Identifying Critical Sub-Systems in the Simulation of Cyber-Physical Systems

Antoine Desmet and Erol Gelenbe
Department of Electrical and Electronic Engineering
Imperial College
London, United Kingdom
$\{$*a.desmet10, e.gelenbe*$\}$*@imperial.ac.uk*

***Abstract*—The great challenge in designing dependable Cyber-Physical Systems (CPS) is to structure them as robust and decoupled sub-systems, where individual sub-system failures will only have a limited impact on the overall CPS' performance. In this paper, we discuss a CPS performing as a building emergency evacuation assistant. The CPS that we consider in this paper, relies on a network of wired hazard sensors, computers and RF devices to provide the evacuees with personalized evacuation information, and it is used to model the effect of a fire, which not only poses a threat to the building's occupants but also damages the CPS' sensors and other systems. This paper introduces a graph theory-based generic method to identify and rank "critical" sensors, i.e. those whose information is most valuable. We validate the proposed approach through a simulation in which the *critical* sensors are "hardened", thereby extending their lifetime under fire. Our simulations show how the proposed approach which selects a few critical sensors that are to be hardened, can improve the outcome of the evacuation.**

***Keywords*-Cyber-physical systems, emergency navigation, pervasive computing, search and rescue systems.**

I. INTRODUCTION

The integration of Cyber-Physical Systems (CPS) to buildings offers interesting perspectives to improve the comfort, health and safety of users. A review of "Smart Building" applications [1] reveals two prominent research areas: building energy efficiency [2], [3], and safety or emergency management [4]. A survey of the literature [5] shows that extensive research has been conducted in the fields of evacuee routing algorithms and guidance systems for evacuees. However, performance under degraded conditions is often overlooked. We believe that the likely presence of hazards (fire, smoke, heat or water) in emergency evacuation scenarios makes hardware component failure virtually inevitable. Furthermore, reliance on multiple tightly-coupled sub-systems (sensors, networks, computational devices, actuators, etc.) makes CPS especially vulnerable to single component failure. The control systems literature provides relevant concepts to measure reliability and robustness, such as variable observability [6], [7], degree of redundancy [8], [9] and fault-tree analysis [10]. However, these approaches are mostly applicable to linear systems, or non-linear dynamical systems of relatively small size, and fail to address the great complexity, strong non-linearity and possible model breakdowns, and the emergent behavior CPS.

This paper studies the performance of a CPS affected by hardware component failure caused by a building fire. The featured CPS advises evacuees on optimal escape routes during a building emergency evacuation to maximize the survival rate. Previous research [11] highlights that the value of the information acquired by the sensors is very heterogeneous: while a majority of sensors provide information of limited value, a few sensors in strategic areas of the building can effectively condition the CPS' performance – and the evacuation outcome.

The dynamic management of such highly distributed and time-varying systems present great challenges, not least because we know that it is extremely difficult if not impossible to maintain a consistent view of the system as a whole [12], and interpreting probabilistic data is in itself challenging [13], while most decisions need to be taken in a distributed manner [14].

The following section introduces the scenario and simulation model. We then introduce two generic, graph-based algorithms which identify critical fire sensors in the context of emergency evacuation. A review of experimental results demonstrated the effectiveness of these algorithms, and conclusions are drawn.

II. SIMULATION MODEL AND SCENARIO

The Distributed Building Evacuation Simulator (DBES) [15] used in the following experiments is a purpose-built Discrete-Event Simulator (DES). The CPS model features a dense fire-monitoring sensor network which is subdivided into "zones". Each zone covers an area of the building, and all sensors within a zone report to the floor's central fire panel using a shared bus wire. Owing to this bus topology, all sensors in a zone are coupled, in terms of failure: any fault or short-circuit will affect the *entire* zone. The sensor model takes into account exposure to the fire hazard, so that a sensor is able to measure moderate fire intensities before eventually defaulting. The CPS also relies on an RF tracking component to localize RF communication devices carried by evacuees. At this stage, this component is modeled at a very high level and allows the CPS to determine on which floor and in which area of the building evacuees are located. We acknowledge that the performance of RF tracking system is impaired in extreme environments involving fire, smoke or

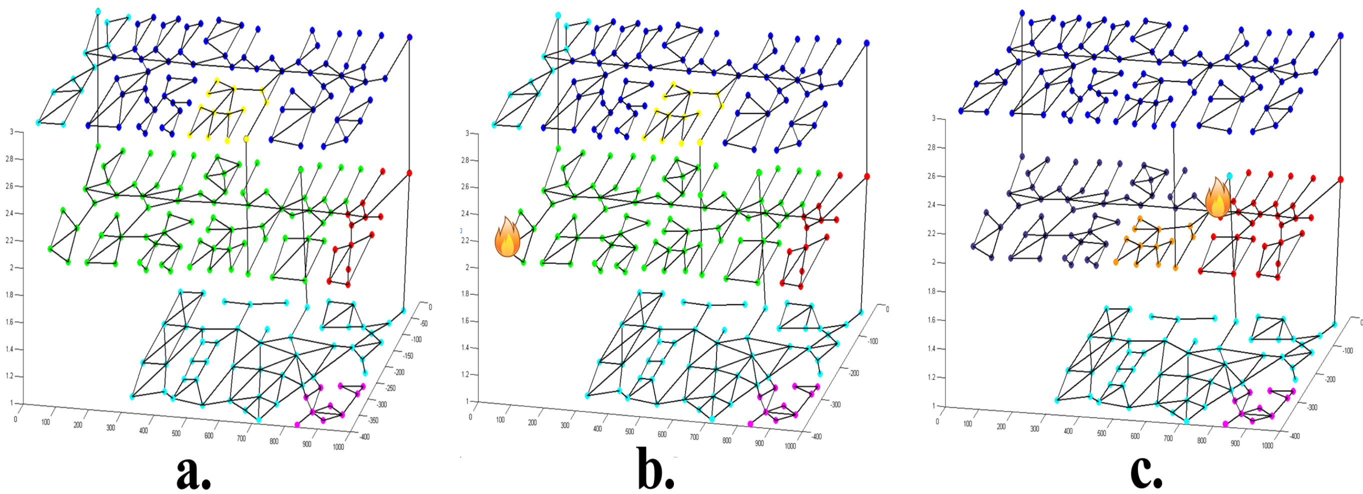

Figure 1. Graphical representation of the algorithms' output: nodes that have the same color will lead to the same floor exit. a). shows the reference map: no fire, b). shows a fire outbreak location which will not modify anyone's evacuation strategy, c). illustrates a fire outbreak location which will disrupt the normal evacuation patterns

heat, hence the accuracy is conservatively lower that state-of-the-art systems, which perform localization at room-level accuracy [16]. Finally, the software component of the CPS fuses the information of both sub-systems to recommend an individualized shortest egress path to each user, which is transmitted to the user's communication device. This path is calculated using the Dijkstra algorithm, from the user's measured location to the nearest exit. By inflating the apparent length of an edge affected by fire, the Dijkstra algorithm outputs fire free paths or, failing this, the safest possible path.

The CPS' decision-making component is affected by faults in the fire-sensing network which cut the supply of "fresh" information on fire spread and intensity. In such cases, the CPS uses the last reading acquired to calculate the recommended escape route. The result is that, as time goes by and the fire expands, the CPS' advice will become increasingly inaccurate. An alternative is to consider an entire fault-affected area as being unsafe, but given the relatively large size of each zone, this may be an overly conservative estimation which may ultimately prove to be detrimental to users. This option will nevertheless be evaluated in further studies.

Two simulation scenarios provide an upper and lower bound of the CPS' theoretical performance: an *optimal* scenario, where all sensors are fire-proof and a *realistic* scenario where every sensor rapidly fails when exposed to fire. The success metric of the CPS is the overall building evacuation ratio.

As a realistic example, a graph is created to represent a three-storey office building. The building has two exits on the ground floor, and two or three stairways connecting each floor. The ground floor mainly consists of an open lobby area, while the two upper floors feature clustered office spaces on both sides of a long corridor. Each simulation scenario is run with varying densities of building occupants, in this case $\{20, 30, 40\}$ occupants per floor.

III. Critical Sensor Assessment

In this section we will detail two algorithms which identify the most critical sensors, i.e. those which provide the high-value information for emergency building evacuation management. These algorithms must rank locations by degree of criticality, and use only *a priori* knowledge, i.e. be solely based on the analysis of the location's properties.

A. Algorithm 1 – Most disruptive fire outbreak locations

This algorithm is inspired from the definition *of most critical node* as that which, if removed, will cause the greatest increase in the shortest-path distance between two nodes [17]. Our approach is different and ranks nodes based on the number of shortest-paths in which they take part. Effectively, we count each departure point in the graph which sees its shortest path disconnected when the node becomes unavailable. The ranking metric is linked to the "disruptiveness" of a location, since the evacuee's instinctive behavior is to follow the shortest path to the nearest exit. The knowledge that a fire has broken out on a node which is part of several exit paths is extremely valuable information, since it indicates that many users will need re-routing and advice. The algorithm proceeds by initially creating a reference map by recording the shortest exit path from every node – without any fire. Following this, a second algorithm iteratively removes each node from the building graph – to simulate an area blocked by fire – and resolves a new set of shortest paths and compares them with the reference. Each change in the updated path corresponds to a shortest

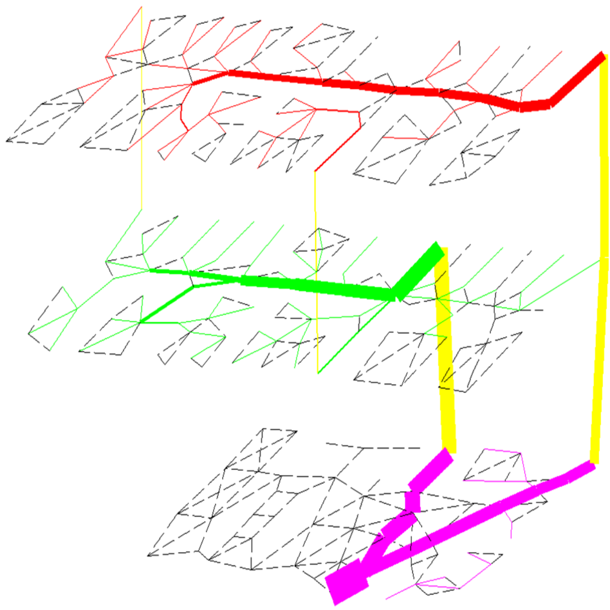

Figure 2. Graphical representation of the output of Algorithm 1, where line thickness increases with visit count. Results are shown for a subset of all building locations

path disrupted by the fire, and the corresponding ranking metric is incremented. The preliminary results indicate that this algorithm may be too sensitive to small differences in paths since it highlights countless negligible changes in paths – mostly pathes bypassing an obstacle using a neighboring node. Thus the resolution was lowered to only take into account variations in paths that correspond to major changes in the evacuation strategy: we therefore define a re-routing as an event where the path leaves the floors through a different exit or staircase. Figure 1 illustrates parts of the algorithm's computational process.

B. Algorithm 2 – Busiest Nodes during Evacuation

Another approach to the problem is to consider the busiest 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 the output of this algorithm.

C. Algorithm output analysis and comparison

The output of each algorithm is presented on Figure 3. Algorithm 1's top-ranked nodes are scattered across the building, covering corridor intersections or "bottlenecks" like staircases. Further down the ranking list, most locations correspond to areas near the exits. Algorithm 2's top-ranking nodes are mostly found in the vicinity of the exits. This is expected, since the collection of all egress paths forms trees rooted to each exit, therefore the most visited locations are on the trunk of each tree. Overall, the top 5 locations

	locations found in both sets
top 5	1
top 10	4
top 20	14

Table I
OVERLAP BETWEEN RESULTS OF BOTH ALGORITHMS

selected by each algorithm are almost completely different, however as the sample size is increased, the degree of overlap between sets increases, as shown in Table I.

IV. SOME SIMULATION EXPERIMENTS

Our hypothesis is that the algorithms presented in this paper identify the most critical sensors: those whose information is instrumental in devising optimal evacuation paths. Given this information, the lifetime of sensors under fire can be extended by upgrading to heat-resistant components and adding redundancy. We refer to this process as "hardening" the sensors. Hardening the essential sensors lets the CPS have access to the most vital information for a longer amount of time, which will result in an overall improvement in advice to evacuees and increase survival rate. Conversely, upgrading sensors which provide irrelevant information is not expected to significantly improve the CPS' performance. Hence, the algorithms' effectiveness can be observed through the evacuation ratio in an experiment where critical sensors are hardened. The following experiments adhere to this logic and explore: *1)* how many hardened sensors are required to provide a significant improvement in the outcome, and *2)* how both algorithms compare.

A. Experiment 1 – quantity of hardened sensors

In this experiment, two different sets of sensors are selected for an upgrade to *hardened* models. This is achieved by increasing their "fire resistance" parameter in the simulation model. Table II details how sensors from the algorithm 1's ranking list are selected for each scenario. Initially, each scenario is repeated with the fire starting at each possible location. The preliminary results showed that nearly 75% of the fire outbreak locations made the evacuation process trivial – and the CPS of little use. We therefore excluded these scenarios to retain only those which really *challenge* the CPS. Figure 4a. shows the results for each scenario. All charts on Figure 4 show the average evacuation ratio, and results are expressed as a precentage of the *optimal* scenario's performance. This experiment shows that by upgrading less than 5% of the sensors to hardened models, a near-optimal (98%) performance level can be reached; while the *realistic* scenario only reaches 90% of the optimal performance in high-density scenarios.

B. Experiment 2 – algorithm comparison

In this experiment, we compare the results of both algorithms by upgrading the 13 top-ranked sensors of each

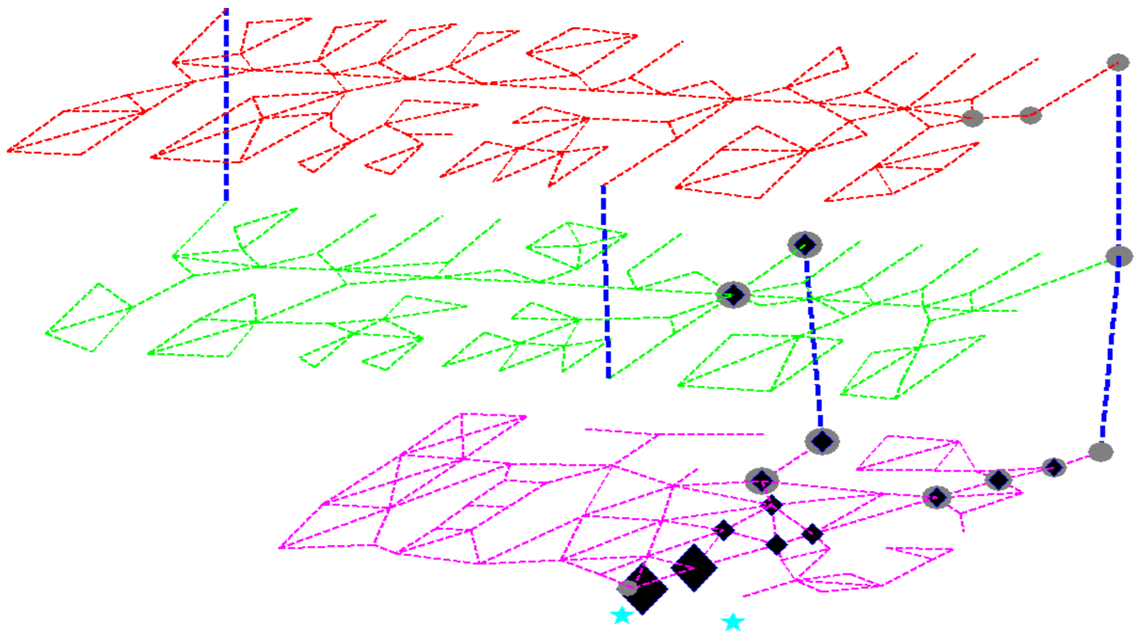

Figure 3. The 13 highest-ranked sensors from each algorithm. Circles correspond to Algorithm 1's output and Diamonds to Algorithm 2's. The size of the marker is proportional to the metric score. The two exits are at the bottom of the figure and marked by a pentagram.

	Nbr. of hardened sensors	Number of re-routings
Set 1	top 13 nodes / 240	All nodes which, if cut off, will cause more than 25 percent of the locations on the graph to re-route
Set 2	top 44 nodes / 240	All nodes for which at least one location will have to re-route if this node is cut off

Table II
SETS OF HARDENED SENSOR, AND SELECTION CRITERIA

algorithm. Figure 4b. shows the result of the simulations. Not only do both algorithms appear to perform well, they also yield equivalent CPS performance levels – with a slight advantage to Algorithm 2. This can be explained by the fact that both sets only differ by 7 sensors (out of 13).

A detailed analysis of individual simulations also revealed that each algorithm may be best suited for particular types of graph:

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

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

V. CONCLUSIONS

This paper examines the issue of (a) determining which of the sensors that are used to determine the outbreak, and then the propagation and management of an emergency, are most critical, and then (b) considers the effect of hardening the most critical sensors and then evaluates the effect of the hardening via simulations.

We introduce two algorithms which rank the sensors by order of criticality, in the context of building evacuations. This ranking is then used to selectively reinforce the sensors which are supposed to cover the most critical areas. Our simulation results show that very few hardened sensors are actually required to approach near-optimal performance. We also see that both of the algorithms presented perform equally well.

In future work we plan to address more carefully how such results can be coupled with adaptive routing techniques for evacuees [18] that can help reach better outcomes. Our future work should also pay more attention to the impact of incomplete or imperfect information [19] and to probabilistic modeling techniques [20] coupled with virtual reality simulations [21], [22]. We also plan to conduct additional research on the use of different types of building maps, with the objective to:

- Compare the performance of the algorithms in either flat, open-space areas or intricate, multi-storey buildings to evaluate which algorithms for criticality detection are better suited to different types of spatial lay-outs.

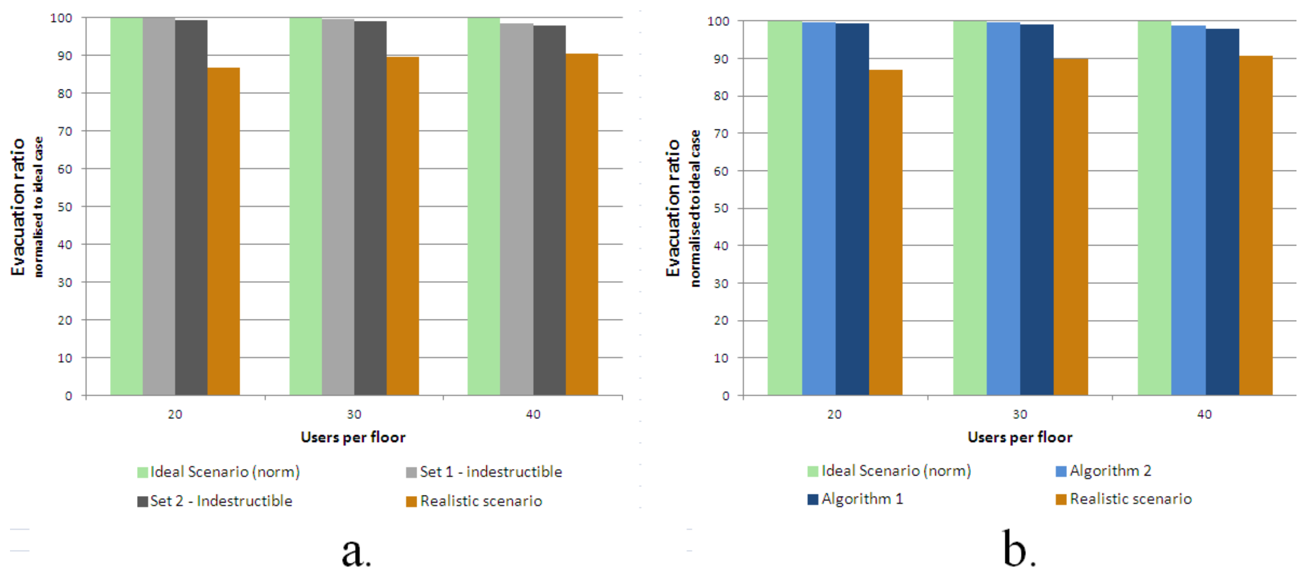

Figure 4. Simulation results

- Study and simulate more complex evacuation scenarios so that the effective performance of different approaches can be better evaluated.

One important issue that any such simulation encounters is that there are multiple types of criteria and levels of importance that need to be taken into consideration, and our work will need to evolve towards approaches that are able to include them within a unified framework [23], [24].

Our future work 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] J. Wong, H. Li, and S. Wang, "Intelligent building research: a review," *Automation in Construction*, vol. 14, no. 1, pp. 143 – 159, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580504000536

[2] L. Schor, P. Sommer, and R. Wattenhofer, "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*, ser. BuildSys '09. New York, NY, USA: ACM, 2009, pp. 31–36. [Online]. Available: http://doi.acm.org/10.1145/1810279.1810287

[3] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, ser. BuildSys '10. New York, NY, USA: ACM, 2010, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/1878431.1878433

[4] D. Snoonian, "Smart buildings," *Spectrum, IEEE*, vol. 40, no. 8, pp. 18 – 23, aug. 2003.

[5] E. Gelenbe and F.-J. Wu, "Large scale simulation for human evacuation and rescue," *Computers & Mathematics with Applications*, no. 0, pp. –, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0898122112002611

[6] G. Hoblos, M. Staroswiecki, and A. Aitouche, "Optimal design of fault tolerant sensor networks," in *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*, 2000, pp. 467 –472.

[7] Y. Ali and S. Narasimhan, "Sensor network design for maximizing reliability of linear processes," *AIChE Journal*, vol. 39, no. 5, pp. 820–828, 1993. [Online]. Available: http://dx.doi.org/10.1002/aic.690390510

[8] F. Wang, K. Ramamritham, and J. Stankovic, "Determining redundancy levels for fault tolerant real-time systems," *Computers, IEEE Transactions on*, vol. 44, no. 2, pp. 292 –301, feb 1995.

[9] D. Maquin, M. Luong, and J. Ragot, "Some ideas about the design of measurement system," in *European Control Conference, ECC'95*, Rome, Italie, Sep. 1995, pp. 3178–3183. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00310532

[10] W. S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie, "Fault tree analysis, methods, and applications 2013; a review," *Reliability, IEEE Transactions on*, vol. R-34, no. 3, pp. 194 –203, aug. 1985.

[11] A. Desmet and E. Gelenbe, "Interoperating infrastructures in emergencies," in *Proc. 27th International Symposium on Computer and Information Sciences*, Oct. 2012, (in press).

[12] E. Gelenbe and K. C. Sevcik, "Analysis of update synchronisation algorithms for multiple copy data bases," *IEEE Transactions on Computers*, vol. C-28, no. 10, pp. 737–747, October 1979.

[13] E. Gelenbe and G. Hébrail, "A probability model of uncertainty in data bases," in *ICDE*. IEEE Computer Society, 1986, pp. 328–333.

[14] J. Aguilar and E. Gelenbe, "Task assignment and transaction clustering heuristics for distributed systems," *Information Sciences*, vol. 97, no. 1, pp. 199–219, 1997.

[15] N. Dimakis, A. Filippoupolitis, and E. Gelenbe, "Distributed building evacuation simulator for smart emergency management," *The Computer Journal*, vol. 53, no. 9, pp. 1384–1400, 2010. [Online]. Available: http://comjnl.oxfordjournals.org/content/53/9/1384.abstract

[16] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, ser. MobiCom '04. New York, NY, USA: ACM, 2004, pp. 70–84. [Online]. Available: http://doi.acm.org/10.1145/1023720.1023728

[17] E. Nardelli, G. Proietti, and P. Widmayer, "Finding the most vital node of a shortest path," in *Computing and Combinatorics*, ser. Lecture Notes in Computer Science, J. Wang, Ed. Springer Berlin / Heidelberg, 2001, vol. 2108, pp. 278–287.

[18] E. Gelenbe, "Steps towards self-aware networks," *Communications of the ACM*, vol. 52, pp. 66–75, 2 2009.

[19] E. Gelenbe and G. Hebrail, "A probability model of uncertainty in data bases," in *Proc. 2nd Inter. Conf. on Data Engineering (ICDE'86)*, Feb. 1986, pp. 328–333.

[20] E. Gelenbe and A. Stafylopatis, "Global behaviour of homogeneous random neural systems," *Applied Mathematical Modelling*, vol. 15, no. 10, pp. 534–541, 1991.

[21] E. Gelenbe, K. Hussain, and V. Kaptan, "Simulating autonomous agents in augmented reality," *Journal of Systems and Software*, vol. 74, no. 3, pp. 255–268, February 2005.

[22] V. Kaptan and E. Gelenbe, "Fusing terrain and goals: agent control in urban environments," in *Multisource Information Fusion: Architectures, Algorithms, and Applications 2006*, vol. 6242. SPIE, April 2006, pp. 71–79.

[23] V. Atalay and E. Gelenbe, "Parallel algorithm for colour texture generation using the random neural network model," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 6, no. 2-3, pp. 437–446, 1992.

[24] E. Gelenbe and J.-M. Fourneau, "Random neural networks with multiple classes of signals," *Neural Computation*, vol. 11, no. 4, pp. 953–963, 1999.