four metrics performs best, because it provides routes that suffice the QoS needs of different categories of evacuees. Hence, fewer evacuees will perish, due to the lack of physical strength and the depletion of battery power. Furthermore, it can form channels for different kinds of evacuees and reduce the

total evacuation time. Compared with scenarios with homogeneous evacuees, less congestion occurs in Dijkstra’s algorithm and CPN with the distance metric owing to the effect of different speeds, which is similar to the “faster-is-slower effect” [52]. This slightly improves the performance of Dijkstra’s algorithm and CPN with the distance metric. CPN with the time metric guides evacuees to the quickest path at the cost of displacing a longer distance. Hence, some aged people or wheelchairs that originate on the second floor may deplete their physical strength or energy and, therefore, causing fatalities.

Figure 9. The percentage of survivors for scenarios with heterogeneous evacuees. The results are the average of 10 randomized simulation runs, and the error bars show the min/max result in any of the 10 simulations. “CPN with four metrics” depicts the CPN-based algorithm to route four kinds of evacuees with the related metric.



Figures 10–14 show the number of visits of each edge in a scenario with 120 homogeneous evacuees. The thickness of the solid line is proportional to the number of visits. The edges that have not been traversed are shown as dotted lines. Figures 10 and 11, which use the shortest path routing metric, show that the shortest egress path on the ground floor is overused by evacuees. Most evacuees select the identical main channel to egress and cause continuous congestion. In both Dijkstra’s algorithm and CPN with a distance metric, 96% of the evacuees use the main exit in the middle, and only 4% of the evacuees escape from the alternative exit. Figure 12, 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 is more balanced between the two exits, with 79% of the evacuees departing from the main exits and 21% of the civilians choosing the alternate exit. Therefore, the probability of congestion is reduced, and the latency in the evacuation process is decreased. Figure 13 shows that CPN with the safety metric initially directs most evacuees on the first and second floors to the central staircases, which are part of the future safest route, because the fire starts near the eastern staircase on the ground floor. The safety metric also adapts the safest path with the spread of the hazard. As can be seen from the ground floor, the main channel is gradually pushed away from the fire source as the fire propagates. Figure 14 confirms that although CPN with the energy metric also employs the congestion ease algorithm, it does not switch routes as frequently as CPN with the time metric. This is because

an energy efficiency route depends not only on the predicted congestion that will be encountered on the path, but also on the length and sinuosity of the path.

Figure 10. Dijkstra's algorithm routing.

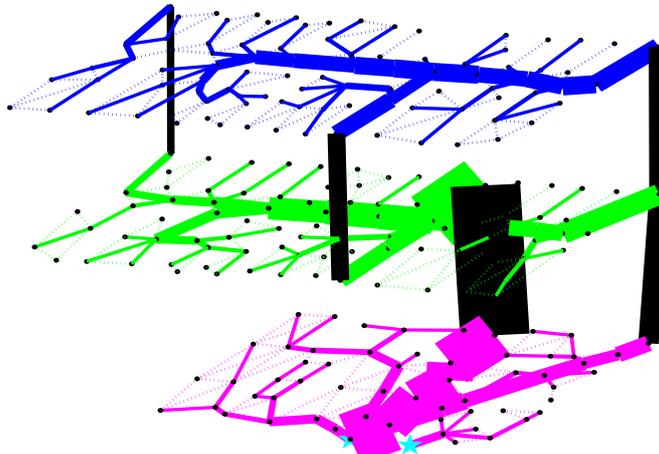


Figure 11. CPN shortest-path metric routing.

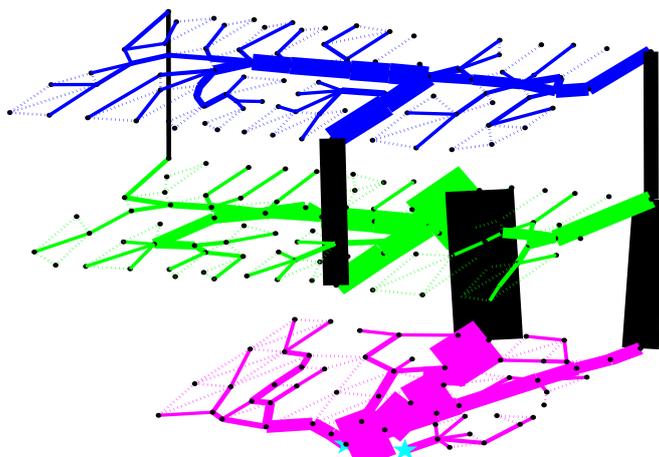


Figure 12. CPN shortest-time metric routing.

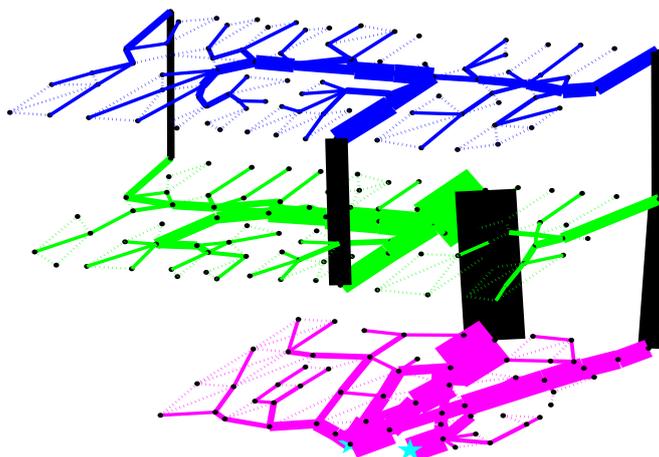
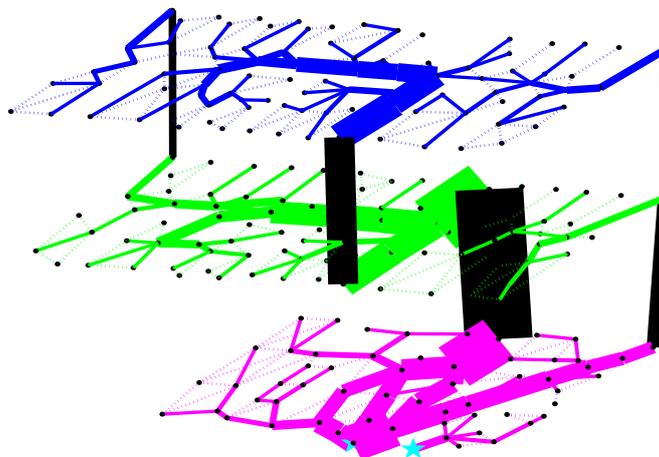
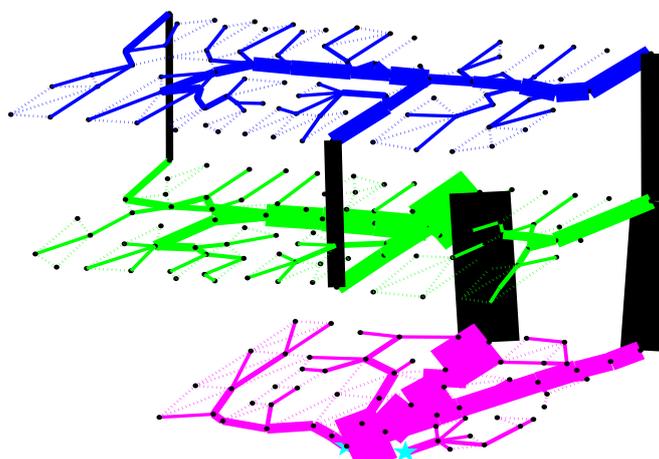


Figure 13. CPN safety metric routing.**Figure 14.** CPN energy efficiency metric routing.

8. Conclusions

We have proposed a CPN-based multi-path routing algorithm to customise routes for diverse categories of evacuees by making use of SP's characteristic that each SP can explore routes based on its own interests. Compared with other optimal algorithms, this algorithm is particularly efficient in resource-limited wireless sensor network-based systems, as CPN allocates overhead in proportion to the possibility of finding a desired path. We have proven that the three novel goal functions can improve the QoS level, as well as increase the total number of survivors by considering the effect of potential congestion. The experiment with heterogeneous evacuees indicates that the CPN-based algorithm with diverse metrics can distribute evacuees to multiple routes and achieve better performance. Due to the multi-domain sensing information required to provide evacuees with specific QoS routes, conventional wireless sensor network-based emergency response systems with limited computational power and battery power may become a bottleneck to the system performance. Hence, future research will be directed to establish an infrastructure-less evacuation system framework based on cloud computing and smart handsets. Because of the powerful computing capacity and the ability of the gathering multi-domain information of cloud-enabled systems, technologies, such as augmented reality, wisdom of the crowd and image-based localization, can be adopted for this framework.

Conflicts of Interest

The author declares no conflict of interest.

References

1. Desmet, A.; Gelenbe, E. Graph and Analytical Models for Emergency Evacuation. *Future Internet* **2013**, *5*, 46–55.
2. Dimakis, N.; Filippoupolitis, A.; Gelenbe, E. Distributed Building Evacuation Simulator for Smart Emergency Management. *Comput. J.* **2010**, *53*, 1384–1400.
3. Gelenbe, E.; Wu, F.J. Large Scale Simulation for Human Evacuation and Rescue. *Comput. Math. Appl.* **2012**, *64*, 3869–3880.
4. Gelenbe, E.; Wu, F.J. Future Research on Cyber-Physical Emergency Management Systems. *Future Internet* **2013**, *5*, 336–354.
5. Wilson, G. Logic engineering in public safety. *Eng. Crowd Saf.* **1993**, pp. 231–237.
6. Gelenbe, E.; Schmajuk, N.; Reif, J.; Staddon, J. Autonomous search by robots and animals: A survey. *Robot. Auton. Syst.* **1997**, *22*, 23–34.
7. Gelenbe, E.; Cao, Y. Autonomous search for mines. *Eur. J. Oper. Res.* **1998**, *108*, 319–333.
8. Gelenbe, E. Search in unknown random environments. *Phys. Rev. E* **2010**, *82*, 061112.
9. Gelenbe, E.; Hussain, K.; Kaptan, V. Simulating autonomous agents in augmented reality. *J. Syst. Softw.* **2005**, *74*, 255–268.
10. Gelenbe, E. Sensible decisions based on QoS. *Comput. Manag. Sci.* **2003**, *1*, 1–14.
11. Chen, W.T.; Chen, P.Y.; Wu, C.H.; Huang, C.F. A Load-Balanced Guiding Navigation Protocol in Wireless Sensor Networks. In Proceedings of the Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008, New Orleans, LO, USA, 30 November–4 December 2008; pp. 1–6.
12. Lightfoot, T.J.; Milne, G. Modelling emergent crowd behaviour. In Proceedings of the The Australian Conference on Artificial Life (ACAL), Canberra, Australia, 6–7 December 2003; pp. 159–169.
13. Pelechano, N.; Malkawi, A. Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. *Autom. Constr.* **2008**, *17*, 377–385.
14. Chalmet, L.; Francis, R.; Saunders, P. Network models for building evacuation. *Fire Technol.* **1982**, *18*, 90–113.
15. Francis, R. *A Negative Exponential Solution to an Evacuation Problem*; National Bureau of Standards, Center for Fire Research: Springfield, VA, USA, 1984.
16. Kisko, T.M.; Francis, R.L. EVACNET+: A computer program to determine optimal building evacuation plans. *Fire Saf. J.* **1985**, *9*, 211–220.
17. Hoppe, B.; Tardos, É. 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, Arlington, VA, USA, 23–25 January 1994; pp. 433–441.
18. Hoppe, B.; Tardos, É. The quickest transshipment problem. *Math. Oper. Res.* **2000**, *25*, 36–62.
19. Lu, Q.; Huang, Y.; Shekhar, S. Evacuation planning: A capacity constrained routing approach. In *Intelligence and Security Informatics*; Springer: Berlin, Germany, 2003; pp. 111–125.

20. Li, Q.; de Rosa, M.; Rus, D. Distributed Algorithms for guiding Navigation across a Sensor Network. In Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, San Diego, CA, USA, 14–19 September 2003; ACM: New York, NY, USA, 2003; pp. 313–325.
21. Pan, M.S.; Tsai, C.H.; Tseng, Y.C. Emergency guiding and monitoring applications in indoor 3D environments by wireless sensor networks. *Int. J. Sens. Netw.* **2006**, *1*, 2–10.
22. Tseng, Y.C.; Pan, M.S.; Tsai, Y.Y. Wireless sensor networks for emergency navigation. *Computer* **2006**, *39*, 55–62.
23. Okazaki, S.; Matsushita, S. A Study of Simulation Model for Pedestrian Movement with Evacuation and Queuing. In Proceedings of the International Conference on Engineering for Crowd Safety, London, UK, 17–18 March 1993; pp. 271–280.
24. Bo, Y.; Yong-gang, W.; Cheng, W. A Multi-Agent and GIS Based Simulation for Emergency Evacuation in Park and Public Square. In Proceedings of the International Conference on Computational Intelligence and Security, IEEE CIS'09, Beijing, China, 11–14 December 2009; Volume 1, pp. 211–216.
25. Park, V.D.; Corson, M.S. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In Proceedings of the INFOCOM'97, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Kobe, Japan, 7–12 April 1997; Volume 3, pp. 1405–1413.
26. Filippopolitis, A.; Gelenbe, E. A Distributed Decision Support System for Building Evacuation. In Proceedings of the 2nd Conference on Human System Interactions, HSI'09, Piscataway, NJ, USA, 21–23 May 2009; pp. 323–330.
27. Pelusi, L.; Passarella, A.; Conti, M. Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks. *Commun. Mag. IEEE* **2006**, *44*, 134–141.
28. Gelenbe, E.; Görbil, G. Opportunistic communications for emergency support systems. *Procedia Comput. Sci.* **2011**, *5*, 39–47.
29. Gelenbe, E.; Loukas, G. A self-aware approach to denial of service defence. *Comput. Netw.* **2007**, *51*, 1299–1314.
30. Ford, D.; Fulkerson, D.R. *Flows in Networks*; Princeton University Press: Princeton, NJ, USA, 2010.
31. Ahuja, R.K.; Magnanti, T.L.; Orlin, J.B. *Network Flows: Theory, Algorithms, and Applications*; Prentice Hall: Englewood Cliffs, NJ, USA, 1993.
32. Lu, Q.; George, B.; Shekhar, S. Capacity constrained routing algorithms for evacuation planning: A summary of results. In *Advances in Spatial and Temporal Databases*; Springer: Berlin, Germany, 2005; pp. 291–307.
33. Chen, P.Y.; Kao, Z.F.; Chen, W.T.; Lin, C.H. A Distributed Flow-Based Guiding Protocol in Wireless Sensor Networks. In Proceedings of the 2011 International Conference on Parallel Processing (ICPP), IEEE, Taipei, Taiwan, 13–16 September 2011; pp. 105–114.
34. Pizzileo, B.; Lino, P.; Maione, G.; Maione, B. A New Escape Routing Strategy for Controlling Evacuation from Buildings. In Proceedings of the 2010 IEEE American Control Conference (ACC), Baltimore, MD, USA, 30 June–2 July 2010; pp. 5124–5130.

35. Wang, P.; Luh, P.B.; Chang, S.C.; Sun, J. Modeling and Optimization of Crowd Guidance for Building Emergency Evacuation. In Proceedings of the IEEE International Conference on Automation Science and Engineering, CASE 2008, Washington, DC, USA, 23–26 August 2008; pp. 328–334.
36. Gelenbe, E.; Muntz, R.R. Probabilistic models of computer systems. Part I: Exact Results. *Acta Inform.* **1979**, *7*, 35–60.
37. Gelenbe, E. Probabilistic models of computer systems. Part II: Diffusion approximations, waiting times and batch arrivals. *Acta Inform.* **1979**, *12*, 285–303.
38. Gelenbe, E.; Pujolle, G.; Nelson, J. *Introduction to Queueing Networks*; Wiley Chichester: Chichester, UK, 1987.
39. Gelenbe, E.; Lent, R.; Xu, Z. Towards networks with cognitive packets. In *Performance and QoS of Next Generation Networking*; Springer: London, UK, 2001; pp. 3–17.
40. Gelenbe, E.; Lent, R.; Nunez, A. Self-aware networks and QoS. *Proc. IEEE* **2004**, *92*, 1478–1489.
41. Gelenbe, E. Steps toward self-aware networks. *Commun. ACM* **2009**, *52*, 66–75.
42. Sakellari, G. The cognitive packet network: A survey. *Comput. J.* **2010**, *53*, 268–279.
43. Timotheou, S. The random neural network: A survey. *Comput. J.* **2010**, *53*, 251–267.
44. Gelenbe, E.; Timotheou, S. Random neural networks with synchronized interactions. *Neural Comput.* **2008**, *20*, 2308–2324.
45. Gelenbe, E.; Seref, E.; Xu, Z. Simulation with learning agents. *Proc. IEEE* **2001**, *89*, 148–157.
46. Gelenbe, E. Natural Computation. *Comput. J.* **2012**, *55*, 848–851.
47. Bi, H.; Desmet, A.; Gelenbe, E. Routing Emergency Evacuees with Cognitive Packet Networks. In Proceedings of the 28th International Symposium on Computer and Information Sciences (ISCIS'13), Paris, France, 28–29 October 2013; Springer: London, UK, 2013.
48. Rentschler, A.J.; Cooper, R.A.; Fitzgerald, S.G.; Boninger, M.L.; Guo, S.; Ammer, W.A.; Vitek, M.; Algood, D. Evaluation of selected electric-powered wheelchairs using the ANSI/RESNA standards. *Arch. Phys. Med. Rehabil.* **2004**, *85*, 611–619.
49. Cooper, R.A.; Thorman, T.; Cooper, R.; Dvorznak, M.J.; Fitzgerald, S.G.; Ammer, W.; Song-Feng, G.; Boninger, M.L. Driving characteristics of electric-powered wheelchair users: How far, fast, and often do people drive? *Arch. Phys. Med. Rehabil.* **2002**, *83*, 250–255.
50. Filippopolitis, A. Emergency Simulation and Decision Support Algorithms. PhD Thesis, Imperial College London (University of London), London, UK, 2010.
51. Gorbil, G. Opportunistic Communications for Emergency Support. PhD Thesis, Imperial College London, London, UK, 2013.
52. Helbing, D.; Farkas, I.; Vicsek, T. Simulating dynamical features of escape panic. *Nature* **2000**, *407*, 487–490.