Future Internet **2013**, *5*, 46-55; doi:10.3390/fi5010046

OPEN ACCESS

future internet

ISSN 1999-5903

www.mdpi.com/journal/futureinternet

Article

Graph and Analytical Models for Emergency Evacuation

Antoine Desmet * **and Erol Gelenbe**

Intelligent Systems and Networks, Department of Electrical and Electronic Engineering, Imperial College, London SW7 2BT, UK; E-Mail: e.gelenbe@imperial.ac.uk

* Author to whom correspondence should be addressed; E-Mail: a.desmet10@imperial.ac.uk.

Received: 31 December 2012; in revised form: 31 January 2013 / Accepted: 7 February 2013 / Published: 21 February 2013

Abstract: Cyber-Physical-Human Systems (CPHS) combine sensing, communication and control to obtain desirable outcomes in physical environments for human beings, such as buildings or vehicles. A particularly important application area is emergency management. While recent work on the design and optimisation of emergency management schemes has relied essentially on discrete event simulation, which is challenged by the substantial amount of programming or reprogramming of the simulation tools and by the scalability and the computing time needed to obtain useful performance estimates, this paper proposes an approach that offers fast estimates based on graph models and probability models. We show that graph models can offer insight into the critical areas in an emergency evacuation and that they can suggest locations where sensor systems are particularly important and may require hardening. On the other hand, we also show that analytical models based on queueing theory can provide useful estimates of evacuation times and for routing optimisation. The results are illustrated with regard to the evacuation of a real-life building.

Keywords: cyber-physical systems; emergency management; queueing models; graph models

1. Introduction

Cyber-Physical-Human Systems (CPHS) offer interesting perspectives for the comfort, health and safety of users [1,2] with applications in building energy efficiency [3,4] and emergency management [5]. While there has been extensive research regarding guidance systems, routing algorithms and efficient task assignment algorithms [6] for emergency evacuation applications, the operation of such systems in degraded conditions often associated with emergencies is generally overlooked. Recent work [7] shows that while a majority of sensors provide information of limited value, only a few strategically located sensors will critically affect the CPHS' performance and the outcome of an evacuation. While relevant analytical methods based on linear control exist to study systems in degraded conditions, in particular reliability and robustness [8,9], redundancy [10,11] and fault-tree analysis [12], these methods do not handle the complex and emergent aspects of a CPHS.

On the other hand, much work in this area relies on experiments with purpose-built simulators such as the Distributed Building Evacuation Simulator (DBES) [13,14], which provides an approach that is similar in many aspects to traditional tactical modelling and simulation in the military domain [15]. The work on the design and optimization of emergency management schemes based on discrete event simulation [16] is faced by the substantial amount of programming that is often needed when existing simulation tools are applied to a new problem, and by the scalability and large computing time needed to obtain useful performance estimates in realistic environments. Thus, this paper proposes a complementary approach that offers fast estimates based on queueing theoretic analytical models [17–19] as well as graph models and algorithms.

The graph representations themselves, and in particular those used in this paper for our examples, will typically represent the space in a specific building. Since the space within modern buildings, or other built spaces such as areas within cities or towns, are typically represented with databases and graphical systems [20], going from the detailed building representation to a graph representation, which includes accurate distances and representative traversal times, can be carried out in an automatic and accurate manner. This is not too dissimilar (but generally simpler) than the construction of "terrain models" in simulation systems used in the military domain [21] where natural terrain features must also be included.

The use of probability and analytical models in this context is not new [22–25], but for a variety of reasons including the large computational costs of the early closed network (finite population) models, such models appear to have been abandoned in this area for the last decade or more. The approach we take here uses open models whose computational needs are very low and where useful formulas, for instance for average traverse times and throughput, can be obtained very rapidly [17,19]. Such probability models are particularly useful in situations where there is a substantial amount of uncertainty [26], while graph models are useful when there is a substantial amount of topological structure such as the rooms, corridors, staircases in a building, or when area-related information will remain fixed for long periods of time.

Combining the probability models for human movement and congestion with graph theory for locations offers a way forward to deal efficiently with the large and realistic representations that are needed in CPHS. Our work suggests that analytical models based on queueing theory can provide fast estimates for the location of points of congestion, for estimating the sensitivity of outcomes to the presence of hazards in particular areas, and for routing optimization.

To illustrate the use of these approaches, we examine an emergency evacuation in the first three floors of a real-life building. The graph models allow us to identify sensors that are particularly critical and that would benefit from being hardened. The probability model predictions are useful in estimating the evacuation times and points of congestions and provide a basis to compare results with simulations.

2. Graph Based Critical Sensor Assessment

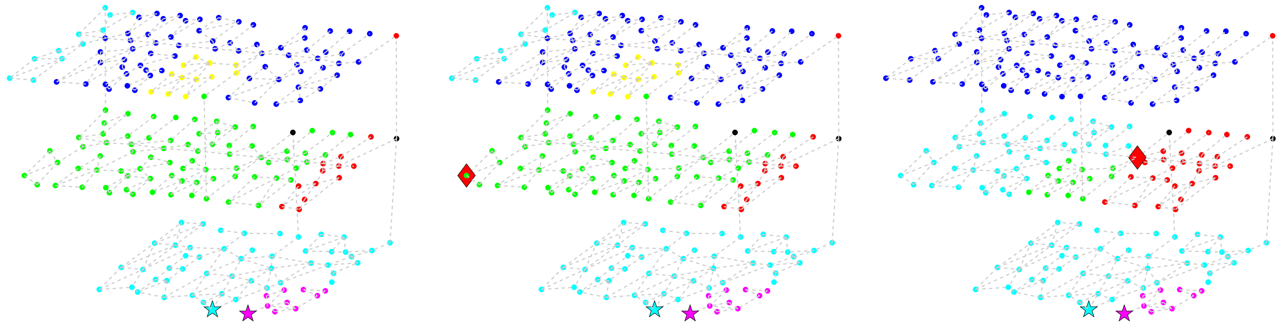
In this section, we show how graph theory can be used to offer a rapid assessment of the critical locations for placing sensors that will provide the high-value information during an emergency evacuation. The two algorithms that we propose rank locations by degree of criticality and use only *a priori* knowledge solely based on a graph theoretic analysis of the connectivity in a building.

Note that consistently with our objective of using realistic graph representations of existing buildings, we test both algorithms using a graph modelling the three first floors of the EEE building on the Imperial College London campus. Each floor in this building has a surface area of approximately 1000 m^2.

Algorithm 1—Identifying the Most Disruptive Fire Outbreak Locations

This algorithm is inspired by the definition of *critical nodes* as nodes that , if removed, will cause the greatest increase in the shortest distance between two other nodes [27]. Our approach differs from this standard definition and actually ranks all nodes according to how many shortest exit paths from nodes occupied by evacuees become disconnected when the examined node becomes inaccessible. The ranking metric is linked to the "disruptiveness" of a location, since the evacuee's instinctive behaviour is to follow the shortest path to the nearest exit. For instance, if a fire breaks out at a highly ranked node, then more evacuees would need re-routing and advice. The algorithm first creates a reference map [Figure 1(a)] that records the shortest exit path from every node. Then it iteratively removes each node from the building graph—to simulate an area blocked by fire—and finds a new set of shortest paths and compares them with the initial one [Figure 1(b), 1(c)]. Each change in the updated path corresponds to a disrupted shortest path and the corresponding ranking metric is incremented. This algorithm is modified so that only nodes whose elimination actually changes the exit point or staircase are considered.

Figure 1. The colours attributed to each node represent the optimal floor exit, e.g., nodes coloured in blue will see their occupants exit through the N-E stairs. The red Diamond shows the fire location, the stars show the building exits. (**a**) shows the reference map when there is no fire; (**b**) illustrates a fire outbreak at a non-critical area—causing very little changes compared with the reference map; (**c**) illustrates a fire outbreak at a critical area—highlighted by the profound changes compared with the reference map.

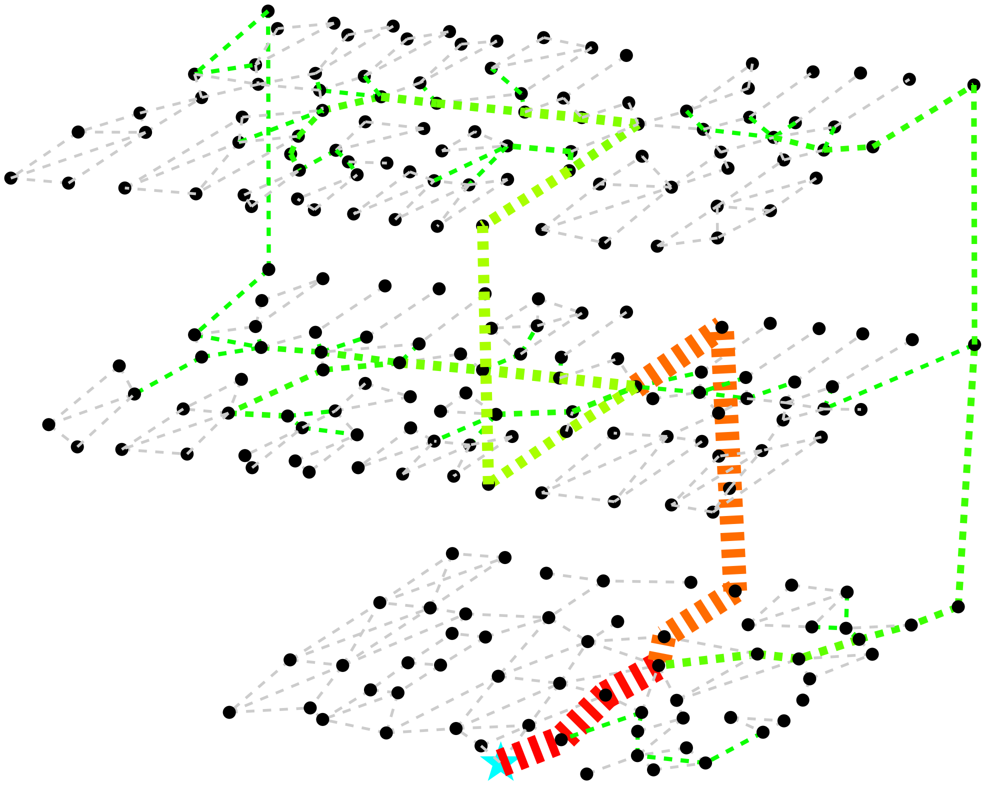


(a) No fire—reference map (b) Fire in a non-critical area (c) Fire in a critical area

Algorithm 2—Determining the Busiest Nodes During an Evacuation

Another definition of a critical node would consider the most used or congested locations during an evacuation as being the most critical. This algorithm iterates through every possible departure point in the graph and increments the ranking metric of each location visited along the shortest exit path. Figure 2 is a graphical representation of its output.

Figure 2. Graphical representation of the output of Algorithm 2, where line thickness increases with visit count. The cyan star on the ground floor marks the exit. Results shown for a subset of all building locations.



Algorithm 1's top-ranking nodes are scattered across the building, covering corridor intersections or "bottlenecks" such as staircases, and locations near the exits also achieve a high rank. Algorithm 2's top-ranking nodes are mostly found in the vicinity of exits since the collection of all egress paths form trees rooted at each exit, and the most visited locations are at the trunk of each tree. Overall, the top five locations selected by each algorithm are almost completely different; however, as the sample size is increased, the degree of overlap between sets increases: there is only 1 location shared between both sets for the top 5 most critical nodes, whereas the top 10 and top 20 overlap by respectively 4 and 14 locations.

2.1. Improving the Critical Sensors

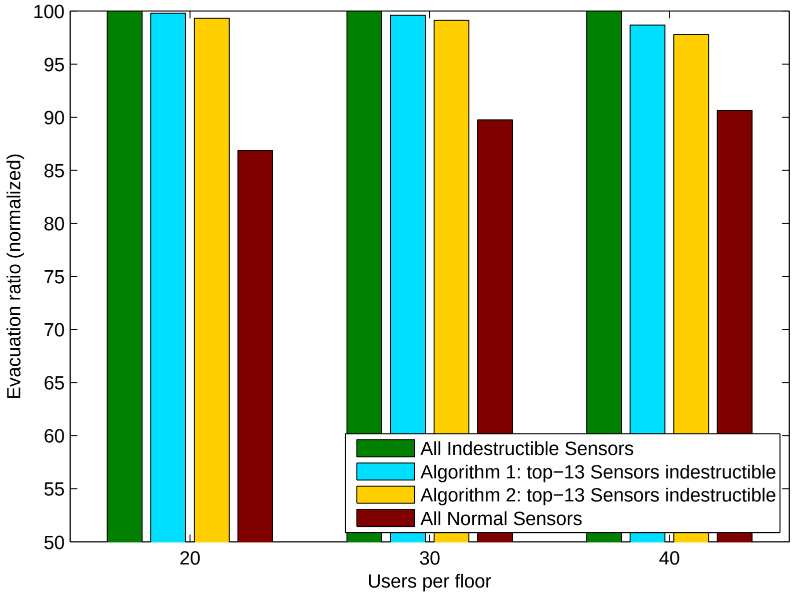
In fact, as we will see now, the algorithms presented in this paper identify (through their locations) the most critical sensors whose information is instrumental in devising the best evacuation paths. Therefore the lifetime of such sensors in presence of an emergency (such as a fire) should be "hardened", e.g., by upgrading them with heat-resistant components, or adding redundancy.

We will illustrate this by showing via simulation how the proportion of successful evacuees increases when critical sensors are hardened. In particular we identify the sensors to be hardened using the top-ranked thirteen locations (out of 240) identified by Algorithm 1 and 2, and the results are given in Figure 3. Results are normalized against a best-case scenario where all sensors are indestructible (green bars). A worst-case scenario where all nodes fail under fire (red bars) is also shown for comparative purposes. Both algorithms appear to perform well with a slight advantage for Algorithm 2. This can be explained by the fact that only seven locations out of thirteen differ amongst the two sets. A detailed analysis of the simulations also reveals that each algorithm may be best suited for particular types of graphs:

- **Algorithm 1** is most effective in intricate graphs featuring staircases, corridors and partitioned space, where an evacuation plan must be decided early based on the designation of strategic areas.
- **Algorithm 2** is best suited for open spaces, where bypassing the fire is generally trivial and the critical decisions relate to the availability of exits and how to approach them.

The building graph used in this simulation happens to be a blend of both types of spaces, which may explain why neither of the algorithms has a clear advantage.

Figure 3. Graph-based algorithms' result comparison.



3. Queueing Analysis

The queueing analysis techniques we suggest are based on open models whose use is computationally very fast due to the much simpler structure of their normalizing constants, whereas earlier work focused on closed queueing network models [22]. The closed model approach is based on a representation where an evacuation indefinitely repeats itself at random under identical stochastic conditions, so that the evacuees are "virtually" re-entering the building after having been evacuated.

On the other hand, the open model approach that we use is useful for situations where the overall evacuation time is very long compared with the individual evacuation steps that evacuees are taking, and during this time evacuees are progressively leaving their offices or other parts of the area being considered to create the "arrival flow" of people into the evacuation process.

We associate with each location L from which evacuees may originate (e.g., a room in an office building) an "arrival rate" λ_L of evacuees. Here L does not represent a segment of a corridor or of a passage area (e.g., area in front of a lift) where evacuees arrive from other locations; L is just a specific room that they will leave during the evacuation.

Specifically let $n(L)$ be the number of people who are present at location L before the evacuation begins. Note that $n(L)$ itself may be random because we actually do not always know how many people are present at each location in the building. Then if $W(L,E)$ is the expected "best case" evacuation time from location L based on no congestion and no impediments on the known normal evacuation path, we will take $\lambda_L \approx \frac{n(L)}{W(L,E)}$. Note that this will provide a larger or pessimistic value of λ_L because in fact the real evacuation time will be larger than the best case value.

Consider any path from some starting in location L to some location i on the evacuation path from L, and denote it by $\pi(L,i)$; note that there may be many such paths. Let us denote by $p(\pi(L,i))$ the probability that the path $\pi(L,i)$ is taken by an evacuee starting at L. Then the total arrival rate of evacuees into the location i will be:

$$\Lambda_i = \sum_{L} \sum_{All\ \pi(L,i)} p(\pi(L,i))\lambda_L \tag{1}$$

3.1. Analytical Results

Each significant section (i,j), such as a section of a corridor between say two adjacent doors, where i and j are adjacent locations, has an associated average traversal delay from i to j that we call $\frac{1}{\mu_{ij}}$, whose value can change with the conditions in the building. For any edge, the average traversal time will then include the congestion and be given by [17]:

$$W_{ij} = \frac{\mu_{ij}^{-1}}{1-\rho_{ij}} \tag{2}$$

where

$$\rho_{ij} = \frac{\Lambda_i . p_{ij}}{\mu_{ij}} \tag{3}$$

and p_{ij} is the fraction of evacuees that arrive at i and then take the section (i,j) towards their goal destination s:

$$p_{ij} = \sum_{\pi(i,s)} p(\pi(i,s)) \qquad \pi(i,s) = (i,j,\pi(j,s)) \tag{4}$$

The average traversal time of a path, and the average evacuation time to any of the exit points will then be obtained in a straightforward manner by summing the appropriate traversal delays.

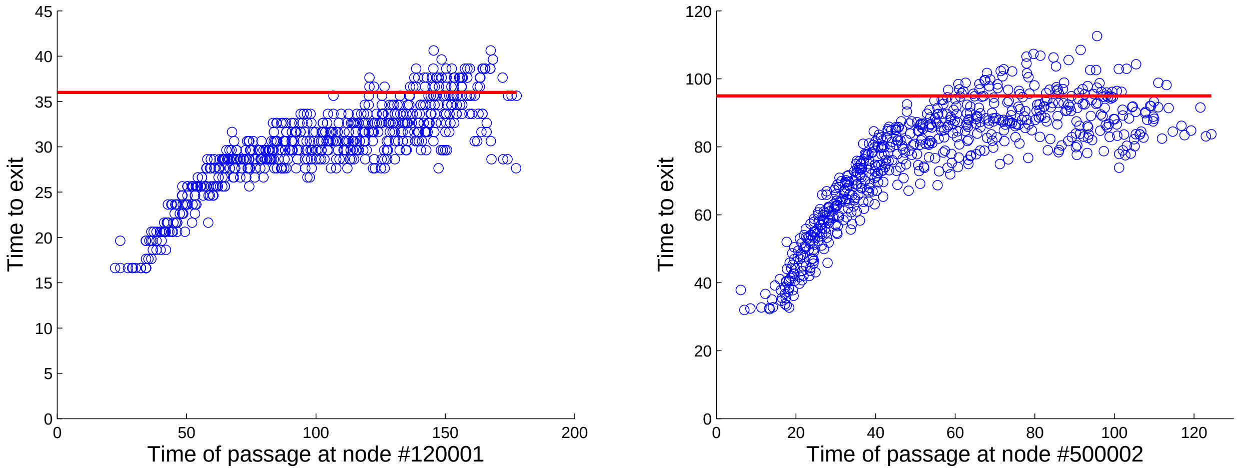
The Expression (2) is a steady-state value, which is equivalent to assuming that the overall duration of an evacuation is much longer than the time it takes to traverse any of the individual sections of the graph model of the building. Furthermore this formula is based on a Jackson network representation [17] of steady-state, which assumes Poisson arrivals to the originating locations, exponentially distributed traversal delays, and that any directional section (i, j) from i to j can only have at most one evacuee at a time.

In order to compare the queuing analysis with the simulation results, we use identical μ_{ij} rates and the same building graph. The set of L "source" nodes corresponds to the offices where users tend to reside in normal conditions (*i.e.*, when there is no emergency) in the simulated environment. The $\pi(L, i)$ values are chosen so that the evacuees are always being directed along the shortest paths to the evacuation points.

Figure 4 shows the results of the queueing analysis and of the simulation. The scatter plot shows the time taken to reach the nearest exit (Y axis) based on the time at which a node on the first-floor lobby [Figure 4(a)] or at a second-floor hallway intersection [Figure 4(b)] is visited. The plot shows the output of 10 iterations of simulations involving 120 building users (which is the average occupancy of the real-life building modelled), with initial locations that have been randomized at each run.

Both scatter plots clearly show a gradual increase in congestion at the beginning of the evacuation, until the network reaches its saturation point and the evacuation times become somewhat steady. The red line shows the corresponding steady-state node-to-exit delay that is predicted by the queueing model.

Figure 4. Time needed to reach the exit based on the time at which the node of interest was visited. The red line shows the steady-state value predicted by the queueing network analysis.



(a) First-floor lobby node

(b) Second-floor hallway intersection

In the examples of Figure 4 we see that the queueing analysis seems to over-estimate (worst case) the average traversal times when compared with the simulation results. This is not surprising because the way we compute the arrival rates in the queueing model can yield high worst case values. Furthermore the simulation assumes *deterministic* traversal times and it is well known that service models with deterministic service times will underestimate the average response or traversal times [18,19].

The deterministic section traversal times in the simulation are obviously unrealistic because different people will actually have different walking speeds, and furthermore the panic and other (e.g., fire, smoke) conditions in an emergency will render things far more random. Thus paradoxically, the analytical model may actually provide a more accurate rendering of what may be really happening during the latter parts (worst case) of an emergency when congestion starts to occur and the behaviour of the various agents is far more random. Thus we suggest that such analytical models are particularly useful for worst case estimation, which is very important in the design and evaluation of emergency procedures.

As an additional benefit of the queueing analysis, we have also considered ranking the criticality of the nodes based on their utilization (ρ) rates and compared this list with the results of the graph-based methods presented in the previous section. It appears that the same nodes appear in the top 10% ranking list, however the precise order is different across all three (two graph-based and one queueing-based) methods.

4. Conclusions

In this paper, we suggest the use of analytical techniques based on queueing network models and graph theory in order to improve the planning of building evacuations. The graph theory techniques that are proposed can be used to identify and then selectively reinforce sensors covering the most critical areas, and identify relatively few locations where hardened sensors can allow us to approach near-optimal evacuation performance. To this effect we have proposed two algorithms, and both have been shown to provide the type of performance improvements that were hoped.

In future work, we propose to examine the use of adaptive evacuee routing techniques [28,29] that can help reach better outcomes, together with probabilistic modelling techniques [30]. Multiple class approaches [31,32] will also be needed to represent and deal with the distinct mobility needs of different categories of users, such as emergency personnel, normal evacuees, or people who are hampered in their mobility or who use wheelchairs. In many such cases, one would also need to develop augmented reality simulations [21] that can offer a more accurate representation of obstacles and of realistic evacuation conditions.

We also plan to conduct additional simulations and experimentation with such analytical techniques in the context of different types of building topologies, with the objective to:

- Compare the performance of the graph-based algorithms in either flat open-space areas or intricate multi-storey buildings to validate the hypothesis that the algorithms presented are better suited to some types of graphs.
- Devise more complex evacuation scenarios so that the performance gap between *optimal* and *realistic* scenarios is widened. The relative ease of evacuating the building resulted in rather high performance in the worst-case scenario, meaning that there is only limited scope for improvement.

Future work on graph-based algorithms will also take into consideration a realistic distribution of users in the building, and let densely-populated nodes have a greater influence on the metrics. However, the distribution of users will rapidly change throughout the simulation and it is unclear whether basing all calculations on the density map at $t = 0$ will lead to sustained improvements. While providing a simple and effective method to identify critical devices in a fire-monitoring sensor network, the algorithms presented in this paper cannot handle cases featuring multiple and simultaneous fire outbreaks, nor do they account for fire expansion or evacuee movement and congestion. Further research will aim at developing extensions that address these issues.

References

1. Wong, J.; Li, H.; Wang, S. Intelligent building research: A review. *Autom. Constr.* **2005**, *14*, 143–159.
2. Gelenbe, E.; Wu, F.J. Large scale simulation for human evacuation and rescue. *Comput. Math. Appl.* **2012**, *64*, 3869–3880.
3. Schor, L.; Sommer, P.; Wattenhofer, R. Towards a zero-configuration wireless sensor network architecture for smart buildings. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, ACM: New York, NY, USA, 3 November 2009; pp. 31–36.
4. Agarwal, Y.; Balaji, B.; Gupta, R.; Lyles, J.; Wei, M.; Weng, T. Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, ACM: New York, NY, USA, 3–5 November 2010; pp. 1–6.
5. Snoonian, D. Smart buildings. *IEEE Spectr.* **2003**, *40*, 18–23.
6. Aguilar, J.; Gelenbe, E. Task assignment and transaction clustering heuristics for distributed systems. *Inf. Sci.* **1997**, *97*, 199–219.
7. Desmet, A.; Gelenbe, E. *Interoperating Infrastructures in Emergencies*; Gelenbe, E., Lent, R., Eds.; Springer: London, UK, 2012; pp. 123–130.
8. Hoblos, G.; Staroswiecki, M.; Aitouche, A. Optimal design of fault tolerant sensor networks. In *Proceedings of the 2000 IEEE International Conference on Control Applications*, Anchorage, AK, USA, 25–27 September 2000; pp. 467 –472.
9. Ali, Y.; Narasimhan, S. Sensor network design for maximizing reliability of linear processes. *AIChE J.* **1993**, *39*, 820–828.
10. Wang, F.; Ramamritham, K.; Stankovic, J. Determining redundancy levels for fault tolerant real-time systems. *IEEE Trans. Comput.* **1995**, *44*, 292–301.
11. Maquin, D.; Luong, M.; Ragot, J. Some ideas about the design of measurement system. In *Proceedings of European Control Conference ECC'95*, Rome, Italy, 5–8 September 1995; pp. 3178–3183.
12. Lee, W.S.; Grosh, D.L.; Tillman, F.A.; Lie, C.H. Fault tree analysis, methods, and applications 2013: A review. *IEEE Trans. Reliab.* **1985**, *R-34*, 194–203.
13. Dimakis, N.; Filippoupolitis, A.; Gelenbe, E. Distributed building evacuation simulator for smart emergency management. *Comput. J.* **2010**, *53*, 1384–1400.
14. Gorbil, G.; Gelenbe, E. Opportunistic communications for emergency support systems. *Procedia Comput. Sci.* **2011**, *5*, 39–47.
15. Gelenbe, E.; Hussain, K.; Kaptan, V. Simulating autonomous agents in augmented reality. *J. Syst. Softw.* **2005**, *74*, 255–268.
16. Filippoupolitis, A.; Gelenbe, E. A distributed decision support system for building evacuation. In *Proceedings of 2nd Conference on Human System Interactions HSI '09*, Catania, Italy, 21–23 May 2009; pp. 320–327.

17. Gelenbe, E.; Muntz, R.R. Probabilistic models of computer systems: Part I (exact results). *Acta Inf.* **1976**, *7*, 35–60.
18. Gelenbe, E.; Pujolle, G. *Introduction to Queueing Networks*; John Wiley: London, UK, 1998.
19. Gelenbe, E.; Mitrani, I. *Analysis and Synthesis of Computer Systems*; World Scientific: Singapore, 2010.
20. Gelenbe, E.; Hussain, K.; Kaptan, V. Simulating autonomous agents in augmented reality. *J. Syst. Softw.* **2005**, *74*, 255–268.
21. Kaptan, V.; Gelenbe, E. Fusing terrain and goals: Agent control in urban environments. In *Proceedings of Multisource Information Fusion: Architectures, Algorithms, and Applications 2006*, Kissimmee, FL, USA, 19–20 April 2006; Volume 6242, pp. 71–79.
22. Smith, J.M. State-dependent queueing models in emergency evacuation networks. *Transp. Res. Part B Methodol.* **1991**, *25*, 373–389.
23. Bakuli, D.L.; Smith, J.M. Resource allocation in state-dependent emergency evacuation networks. *Eur. J. Oper. Res.* **1996**, *89*, 543–555.
24. Hasofer, A.; Odigie, D. Stochastic modelling for occupant safety in a building Fire. *Fire Saf. J.* **2001**, *36*, 269–289.
25. Elms, D.; Buchanan, A.; Dusing, J. Modeling fire spread in buildings. *Fire Technol.* **1984**, *20*, 11–19.
26. Gelenbe, E.; Hébrail, G. A probability model of uncertainty in data bases. In *Proceedings of the Second International Conference on Data Engineering*, Los Angeles, CA, USA, 5–7 February 1986; pp. 328–333.
27. Nardelli, E.; Proietti, G.; Widmayer, P. Finding the most vital node of a shortest path. *Theor. Comput. Sci.* **2003**, *1*, 167–177.
28. Gelenbe, E. Sensible decisions based on QoS. *Comput. Manag. Sci.* **2003**, *1*, 1–14.
29. Gelenbe, E. Steps towards self-aware networks. *Commun. ACM* **2009**, *52*, 66–75.
30. Gelenbe, E.; Stafylopatis, A. Global behaviour of homogeneous random neural systems. *Appl. Math. Model.* **1991**, *15*, 534–541.
31. Atalay, V.; Gelenbe, E. Parallel algorithm for colour texture generation using the random neural network model. *Int. J. Pattern Recognit. Artif. Intell.* **1992**, *6*, 437–446.
32. Fourneau, J.M.; Gelenbe, E. Random neural networks with multiple classes of signals. *Neural Comput.* **1999**, *11*, 953–963.

© 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).