Routing and Guidance of Evacuees: a Reactive and Proactive Approach

Antoine DESMET
Dept of Electrical and Electronics Engineering
Intelligent Systems and Networks Group,
Imperial College,
London,
United Kingdom,
a.desmet10@imperial.ac.uk

April 7, 2014

Abstract

In this paper we consider algorithms and systems which dynamically guide evacuees towards exits during an emergency to minimise building evacuation time. Often, a static evacuation plan based on the "shortest safe path" routing approach is inadequate, especially when congestion is a predominant factor. We therefore present and simulate systems which distribute and balance traffic loads to improve the overall flow of evacuees across the building.
We compare a "reactive", real-time approach for congestion management to a forecast-based "proactive" approach. Our evacuation simulations show that these methods are most effective when evacuees are nod distributed evenly across the building. While both methods make reduce evacuation times, the simulations show that a real-time approach requires frequent corrections and is prone to route oscillations, which results in needlessly long and sometimes incoherent evacuation routes. On the other hand, the forecast-based method can operate off-line and is stable, but this approach may become ineffective if the motion model is inaccurate as the ground truth will deviates congestion forecast. We also compare both approaches in terms of hardware requirements, and more.
Finally, we consider guiding evacuees using dynamic exit signs, whose pointing direction can be controlled. Dynamic signs can readily be used with the real-time "reactive" approach, but the proactive approach requires individualised guidance since routes are assigned on an individual basis. This is incompatible with dynamic signs; therefore we propose a novel flow-based algorithm to overcome this limitation. Once again, we simulate both systems, compare their performance, and review their practical limitations. For both approaches, we find that updating the sign's display more often improves performance, but this may reduce evacuee compliance and make the system inefficient in real-life conditions.

1 Introduction

Events such as fires, explosions or even bomb threats put the lives of building occupants at risk and mandate their immediate evacuation. The process of evacuating a building should ideally take the least amount of time, to limit the occupants' exposure to whichever hazard(s) triggered the evacuation. As evacuees may get lost during the evacuation or unknowingly take detours, an evacuation plan is generally designed to guide evacuees towards the exits. This plan ultimately determines the direction and location of the static signs fitted in the building to guide evacuees.

Evacuation plans generally tend to guide evacuees so that the building evacuation time is minimal, taking into account the capacity of each exit path, the most likely locations of evacuees, their distance to exits, etc. When planning for an evacuation, a scenario where the building is filled to capacity is often considered: this is effectively a conservative approach, as evacuating larger crowds inherently poses a higher risk. Thus the resulting evacuation plan is mostly designed to optimise the outcome of the "worst-case" scenario.

This static approach to guiding evacuees is inherently limited: some signs may direct evacuees towards areas of the building which are affected by the hazard. Evacuees may also be unevenly disseminated across the building, and exits covering the crowded areas of the building will experience higher levels of congestion, which ultimately slows down the evacuation.

This paper proposes to overcome the limitations of static evacuation schemes by using a *dynamic* evacuee assistance system. The proposed guidance system senses the current conditions in the building (position of evacuees, hazards, etc.), produces an evacuation plan tailored to the current scenario, and directs evacuees according to this optimal plan. The system's objective is to divert evacuees from exit paths which are hazardous or those which are experiencing high levels of congestion, and disseminate them evenly on all safe paths. Our initial research [6] investigated some capacity-constrained routing algorithms for the evacuee guidance system. Following this, we proposed methods to direct evacuees using dynamic signs, based on the routing component's output [7], while Cognitive Packet Network based routing evacuation is also studied in [2] and other fast optimisation techniques are investigated in [20].

This paper mainly aims at providing an overview of the whole research project, and also provides a deeper review of the techniques we developed to guide and direct evacuees in the building. Hence this paper focusses on the *decision* and *control* components of dynamic evacuee guidance systems. We first present a side-by-side comparison of a "reactive" and "proactive" approach to evacuee routing in capacity-constrained environments. The second part of this paper considers means to inform evacuees of the optimal routes using dynamic signs, whose pointing direction can be controlled by the system.

2 Background

Since the lives of individuals depend on the evacuee guidance system, reliable operation is an essential requirement. This requirement is particularly challenging, as the environment in which such systems are expected to operate is likely to cause component failure. Indeed, the fire, flood or blast which may trigger the evacuation can also damage hardware, disrupt communications or cut power supplies. Based on this observation, most designs presented in the literature place an emphasis on fault-tolerance, often achieved through decentralised and distributed operation and with minimal use of infrastructure [8, 22].

For instance, the concept of opportunistic communications [27] reduces the guidance system's reliance on building infrastructure by using devices carried by evacuees to form a network, and carry out route calculations. This approach does not rely on a sensor network to gather hazard sensor readings, instead, readings are transmitted opportunistically to evacuees passing by, through a short-range wireless link. In turn, as evacuees walk past each other, their devices exchange information, which contributes to the dissemination of the sensory readings. Each time a new piece of information becomes available, the evacuee's device independently runs a routing algorithm which identifies the current optimal exit path, based on the information gathered so far. While this approach theoretically provides a high level of resilience as it makes minimal use of "vulnerable" infrastructures [26], its performance depends on the number and mobility of evacuees since they are used to convey the information. Opportunistic networks are also delay-prone, and adequate information dissemination can never be guaranteed. To perform load-balancing in environments as dynamic as evacuations, a routing algorithm must have a complete view of the congestion in the building, possibly with low latency: these requirements are somewhat incompatible with opportunistic communications.

In many cases, using the building's infrastructure improves connectivity and reduces latency, however the reliance on such infrastructure increases the risk of system fault due to component failure. This can be mitigated by designing a decentralised system whose operation does not depend on any single node, and which is also able to cope with the failure of some nodes and operate in degraded mode. For instance, systems based on the concept of "gradient-descent" [1, 42, 10] routing are decentralised, operate in a distributed manner over a group of nodes disseminated in the building, and only require a short-range communication media to operate. This concept assigns a "potential" to each area of the building which reflects their perceived distance to the exit (path cost). Once this "potential field" is set up, finding the shortest way to the exit is straightforward: it consists of following whichever direction yields the greatest drop in potential, until an exit is reached. An algorithm allows each computational node to autonomously determine its area's potential only by comparing the potential of "one-hop" neighbours. The process begins at the exit nodes (which set an arbitrary low potential) and gradually propagates away from the exits until the entire network of nodes has converged. However this convergence mode means each time the cost of an edge varies, the potential of all "upstream" nodes will be affected and require a new convergence process. This is not a concern if the path metric is static (e.g. path distance), however, if the metric include dynamic measurements (hazard levels, congestion) the rate at which variations occur may exceed the time required for the nodes to converge. In this case, the network may be in a permanent transient state and will provide inconsistent guidance: thus these algorithms are not suitable to highly dynamic routing problems like load-balancing. While this system can, in theory, operate with any path metric, compounding heterogeneous measurements (distance, hazard, congestion) into one metric is not trivial: it requires each component of the metric to be scaled according to its importance [4].

Techniques inspired from the field of network optimisation are also relevant to the evacuation problem: they can be applied to form the core routing component of the evacuee assistance system. Most algorithms require a graph representation of the building with estimations of edge capacity and free-flow walking times. Francis' "principle of uniformity" [15] states that a building is evacuated in minimal time if all exits clear at the same time. This can be done by allocated evacuees to exit paths proportionally to the exit's capacity. However, this principle is based on some strong assumptions, such as the fact that all evacuees have several alternative evacuation paths, and that these paths have similar length. This mostly limits this principle to simple evacuation scenarios, or to calculate lower bounds. The Min Cut-Max Flow algorithm [14, 9] determines the set of paths which maximises the throughput of a network in steady-state: this is useful to identify the bottlenecks and to estimate the maximum flow of all combined evacuation paths. However, an evacuation has a finite (and often short) duration, and this makes a steady-state result mostly irrelevant. The time dimension can be incorporated by "expanding" the building graph: this consists of replicating the graph over several time-steps. With the time-expanded graph constraining both edge capacity and transit times, linear optimisation algorithms can identify a set of optimal path allocations for a particular evacuation scenario [31], or even find an optimal combination of quickest and shortest paths [28]. However, the time-expansion process greatly increases the search space, and despite some efforts to reduce computation times by allowing sub-optimal solutions, this solution is often prohibitively complex [37] and results in long computation times. This is incompatible with our system's requirement to have a routing strategy ready within seconds of the evacuation signal going off: cleary, we can not expect evacuees to wait in place while a solution is being computed. Furthermore, most graph-based algorithms cannot be distributed: typically, a single machine solves the entire route assignment problem. If this particular machine fails, the entire assistance system may become inoperative. Because of this relatively high potential for failure, this approach cannot be considered for a critical system like the one we are developing.

While most work found in the literature focus on the routing component, an evacuee assistance system is in fact a Cyber-Physical System (CPS) [5] composed of several components, which all present specific challenges [23]. In particular, the system is expected to deal with heterogeneous information sources, variable delays, failures, or even the fact that there are multiple conflicting objectives [29, 32].

3 Proposed approaches: Proactive and Reactive

In this section, we compare two distributable and decentralised approaches to form the core evacuee routing component. The first approach is "reactive": it uses real-time congestion measurements to detect load imbalances and attempts to addresses them. The other approach is "proactive": it uses a congestion forecast, updated after each route assignment, to assess future congestion and assign evacuees to the quickest paths. In both cases we use the same routing algorithm: the difference between both approaches lies in the calculation method for the path metric.

3.1 Routing Algorithm

We use the Cognitive Packet Network (CPN) concept [18, 24] to find optimal routes for evacuees. CPN was originally designed to route data packets in computer networks, but similarities in models between human navigation and data packet routing mean it can be applied to our evacuation problem. Unlike some of the algorithms presented earlier – which find optimal solution through extensive graph searches – CPN has the particularity of intelligently allocating route discovery overhead to the areas of the graph deemed most worthy. One of CPN's aim is to achieve an optimal balance between routing overhead and solution optimality or latency. The graph exploration and knowledge update is carried out by dedicated exploratory packets, referred to as "Smart Packets" (SP). This discovery process is decentralised: any node in the network is free to issue SPs, and each node also independently decides where to forward incoming SPs. Once an SP reaches its target, it backtracks and informs all nodes visited along the path: this information allows every node to build its own routing table. The way in which SPs are guided is a critical factor to CPN's performance. While forwarding SPs at random means that the entire graph will eventually be explored, these SPs will effectively perform a random walk, and as a result, convergence may be slow. In contrast, heuristic methods can guide SPs towards areas of the network deemed more worthwhile for faster convergence [21]. While some of these techniques require prior knowledge or have limited performance, hosting a Random Neural Network (RNN) [16, 17] on each node allows the nodes to "learn" which directions appear as most worthwhile. The RNNs are trained through reinforcement learning: if the decision to route an SP in a particular direction eventually results in the discovery of a better path, the neuron associated to this direction is reinforced. Nodes also occasionally forward SPs at random to prevent overtraining the neural network. The advantages of this approach are:

- Robustness: the routing algorithm is decentralised, and no single node failure can prevent the system from operating. Instead, faults will gradually reduce the system's performance as paths going through areas covered by failed nodes will no longer be considered [39].
- Efficiency: the RNNs efficiently guide the network exploration and updating process. The SP transmission rate can be tuned to meet a specific routing overhead budget and performance requirements: sending more SPs increases performance (lower latency, better routes), but also generates more overhead, and vice-versa.
- Operation: unlike routing algorithms which involve a convergence process which blocks the entire system, CPN resolves sub-optimal routes very quickly, and constantly refines them. The longer CPN is allowed to search, the more SPs the nodes can send, and the better the routes.

3.2 Routing Metrics

Having introduced our routing algorithm, we now present our routing metrics. Both of our metrics rely – to various extents – on a framework to provide an estimation of the number of evacuees in each area of the building. At this stage, we abstract this sensing component to focus on the routing decision and evacuee control system.

3.2.1 Reactive Routing

Reactive routing is based on real-time congestion measurements in the building. These measurements allow CPN to compare paths based on their *current* transit time. This approach is based on the assumption that congestion in the building is steady, and that transit delay does not vary significantly from the time a path is assigned, to the moment the evacuee actually travels along this path. In reality, transit delay is a sensitive metric: it varies based on the number of times this route is assigned. As a result, the steady-state assumption is unlikely to be *strictly* validated, and the algorithm will need to to issue route corrections regularity as conditions change. This also requires a constant supply of information regarding the localisation of evacuees.

Furthermore, as the reactive approach ignores the *delayed* feedback loop between path assignment and the corresponding rise in congestion, and does not take into account the available capacity on each route while diverting evacuees from congested routes, it may be prone to oscillations.

3.2.2 proactive Routing

Unlike the on-line Reactive path metric, the Proactive approach attempts to forecast the congestion that arises as a result of each routing decision. This forecast can then applied to determine, ahead of time, the transit delay of any candidate path. As a result, the system operates off-line and information on the evacuee's localisation is only required at the beginning of the evacuation: afterwards, the system relies on an evacuee mobility model to determine future congestion. The forecast is built by "reserving" future capacity on every edge of a newly-assigned path, at the expected time of arrival. The edge's capacity and maximal allowable flow are enforced by limiting the number of capacity reservation which can be made within a discrete time-step. An edge which has all its capacity reserved for a time-step is *saturated*, i.e. it cannot accept any more traffic during this time-step. In this case, the path transit time is increased to represent the evacuee's queuing time until the next time-step with spare capacity. This concept of future capacity reservation has been applied in the CCRP algorithm [37, 34, 35], which relies on modified version of Dijkstra's algorithm to identify the best route assignment. This approach is centralised and not distributable, which may result into poor scalability and low robustness. Our approach uses CPN, and its distributed nature means that the routing decisions can be made by nodes disseminated in the building, which could also embed hazard sensors and displays to inform and direct evacuees.

3.3 Experimental Results

We use the DBES (Distributed Building Evacuation Simulator) [11] to validate our proposed evacuation system. The DBES is a simulation platform which was launched in the late 2000s, and has since then been used to simulate city-wide evacuations with a high level of precision [25], search-and-rescue operations[13], or interfacing with a real-life

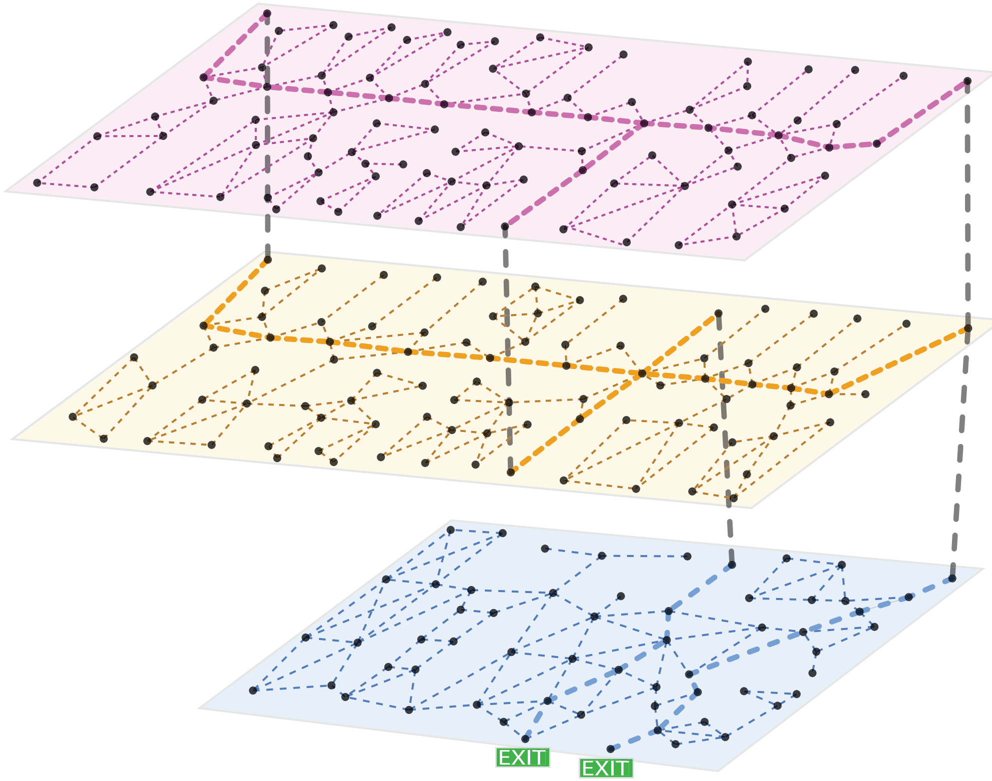


Figure 1: Three dimensional representation of the building graph used in the simulations.

wireless sensor network [12]. The building graph (Figure 1) represents the first three floors of a faculty building, with a large lobby on the ground floor and mixed classroom and office spaces on the two upper floors. In real-life evacuations, evacuees do not immediately respond to the evacuation signal [40, 41], and we model this behaviour by applying a randomised delay before allowing the evacuees to move towards the exit. To obtain an accurate representation of the system's performance under diverse conditions, each scenario is run 20 times with randomised evacuee departure points. At this stage, we assume evacuees are rational and follow the advice provided by the evacuee guidance system. We aim to model more realistic behaviours (such as following a leader [30] or acting selfishly [36]) in later stages of this project. This experiments also abstract the way in which evacuees are informed of the routing algorithm's decisions, as this will be covered in the second part of this paper.

In order to make the evacuation non-trivial, all evacuees are concentrated on the first floor: under an intuitive "shortest-path" routing approach, the building's central staircase would be over-used, while the other staircase on the right of Figure 1 would remain virtually empty. This scenario challenges the capacity-constrained routing algorithm: to achieve the quickest flow, a precise amount of evacuees must be diverted onto the other staircase.

We use Chen and Hung's formula [3], to get a lower bound on evacuation time of n individuals through one of the staircases: $T_P(n) = T_P(1) + (n-1)t_{max}$: the transmission time of n units through a path P equals to the "lead-time" of P and the time to clear $n-1$ units through the path's bottleneck, i.e. the edge with the highest transit time t_{max}. We evaluate different route assignment options and select the one that yields the lowest evacuation time. The lower-bound evacuation time appears in purple on Figure 2. The Figure shows near-identical, and near-optimal results for both approaches. While the evacuation times are comparable, Figure 3 shows large discrepancies in evacuation path lengths: routes followed by evacuees guided by the Reactive system are significantly longer, compared to its Proactive counterpart. An in-depth analysis of individual evacuee's paths reveals that the system tends to shifts evacuee back and forth between parallel bottlenecks, in particular, the two staircases leading to the first floor. This is a form of routing oscillation: when the Reactive system detects an imbalance in loads, it attempts to divert evacuees towards a path with a lighter load level. However, as the system ignores the processing capacity limits on each path, it diverts all evacuees by default – in theory, the following route corrections will interrupt the redistribution process once the loads have become balanced again. However, there is a delay in the process of shifting evacuees from a path to another: the effects

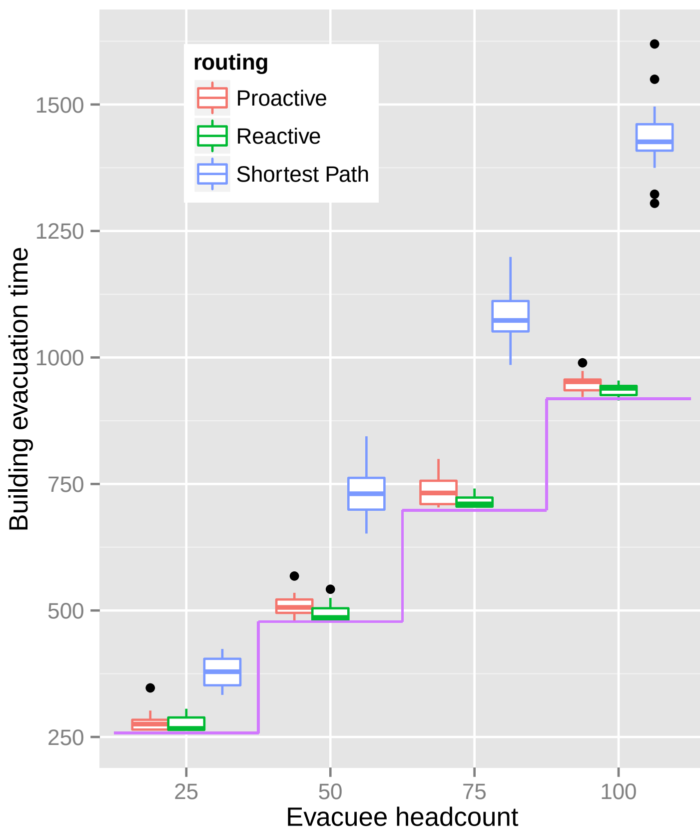


Figure 2: Box-plot representation of the building evacuation times under different routing approaches and different evacuee headcount. The Shortest-path routing approach is provided for comparison purposes. The “whiskers” represent the upper and lower quarters of the sample, and the “box” represents the remaining 50%, with the horizontal line marking the median. The purple line shows the lower-bound evacuation times.

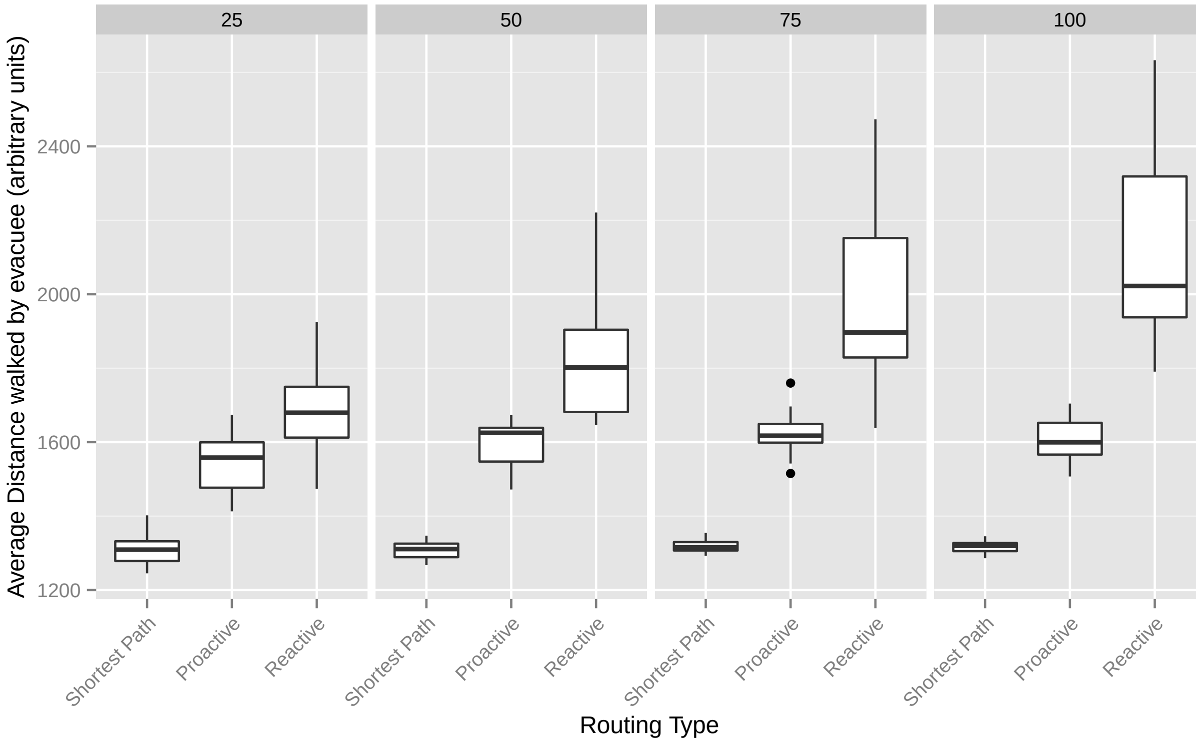


Figure 3: Average distance walked by evacuees under different routing approaches.

of the redistribution decision do not appear immediately. As a result, the system tends to divert too many evacuees. This effectively reverses the original situation: the path which was previously under-loaded becomes overloaded, and vice-versa. This creates an oscillating pattern, which continues until all evacuees have cleared the bottlenecks. While our attempts to implement relevant oscillation-damping techniques [19] were successful, we found that each method has a "damping" parameter which influences the system's performance. We determined that the optimal value of this parameter depends on the number of evacuees, their initial distribution, and the building topology. This makes it difficult to guarantee adequate performance in any condition.

3.4 Discussion

We have seen that both approaches effectively reduce building evacuation times, by balancing the load on all available paths. However, behind apparently equal performance levels, we have uncovered major differences between both approaches.

Using a Reactive approach to route evacuees is likely to result in incoherent directives, as the routing algorithm oscillates. Limiting these oscillations is possible, but requires tuning parameters in an "ad-hoc" manner. The requirement for a constant feed of information on the distribution of evacuees in the building can also be seen as a drawback: the network of sensors may be particularly vulnerable during a fire, and faults would impact the guidance system's performance. On the other hand, using real-time measurements lets the system automatically react to path suddenly becoming impassable, adapt to variations in evacuee walking speeds, or to evacuees which do not follow the signs' advice. The proactive approach, by comparison, is "rigid": the congestion forecast elaborated at the start of the evacuation cannot be corrected at a later stage using real-time measurements. If the evacuee mobility model is inaccurate, the forecast will drift from the ground truth and the routes will become less and less optimal. While the use of a model confers stability to the routes issued by the Proactive approach and reduce its reliance on sensors, the system may become ineffective if unforeseen events invalidate the congestion forecast.

Clearly, both approaches have their advantages and drawbacks, and the potential of a solution blending both should be evaluated. This could be based on the Kalman Filter concept [33], which combines a model-based prediction with regular “corrections” obtained from physical measurements. However in our case, it is unclear how corrections can be made without re-computing the entire forecast: modifying the capacity reservations made for a single individual may have a ripple effect and affect all subsequent reservations, especially if the path is saturated.

4 Evacuee Guidance

In the previous section, we have compared two approaches to build the guidance system’s routing component. Yet our system consists of many more components, for instance, a sensing system is required to inform the routing algorithm of the evacuee’s locations, and a control system must steer evacuees accordingly. In this section, we consider an evacuee control system based on dynamic signs, whose pointing direction can be controlled. Some designs have been suggested to integrate dynamic signs to building automation systems [43], and virtual reality tests have been conducted to evaluate the response of evacuees to these signs [38]. Yet research projects in the literature generally abstract the evacuee control system, for example by assuming evacuees carry communication devices, such as smartphones, which can be used to display their exit path. We believe this solution is not suitable: issues such as ownership of the device, state of charge and device compatibility could limit the system’s effectiveness. Expecting evacuees to focus on the screen of a small device is also fundamentally unsafe: they should instead be paying attention to their environment to avoid falls or stampedes. By comparison, displays integrated into the environment are safer, and also more likely to be seen. As their location and orientation is fixed and known, they do not suffer from localisation or orientation errors which may affect personal devices. They can also be hardened against likely hazards, and fitted with required levels of redundancy to meet robustness requirements.

However, dynamic exit signs are not directly compatible with our routing component: exit signs mandate a *hop-by-hop* routing approach, while CPN performs *source-routing*. Furthermore, exit signs cannot display individualised information: all evacuees in the sign’s coverage area will see the same information. This is problematic as our Proactive routing component issues routes on an individual basis.

4.0.1 Dynamic Sings and Reactive Routing

Instead of providing evacuees with a complete route – only to correct it at regular intervals – we let each node independently decide the next-hop direction to forward *local* evacuees. In practice, this is equivalent to issuing route corrections at *every* hop: the sign is set according to the first hop of the best path known to the Reactive CPN at the time. We assume the signs are placed slightly before the area where queues tend to form: this means that the system can only divert *incoming* evacuees, and has no influence on evacuees which have already joined a queue. By reducing the system’s ability to shift large groups of evacuees, we expect to see a reduction in oscillations.

4.0.2 Dynamic Signs and Proactive Routing

The proactive approach performs source-routing, and issues routes on an individual basis. Theoretically, the most straightforward method to combine this routing component with dynamic exit signs would be to identify which evacuee is walking past a sign, look up the route assigned to this individual and display the relevant next-hop instruction on the sign. This is impractical for several reasons, and particularly because it requires the identification of every evacuee which walks past each sign.

Assuming our system is only used to route able-bodied users (disabled persons would be routed using a different system) and that these users can categorised into one broad class (in terms of walking speed, path requirements, etc.), a particular route assigned by the routing component to a particular evacuee can be arbitrarily reassigned to any other located at the departure point. As we consider evacuees have *similar* walking speed and mobility characteristics, exchanging paths between two evacuees as they walk past each other will not impact the congestion forecast either. If the graph is dense enough and time steps small, we can consider that paths can be arbitrarily reassigned amongst evacuees arriving at the same graph node within the same time-step. For instance, if the routing algorithm has forecasted 10 arrivals in a particular time-step, and an analysis of the routes reveals 6 evacuees continue with a left turn, and the remaining 4 with a right turn, the sign does not need to match the next-hop direction to the original individuals, but instead could simply indicate a right turn to the first 6 evacuees showing up in the time step, and left to the 4 others.

We have removed the requirement for individual identification, but this approach still requires counting the number of evacuees walking past each sign in every time-step. Instead of physically counting evacuees, we can estimate their number based on elapsed time and their arrival rate. Time measurement is trivial, and the mean arrival rate over a time step can be inferred based on the number of reservations in a time step, and its duration. Assuming the evacuee's rate of arrival does not vary significantly during a time-step; each sign displays a direction for a duration proportional to the number of evacuees the system intends to sent in this direction. Considering the previous example, the sign would point towards the right for 60% of the time-step's duration, and left for the remaining 40%.

This solution does not require any additional sensors – beyond what is required for the Proactive routing approach – however it relies on several assumptions, in particular, that the flow of evacuees is somewhat constant, and the routing probabilities are fixed within a time-step. These assumptions are less likely to be valid for long time-steps, which is why we decided to explore the influence of the time-step duration in our experiments.

4.1 Results

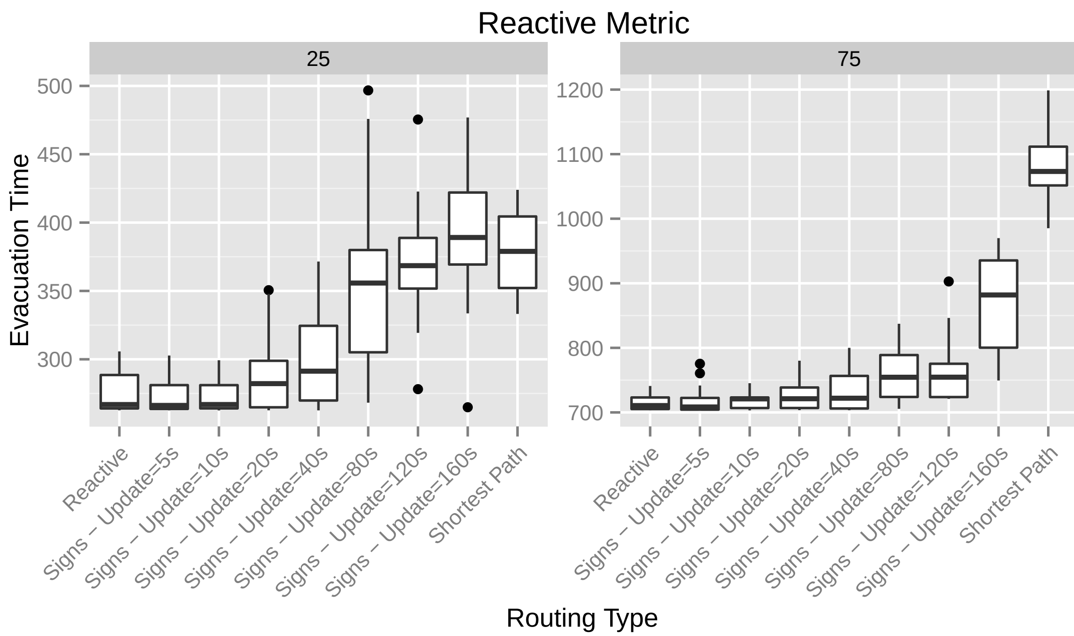
We add the "dynamic sign" component to the simulations introduced in the previous section, and analyse the system's performance. In this process, we also explore the effect of the signs' update rate on the overall system's performance: we believe this is a sensitive parameter as it may affect the evacuees' compliance and the system's performance. The sign's update rate can be modified through the time-step duration (T_{STEP}, Proactive), or by imposing a minimal display time before the sign is allowed to look up the CPN route table and update its suggested direction (Δt_R, Reactive).

4.1.1 Reactive

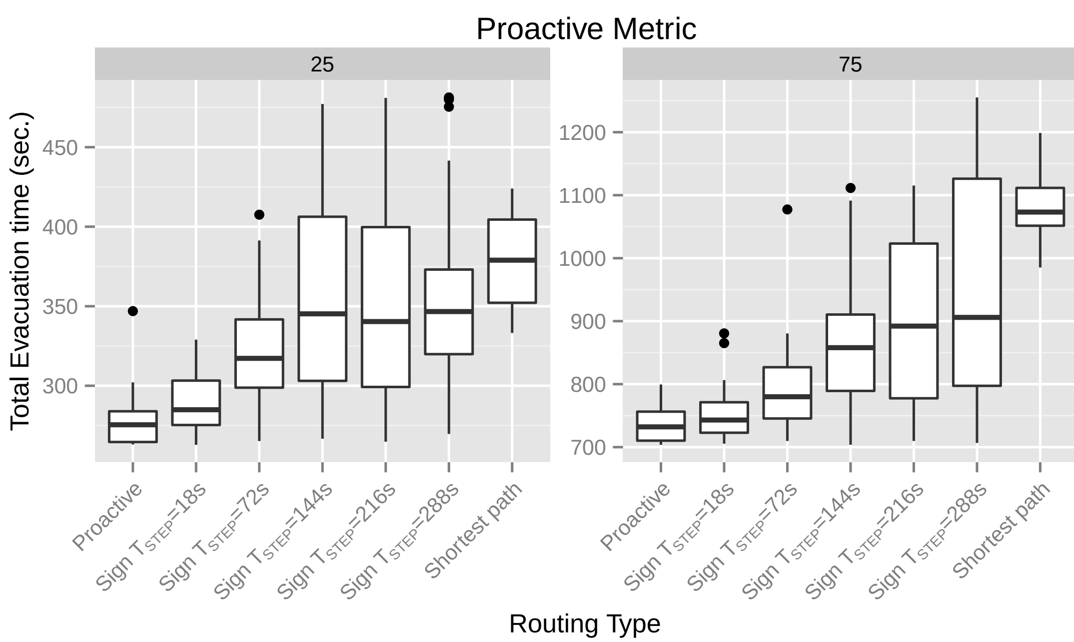
Figure 4a shows the evacuation times for the Reactive system using dynamic signs. We can see that signs which update most frequently ($\Delta t_R \geq$10 sec.) do not introduce any significant error, compared to the experiments which do not feature dynamic signs. To justify these results, let us recall that the Reactive routing approach tends to send all evacuees towards the shortest path at the beginning of the evacuation: as there is no congestion yet, this is the quickest path. As time goes, congestion gradually appears and the algorithm responds by diverting evacuees, and distributes them on longer but less-congested paths. If signs perform updates frequently, they can closely reflect the variations in CPN's optimal routes and introduce little error. Furthermore, switching between directions at a fast rate theoretically affords the system a very fine control over the evacuees, to a point where they can be routed on a one-by-one basis.
These advantages dissipate as the period between two consecutive updates increase. Once this period reaches the order of magnitude of the entire evacuation, the dynamic signs do not have time to perform enough updates to effectively balance traffic, and the system's performance decreases. As the update interval is further increased, eventually, the first update only occurs after all evacuees have left the building: in this case, the system is effectively static, and equivalent to routing evacuees down the initial shortest-path route. This explains why the results tend to converge towards those of the shortest-path approach, as the update period increases.
We also notice that, for a given update rate, the performance also depends on the evacuee headcount: as the number of evacuees decreases, evacuations become shorter. As a result, the the signs perform less updates, which limits their ability to perform load-balancing. This explains why performance degradations are observed from $\Delta t_R \leq$ 20 sec. for 25 evacuees, but only from 160 sec. for 75 evacuees.

4.1.2 Proactive

We recall that the Proactive approach issues a flow-optimal set of route assignments at the beginning of the evacuation. The sign-driving algorithm then processes these routes and outputs a schedule of next-hop directions for each sign to display. Unlike the Reactive approach, where routing decisions and sign orientation are tightly coupled, this two-tier operation allows us to isolate the bias introduced by the dynamic signs. We decide to focus our analysis on the most critical area of the building: the two staircases which lead to the first floor, as they are the building's main bottlenecks. We compare the number of evacuees that the routing algorithm intends to send on each staircase to the value observed at the end of the simulation: the difference gives us a measure of the error introduced by the signs. Figure 5 shows the empirical Probability Density Function (PDF) of this error, expressed as percentage points. For instance, if the routing algorithm's output resulted in a 50-50% assignment to each staircase, but that a 60-40% assignment was observed at the end of the evacuation, we consider the bias contributed by signs as being 10 percentage points. The PDF shows that low time-steps tend to reduce the error introduced by signs: as the system operates at a higher temporal resolution, the signs can coordinate their display more accurately with the scheduled arrival of individuals to each node. As the time-step duration increases, the PDF flattens, which indicates the dynamic signs introduce a larger bias. Figure 4b shows the result of the simulation in terms of building evacuation times. Much like the Reactive system, reducing the



(a) Reactive routing with dynamic signs



(b) Proactive routing with dynamic signs

Figure 4: Evacuation times for two sample evacuee headcounts (25 and 75). We also recall the results of the original algorithm (far left), and the shortest-path routing approach (far right)

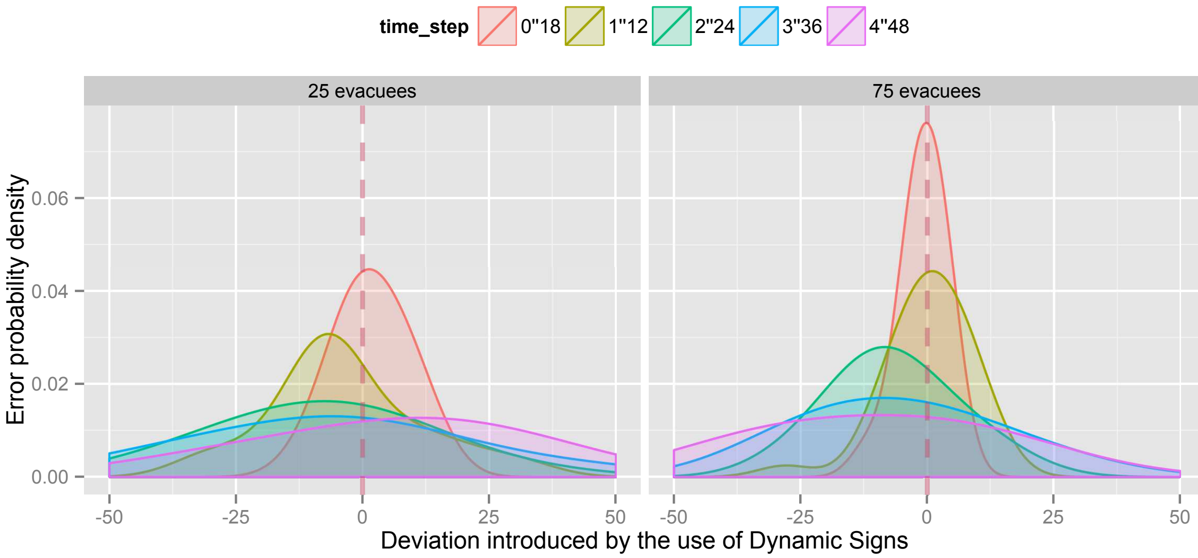


Figure 5: Empirical Probability Density Function of the error introduced by the dynamic signs, using the proactive approach.

time-step duration lets the signs alternate between directions more frequently: the sign effectively steers smaller groups of evacuees and is better able to implement the flow-optimal route assignments produced by the routing algorithm. Like the Reactive system, the time-step duration must be kept at least one order of magnitude below the building evacuation time: this ensures the signs are able to perform enough time-steps to effectively distribute evacuees on all available paths.

4.2 Discussion

In both cases, we have seen that allowing the dynamic signs to switch between directions more frequently improves the system's performance, as it allows the system to control smaller groups of evacuees, and closely implement the instructions of the routing component. This experiment has been conducted under the assumption that evacuees systematically follow the directions displayed on the signs. However, evacuees may be confused if they see a dynamic sign "flickering", i.e. displaying a rapid sequence of directions while they walk past. This may lead them to distrust the system, and will reduce evacuee compliance. Clearly, there is an optimisation problem: the signs must switch directions regularly to effectively disseminate evacuees on the different paths available, but switching too often will cause confusion amongst them. Reaching a definitive conclusion is difficult as we are not aware of any research on the compliance of evacuees guided with dynamic signs. We therefore recommend a conservative approach, which is to select the lowest possible update frequency, where performance remains acceptable.
We have also seen that, as the number of evacuee decreases the signs must switch between directions more and more often to maintain adequate performance. We conclude that the system can only be practical and effective if there is a minimum number of evacuees in the building. Indeed, our system reduces evacuation times by managing congestion, yet if there are not enough evacuees in the building to cause any congestion, our proposed congestion-aware evacuee guidance system becomes somewhat irrelevant.

5 Conclusion

In this paper, we have studied the routing and information display components of an evacuee assistance system for emergency building evacuations. The system aims to reduce evacuation times by making optimal use of every path available in the building through load-balancing. We have chosen the CPN routing algorithm to search for paths in the building graph, as its distributed and decentralised operation creates a robust system. We have used CPN with two routing metrics, which operate in a "reactive" and "proactive" manner. The Reactive approach requires a constant feed of congestion measurements throughout the evacuation, and is prone to oscillations. On the other hand, the Proactive metric forecasts the congestion which arises from each route assignment: since congestion is accounted for, the routes are stable and no further measurements nor route corrections are required. However, the accuracy of the forecast – and the routing algorithm's performance – depend on the accuracy of the evacuee mobility model. The Reactive approach also has the advantage of being able to adapt to unforseen events, which the Proactive approach cannot do without recomputing its congestion forecast. We have suggested a hybrid approach based on the Kalman filter to create a stable yet responsive system.

In the second part of this paper, we have considered means to inform evacuees of the optimal paths issued by the routing component using dynamic exit signs. While this can be readily implemented with the Reactive approach, the Proactive approach requires a specific algorithm to generate a schedule of directions to point at for each sign. In both cases, we did not add any sensory requirements beyond those required for the routing component. We have found that the performance depends on how often the signs are able to switch between directions: switching at a faster rate theoretically improves performance, but there is however a limit where evacuees will distrust the dynamic sign's information if its display switches between directions too frequently: this is an optimisation problem. The lack of studies in the response of evacuees to dynamic signs only allows us to make a conservative recommendation on optimal settings.

The next step in our research will be to conduct a sensitivity analysis on the system, in particular, to inaccurate localisation or congestion measurements, or to inaccurate estimation of the evacuee's walking speed (in the case of the Proactive approach). In the long term, we aim to develop the remaining components of our system, in particular, the sensing components and the network infrastructure.

References

[1] Matthew Barnes, Hugh Leather, and DK Arvind. Emergency evacuation using wireless sensor networks. In *Local Computer Networks, 32nd IEEE Conference on (LCN 2007)*, pages 851–857. IEEE, 2007.

[2] Huibo Bi and Erol Gelenbe. Routing diverse evacuees with cognitive packets. In *Proceedings IEEE PerCom 2014, Fourth International PerNEM Workshop on Pervasive Networks for Emergency Management*. IEEE Xplore, March 2014.

[3] Gen-Huey Chen and Yung-Chen Hung. On the quickest path problem. *Information Processing Letters*, 46(3):125 – 128, 1993.

[4] Wen-Tsuen Chen, Po-Yu Chen, Cheng-Han Wu, and Chi-Fu Huang. A load-balanced guiding navigation protocol in wireless sensor networks. In *Global Telecommunications Conference (IEEE GLOBECOM 2008)*, pages 1–6. IEEE, 2008.

[5] P. Derler, E.A. Lee, and A.-S. Vincentelli. Modeling cyber physical systems. *Proceedings of the IEEE*, 100(1):13–28, 2012.

[6] A. Desmet and E. Gelenbe. Graph and analytical models for emergency evacuation. *Future Internet*, 5(1):46–55, 2013.

[7] A. Desmet and E. Gelenbe. Reactive and proactive congestion management for emergency building evacuation. In *38th Annual IEEE Conference on Local Computer Networks (LCN'13)*, Sydney, Australia, October 2013. Accepted for publication.

[8] Nikolaos Dimakis, Avgoustinos Filippoupolitis, and Erol Gelenbe. Distributed building evacuation simulator for smart emergency management. *The Computer Journal*, 53(9):1384–1400, 2010.

[9] Peter Elias, Amiel Feinstein, and Claude Shannon. A note on the maximum flow through a network. *Information Theory, IRE Transactions on*, 2(4):117–119, 1956.

[10] Avgoustinos Filippoupolitis and Erol Gelenbe. A distributed decision support system for building evacuation. In *Human System Interactions, 2009. HSI'09. 2nd Conference on*, pages 323–330. IEEE, 2009.

[11] Avgoustinos Filippoupolitis, Laurence Hey, Georgios Loukas, Erol Gelenbe, and Stelios Timotheou. Emergency response simulation using wireless sensor networks. In *Proceedings of the 1st International Conference on Ambient Media and Systems*, Ambi-Sys '08, pages 21:1–21:7, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[12] Avgoustinos Filippoupolitis, Laurence Hey, Georgios Loukas, Erol Gelenbe, and Stelios Timotheou. Emergency response simulation using wireless sensor networks. In *Proceedings of the 1st international conference on Ambient media and systems*, page 21. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[13] Avgoustinos Filippoupolitis, Georgios Loukas, Stelios Timotheou, Nikolaos Dimakis, and Erol Gelenbe. Emergency response systems for disaster management in buildings. In *Proceedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management, Bucharest, Romania*, pages 1–14, 2009.

[14] Lester R Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404, 1956.

[15] Richard L Francis. A 'uniformity principle' for evacuation route allocation. *Journal of Research of National Bureau of Standards*, 86:509–513, 1981.

[16] Erol Gelenbe. Random neural networks with negative and positive signals and product form solution. *Neural computation*, 1(4):502–510, 1989.

[17] Erol Gelenbe. Learning in the recurrent random neural network. *Neural Computation*, 5(1):154–164, 1993.

[18] Erol Gelenbe. Cognitive packet network. US Patent #6,804,201, 10 2004.

[19] Erol Gelenbe and Michael Gellman. Oscillations in a bio-inspired routing algorithm. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1–7. IEEE, 2007.

[20] Erol Gelenbe and Qing Han. Near-optimal emergency evacuation. In *Proceedings IEEE PerCom 2014, Fourth International PerNEM Workshop on Pervasive Networks for Emergency Management*. IEEE Xplore, March 2014.

[21] Erol Gelenbe, Esin Seref, and Zhiguang Xu. Simulation with learning agents. *Proceedings of the IEEE*, 89(2):148–157, 2001.

[22] Erol Gelenbe and Fang-Jing Wu. Large scale simulation for human evacuation and rescue. *Computers & Mathematics with Applications*, 64(12):3869 – 3880, 2012. Theory and Practice of Stochastic Modeling.

[23] Erol Gelenbe and Fang-Jing Wu. Future research on cyber-physical emergency management systems. *Future Internet*, 5(3):336–354, 2013.

[24] Erol Gelenbe, Zhiguang Xu, and Esin Seref. Cognitive packet networks. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, pages 47–54. IEEE, 1999.

[25] G. Gorbil and E. Gelenbe. Disruption tolerant communications for large scale emergency evacuation. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 540–546, March 2013.

[26] Gokce Gorbil and Erol Gelenbe. Resilience and security of opportunistic communications for emergency evacuation. In *Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, pages 115–124. ACM, 2012.

[27] Gokce Gorbil and Erol Gelenbe. Resilient emergency evacuation using opportunistic communications. In *Computer and Information Sciences III*, pages 249–257. Springer London, 2013.

[28] H. W. Hamacher and S. Tufekci. On the use of lexicographic min cost flows in evacuation modeling. *Naval Research Logistics (NRL)*, 34(4):487–503, 1987.

[29] L. Han, F. Yuan, and T. Urbanik. What is an effective evacuation operation? *Journal of Urban Planning and Development*, 133(1):3–8, 2007.

[30] Dirk Helbing, Illes Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.

[31] Bruce Hoppe and Eva Tardos. The quickest transshipment problem. *Mathematics of Operations Research*, 25(1):pp. 36–62, 2000.

[32] John J. Jarvis and H. Donald Ratliff. Note – some equivalent objectives for dynamic network flow problems. *Management Science*, 28(1):106–109, 1982.

[33] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[34] Sangho Kim, Betsy George, and Shashi Shekhar. Evacuation route planning: Scalable heuristics. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS '07, pages 20:1–20:8, New York, NY, USA, 2007. ACM.

[35] Sangho Kim, Betsy George, and Shashi Shekhar. Evacuation route planning: scalable heuristics. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, page 20. ACM, 2007.

[36] SM Lo, Hui-Chi Huang, Peng Wang, and KK Yuen. A game theory based exit selection model for evacuation. *Fire Safety Journal*, 41(5):364–369, 2006.

[37] Qingsong Lu, Betsy George, and Shashi Shekhar. Capacity constrained routing algorithms for evacuation planning: A summary of results. In *Advances in Spatial and Temporal Databases*, volume 3633 of *Lecture Notes in Computer Science*, pages 291–307. Springer Berlin Heidelberg, 2005.

[38] Roya Olyazadeh. Evaluating dynamic signage for emergency evacuation using an immersive video environment. Master's thesis, Instituto Superior de Estatstica e Gesto de Informao (ISEGI), 2013.

[39] Georgia Sakellari and Erol Gelenbe. Demonstrating cognitive packet network resilience to worm attacks. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 636–638. ACM, 2010.

[40] T.J Shields and K.E Boyce. A study of evacuation from large retail stores. *Fire Safety Journal*, 35(1):25 – 49, 2000.

[41] John H Sorensen. When shall we leave? factors affecting the timing of evacuation departures. *International Journal of Mass Emergencies and Disasters*, 9(2):153–165, 1991.

[42] Yu-Chee Tseng, Meng-Shiuan Pan, and Yuen-Yung Tsai. Wireless sensor networks for emergency navigation. *Computer*, 39(7):55–62, 2006.

[43] Armin Veichtlbauer and Thomas Pfeiffenberger. Dynamic evacuation guidance as safety critical application in building automation. In *Critical Information Infrastructure Security*, volume 6983 of *Lecture Notes in Computer Science*, pages 58–69. Springer Berlin Heidelberg, 2013.