Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

VALIDATION AND EVALUATION OF EMERGENCY RESPONSE PLANS THROUGH AGENT-BASED MODELING AND SIMULATION

Joseph E. Helsing
Harsha Gwalani
Armin R. Mikler

Department of Computer Science and Engineering
University of North Texas
1155 Union Circle
Denton, Texas, USA, 76203

Sultanah M. Alshammari

Computer Science Department
King Abdulaziz University
Jeddah, Saudi Arabia, 21589

ABSTRACT

Biological emergency response planning plays a critical role in protecting the public from possible devastating results of sudden disease outbreaks. These plans describe the distribution of medical countermeasures across the affected region using limited resources and within a restricted time window. The ability to determine that such plans will be feasible, in terms of successfully providing service to affected populations within the time limit, is crucial. Current efforts, such as live drills and training, to validate plans may not test plan activation at the appropriate scale or account for dynamic real-time events. This paper presents Validating Emergency Response Plan Execution Through Simulation (VERPETS), a novel computational system for the agent-based simulation of biological emergency response plan activation. This system integrates raw road network, population distribution, and emergency response plan data, and simulates traffic in the affected region using SUMO, or Simulations of Urban Mobility.

1 INTRODUCTION

Biological emergency response planning plays a crucial role in protecting the public from possible devastation resulting from disease outbreaks. In order for cities/counties to adequately prepare for potential biological emergencies, two key steps need to be taken. First, response plans need to be developed describing how a city would actually distribute medical countermeasures. This task is often performed by emergency managers and planners, who use personal knowledge of the area to decide where and how points of dispensing (PODs) should be set up to distribute these countermeasures. More recently, emergency managers and planners are utilizing software systems, such as RE-PLAN (O'Neill II, Mikler, and Schneider 2014) or RealOpt(Lee 2018), to make data-driven decisions. Second, the execution of these plans needs to be evaluated and validated, which is typically achieved through the use of drills and exercises. These live testing scenarios, however, are very limited with respect to scale as only a small percentage of the total population participates in these exercises. Additionally, real-time dynamic situations such as road blockages or closure of a POD are not tested in these exercises.

These limitations of live testing pose a significant risk to the public at large; therefore it is critical to find alternatives to exercises and drills as a means for evaluating and validating plans. Specifically, both the feasibility and efficacy of a plan must be thoroughly analyzed before that plan is executed. Feasibility refers to a plan's ability to meet certain key criteria, such as being able to medicate the entirety of the affected population within a specified time window. If a plan is determined to be feasible after testing, it can be deemed usable for a given emergency scenario. Efficacy describes, in general, how well a plan performs during ideal conditions or a controlled scenario. This can be measured in terms of how POD

978-1-7281-3283-9/19/$31.00 ©2019 IEEE

placement or the removal of road network segments affects network-wide traffic flow. Information of that nature can help distinguish between similarly feasible plans, in order to select a more optimal plan for later execution.

Validating Emergency Response Plan Execution Through Simulation(VERPETS), a novel agent-based system that simulates a response plan activation is presented in this study. VERPETS evaluates the feasibility of an emergency response plan by simulating the traffic activity caused by plan activation. Each individual or head of household in the affected region is treated as an agent in the system. During plan activation, these agents drive the shortest route to the POD or service center assigned to them to obtain medical supplies and then return home after receiving service. This research highlights the need for computational tools to validate response plans and shows that the proposed system VERPETS is successful in handling the dynamic nature of plan activation. VERPETS can be used to investigate the effect of traffic congestion and real-time road closures on the feasibility of the plan.

Existing computational tools to aid emergency response planners are discussed in the next section. The VERPETS system is described in detail in Section 3. Experiments and results are presented in Section 4, followed by the conclusion in Section 5.

2 BACKGROUND

Attempting to validate response plans only through the use of live drills and training exercises is both prohibitively expensive for examining multiple different plans and may not examine the PODs at full capacity (Nelson et al. 2012). Thus, it has been necessary to use modeling and simulation to evaluate various aspects of emergency response. The Integrated Emergency Response Framework (iERF) developed by Jain et. al. sought to bring a holistic approach to modeling emergency responses. In their work, they describe a set of tools and standards to address different emergency response needs. They highlight the importance of interoperability between available tools as well as support and training for responders and planners who will be using these tools (Jain and McLean 2003) (Jain 2003).

In addition to interoperability, the ability to simulate and examine plans from a variety of geographic scopes can provide emergency planners with a holistic view of a planned emergency response. From macro-scale simulations of national, state, or multi-county responses, to the mezzo-scale simulation of a single countys response plan execution, to the micro-scale level of the inner workings of a single POD, each scope sheds light on different facets of plans that can be useful for evaluation. At the macro-scale are simulations which examine emergencies that encompass hundreds of thousands to millions of people, and may involve one or more states or counties. At this scale, the geographic elements become increasingly important, such as ensuring that medicines are distributed efficiently to affected counties from the Strategic National Stockpile (SNS) to regional supply centers and then to individual counties. This takes into account individual PODs and how and when supplies are delivered to them. For example, the RE-PLAN system can be configured to create and evaluate biological emergency response plans at this scale. It allows users to design plans for one or more counties at a time, which potentially allows for cross county resource sharing. This takes into account the populations in those areas, their distribution, and the road network in the areas. Once a plan has been developed, RE-PLAN allows users to evaluate its feasibility and efficacy. This is performed by first determining if all PODs will be able to provide service for their respective populations, given a set of throughputs for each POD. After that, through the use of mathematical models, a user can analyze if the surrounding road networks will be able to adequately accommodate the traffic created by the activation of a plan. While this provides a substantial level of analysis, it ignores dynamic problems during plan execution such as traffic accidents or POD closures. Further, the mathematical rates also ignore fine grain traffic dynamics that appear when using agent-based models, such as the fact that vehicles may not travel at a constant speed at all time or the occurrence of gridlock (O'Neill II, Mikler, and Schneider 2014). Another example of planning and evaluation at the macro-scale is the RealOpt Regional system in a variety of different scenarios from the perspective of being on the ground during the emergency(Environmental Tectonics Corporation 2016).

At the micro-level, examining emergency response frequently provides a detail-oriented view of an emergency within a small area such as a portion of a city, a section of an area affected by the emergency, or inside a POD. For instance, DrillSim, as developed by Massaguer et. al., was constructed specifically to test a variety of IT solutions during an emergency situation. It evaluates a particular IT configuration by modeling agents, the affected area, the crisis itself, and the infrastructure involved. It further provides a number of visualizations for users to examine, in both a 2-D and 3-D live renderings, how the drills are executing the planned response(Balasubramanian, Massaguer, Mehrotra, and Venkatasubramanian 2006).

RealOpt also provides some POD evaluation functionality with RealOpt-POD. This system allows users to design and compare POD floor plans inter terms of usage and efficiency. It also analyses personnel usage and placement to determine how to minimize the average waiting time for individuals seeking treatment. Further, RealOpt-POD allows for simulations of the inner workings of a POD during POD activation, to allow for alternative analysis of multiple layouts and staffing configurations (Lee 2018).

Based on this review, there is a clear need for the use of computational tools in the field of emergency management. In the case of RE-PLAN and RealOpt, these systems allow the emergency manager and planner to develop response plans more quickly and easily. However, while they do provide some evaluative processes, this can be expanded upon in terms of geographic scale and the addition of more realistic processes. Thus, this work presents a new system that expands upon current emergency response plan evaluation and validation methodologies, while introducing new metrics and processes.

3 METHODOLOGY

The VERPETS system takes an emergency response plan as an input and simulates its activation to validate its feasibility. The input emergency response plan is created by RE-PLAN. This plan is described briefly in the next section.

3.1 Biological Emergency Response Plan

A Biological Emergency Response Plan (BERP) consists of a mapping of individuals or heads of households in the affected region to a facility where they can receive necessary medical supplies. A response plan may be created either at the individual level for vaccines or the household level for pill bottles depending on the type of medical service being provided. Point locations for individuals or households in a region are not known, so centroids of a Geographical Spatial Unit (GSU) are used as representative locations for all individuals/households in the GSU. The U.S. Census Bureau has divided each county in the U.S. into a hierarchy of different sized GSUs. Each county is made up of census tracts, and census tracts are further divided into census block groups. The lowest GSU in this hierarchy is a census block, i.e. each census block group contains one or more census block(s). RE-PLAN uses census block groups, the second lowest level in the hierarchy to represent individuals as the most reliable and up-to-date data is available at this level (Gwalani, Mikler, Ramisetty-Mikler, and O'Neill 2017). The block groups assigned to the same POD are combined to create catchment areas. Figure 1 shows an example of a response plan created by RE-PLAN. Each POD is usually equipped with multiple lanes or booths that can cater to the incoming population in parallel. Each POD P_i in the response plan must satisfy Equation 1 for the plan to be feasible. $TimeWindow$ is the time limit within which the entire population needs to be served. $pop(P_i)$ is the total demand to be served at POD P_i, while $numberOfBooths(P_i)$ defines the number of individuals that can be served concurrently, and processing time is the time need to serve one individual/household.

$$Time\ Window \geq \frac{pop(P_i) * ProcessingTime}{numberOfBooths(P_i)} \tag{1}$$

An emergency response plan is represented by a graph G(V,E), with vertices V and edges E. V is the combined set of all POD locations P, U: the set of nodes representing the centroids of the GSUs and set of nodes representing the intersection of two or more road segments I, i.e $V = P \cup U \cup I$. E is the set of all

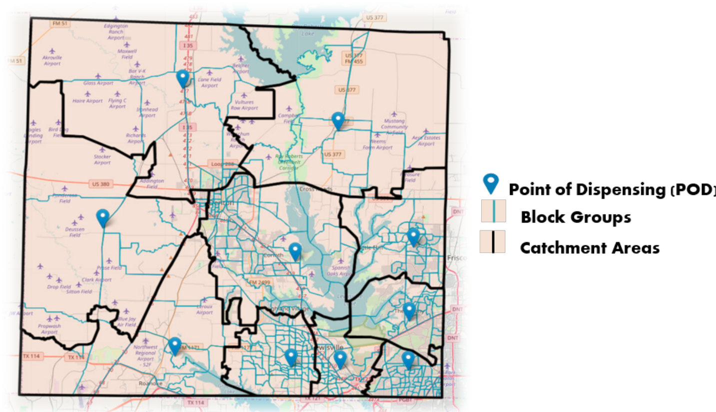

Figure 1: Biological Response Plan for Denton County.

road segments. The goal of the system is to simulate the movement of agents from the respective GSUs to the POD such that the travel time is optimized.

3.2 Data Requirements

The creation and validation of an emergency response plan require integration of demographic data, geospatial data on the affected region and the locations of the service centers or PODs, and road network data. These data requirements and corresponding sources are listed in Table 1.

Table 1: Data Requirements and Sources.

Data	Source
Demographic Data	American Community Survey, Tables: B01001 and B25002
Geo-spatial Data	Topologically Integrated Geographic Encoding and Referencing
Road Network Data	Open Street Maps

3.3 VERPETS Framework

As seen in Figure 2, the VERPETS framework is divided into three major components: the Master Controller, the Data Manager, and the Simulation Manager. The Master Controller initializes the VERPETS system via a configuration file, facilitates all communication between the Data Manager and the Simulation Manager, and reports final results. The Data Manager component functions as a centralized data storage and quick-access location for the VERPETS system. The response plan data generated by RE-PLAN and the road network data is stored in a PostgreSQL database. The Data Manager reads in the database tables and creates XML files needed to run the simulations. The Simulation Manager component is the core of the system's ability to evaluate emergency response plans. It handles the generation of files required by the simulator, the execution of one or more simulations, and the recording and processing of simulator output data. The Simulation Manager also performs static flow analysis.

Figure 3 presents a flow diagram of the major tasks performed by VERPETS. These tasks include preprocessing the OSM road network data to create a strongly connected network that can be used in the simulation, reading in the response plan created by RE-PLAN from the corresponding PostgreSQL tables and integrating the road network data created in the previous step, executing the traffic simulation at the PODs and reporting results on the success or failure of a plan activation. The VERPETS system can also be used to perform a static flow analysis to identify road segments that might suffer from heavier traffic flow due to their locations. These analyses can help emergency planners in taking additional steps to avoid

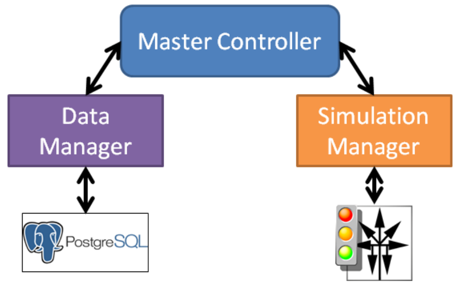

Figure 2: The three major components of the VERPETS system.

slowdowns if needed. The remainder of this section describes these tasks in detail.

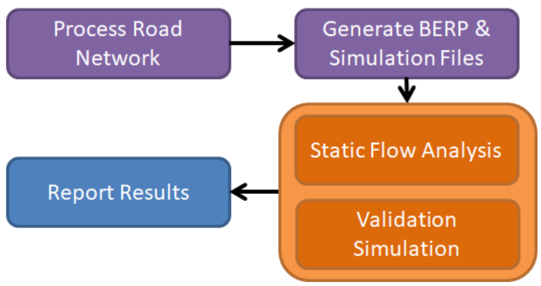

Figure 3: VERPETS: Flow Diagram.

3.3.1 Road Network Preprocessing

Simulation of Urban MObility (SUMO) was selected as the simulation engine for VERPETS. This engine boasts high portability, dynamic routing, and is free and open-source. Additionally, it requires only a simple set of XML files to setup an experiment and provides a concise XML formatted output that can be configured to display all or a selection of vehicles in the output. Further, it has been used to simulate the traffic patterns in the city of Cologne, Germany, containing roughly one million people, during a visit from the Pope and during the 2006 Soccer World Cup, so it can function with a large number of concurrent agents (Krajzewicz and Hertkorn 2002) (Behrisch, Bieker, Erdmann, and Krajzewicz 2011).

The OSM file downloaded for the road network data in the affected region cannot be used by VERPETS in its raw format to simulate the traffic during plan activation. An OSM file contains a list of tagged ways, where a way is a sequence of nodes or locations that make up a road or path. The simulation environment, SUMO provides a tool called NETCONVERT, that can take in an OSM file and convert it to a proprietary network file. This process breaks an OSM way into its individual road segments for easier simulation. Since many drivers rely on large road arteries when driving, this assumption was taken into account by NETCONVERT, whereby only highway roads that were labeled by OSM as primary, secondary, tertiary, motorway, or trunk were retained for the actual simulation. Additionally, any segment that was not attached to the majority of the network was deemed isolated and removed. The removal of these road segments, however, caused some locations in the network to become unreachable. While the majority of the U.S. road network has multiple routes to connect any road to other roads, by only selecting a portion of the road network and removing lower class roads, some of the highway class roads ended in a one-way segment that had no turn-around. In order to solve this problem, Tarjan's strongly connected components algorithm (Tarjan 1972) was employed on the dual graph to remove any road segment that could not reach or be reached by every other segment.

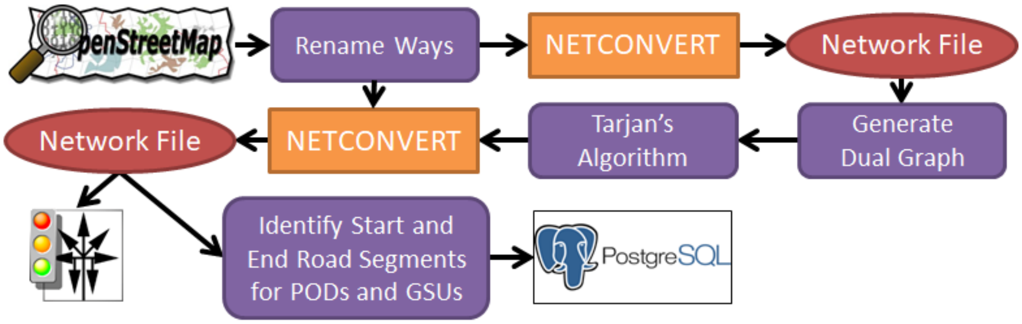

Figure 4: Workflow for processing OSM data.

Once the weakly connected components are discovered, NETCONVERT is executed again to remove those particular road segments. These network files are XML files that define each road segment (edge) and the corresponding attributes including coordinates, lanes, speed limit and length. This produces a strongly connected road network on which routing can now be automatically performed and a round-trip route for a census block group to its associated POD is guaranteed. This road network data is then used to identify the road segments corresponding to the GSUs and PODs to mark the start and end point of the trip for an agent.

3.3.2 Validation Simulation

The agents or cars leave from the edge closest to the corresponding GSU, reach the edge their PODs are located at and make it back to the starting edge in the simulation. These edges form the destinations a vehicle must visit during its trip. SUMO requires the creation of this trip file to generate a route for a vehicle. The edge-by-edge route for each vehicle is generated using DUAROUTER, which uses Dijkstra's shortest path algorithm on the road network to compute the routes where edges are weighted by travel time. In order to receive treatment at a POD location, an individual must spend a certain amount of time in processing at the POD. This processing time accounts for activities such as filling out and submitting forms, standing in line, answering questions, and actually receiving the medication. This processing time was controlled by the stop parameter in SUMO. The number of spots in the parking lot at a POD simulated the lanes or booth functionality.

The departure time of a vehicle can be altered to simulate different vehicles leaving at different times. Effectively, the simulator loads a car into the simulation at its departure time, where it waits in a queue, the insertion-backlog, to be inserted onto the road network. Once inserted, the vehicle follows its route to its POD location and then back to the road segment it was inserted on. Upon reaching the final edge in the route, the car is removed from the simulation and marked as arrived.

The vehicle release interval controls how often cars are loaded into the insertion-backlog. If the vehicle release interval is set to 0, all cars are loaded into the simulation and their respective backlogs at the beginning of the simulation. Given a sufficiently large number of agents, this can cause significant slow-down in the performance of the simulator.

One of the strategies to increase performance and decrease the time to simulation completion was to break the plan into its component catchment areas and simulate each catchment area in parallel. Due to these catchment areas being formed by Voronoi partitioning, the overlap between them was minimized. However, because of the possibility of perturbations caused by these overlaps, experiments were performed to analyze the difference between simulating all of the agents in a plan at the same time and simulating the catchment areas in parallel.

A different strategy employed was to simulate all agents at the same time, halt the simulation after the cut-off time, and predict the remaining time using a rate based analysis. In the event that the average arrival rate is greater than the processing rate, cars will begin to queue at the POD. This is significant because if the arrival rate is maintained throughout the simulation, once cars begin queuing the queue will not become empty until the last waiting car is processed. We call this situation of a perpetually expanding queue at the POD runaway congestion. Runaway congestion significantly decreases the relevance of travel-time on plan feasibility, as the POD processing rate becomes the rate-determining step due to the POD being at perpetual maximum utilization.

If runaway congestion is allowed to occur, then after a period of time, travel time does not impact plan completion. Thus, after this occurs, a rate-based calculation can be performed to estimate plan completion time. To calculate this, first the total number of unprocessed vehicles must be calculated using Equation 2, where *Total Vehicles* is the count of all households in the county, *Total Processed Vehicles* is the number of vehicles processed before the simulation was halted, and *Total Exited Vehicles* is the number of vehicles that have been processed and exited the simulation. Once the number of unprocessed vehicles has been determined, the BERP completion time can be calculated using Equation 3. In this equation, *Final Timestep* is the last timestep simulated by the simulator, *Average Processing Time* is the average processing time for all booths, *Total Booths* is the total number of booths in the BERP, and *Unprocessed Vehicles* is the result from Equation 2. These equations effectively estimate the completion time due to the processing rate becoming the rate-limiting step of the simulation.

$$\textit{Unprocessed Vehicles} = \textit{Total Vehicles} - (\textit{Total Processed Vehicles} \ - \textit{Total Exited Vehicles}) \tag{2}$$

$$\textit{BERP Completion Time} = \textit{Final Timestep} + \frac{\textit{Average Processing Time}}{\textit{Total Booths}} * \textit{Unprocessed Vehicles} \tag{3}$$

However, before these equations can be used, it must be determined when a POD has reached runaway congestion. Preliminary experiments were performed to test whether some standard time could be used to halt the simulation and then estimate the final completion time. Using SUMO's GUI, a visual inspection of when a POD reached runaway congestion was performed. In this case, a POD was said to have achieved runaway congestion if the number of agents waiting to enter the POD was one plus the number of booths.

4 Experiments and Results

To test the effectiveness of VERPETS's ability to evaluate emergency response plans, multiple experiments were performed using different emergency response plans. In this study, three plans were developed to examine VERPETS's performance and is described here. VERPETS's ability to validate each plan was examined, and compared to analysis available via RE-PLAN.

4.1 Response Plans

As stated, in order to evaluate VERPETS's ability to validate biological emergency response plans (BERPs), it was necessary to generate multiple BERPs. This was performed by using RE-PLAN to develop plans for Denton County, Texas, USA. Figure 5 shows the census block groups and the road network and general demographic statistics for Denton County.

Table 2: A list of the response plans used for evaluating VERPETS.

County	Response Plan Name	Number of PODs	Total Number of Booths	Processing Time (sec)
Denton	denton_1	29	429	180
	denton_2	22	322	180
	denton_3	24	358	180

In order to simulate a BERP for each of these counties, three emergency response plans were created. Table 2 describes the three plans. All of these plans assume a 36 hour deadline for the treatment of

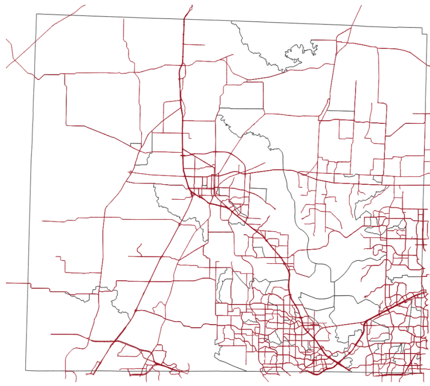

Figure 5: Census block groups(black) and the road network(red) in Denton County.

the entire population, and were constructed to examine the four different possible outcomes of plan development: Theoretically Feasible and Realistically Feasible, Theoretically Feasible and Realistically Infeasible, Theoretically Infeasible and Realistically Feasible, and Theoretically Infeasible and Realistically Infeasible.

A plan is said to be theoretically feasible if the plan satisfies Equation 1. It is deemed theoretically infeasible otherwise. A plan is said to be realistically feasible if the total completion time, as determined by plan activation simulation, is less than or equal to the time limit. A theoretically infeasible plan cannot be realistically feasible. The total processing time of a plan is defined by Equation 4. The completion time for a plan is always less than the total processing time.

$$\textit{Total Processing Time} = (\textit{Total Population} * \textit{Processing Time})/\textit{Total Number Of Booths} \tag{4}$$

The first plan was designed to have a total processing time of around 30 hours, which is well below the deadline. This represents a plan that was both theoretically and realistically feasible. By setting the processing time at 30 hours, this ensured that even with some possible traffic delays, the plan would complete before the 36-hour limit. The second plan for each county was designed to have a processing time for 40 hours, thus causing it to exceed the deadline by a great margin. This represents a plan that was both theoretically and realistically infeasible, due to the insufficient number of resources allocated to the plan. The third plan for each county was designed to have a processing time as close to, but below, the 36-hour time limit. This plan was designed to be theoretically feasible, but potentially could be realistically infeasible when activated due to traffic dynamics and booth distribution.

4.1.1 Static Flow Analysis

Once a response plan was generated, an analysis of the traffic flow across the road network could be performed. Because SUMO generated a route for each census block group before simulation, the number of individuals on each road segment could be calculated. Thus, an analysis of which road segments had the largest number of agents cross them during plan activation could be performed. The Jenks natural breaks classification method (Jenks and F. 1971) was used to generate the five classifications. Figure 6 shows the results of these analyses for denton_1,denton_2 and denton_3.

As can be seen in Figure 6, some of the PODs are relatively close together, which may cause some census block groups from one POD to use the same road network segments as census block groups from another POD. In order to determine the extent of this potential overlap, further analysis was performed.

To calculate the overlap between different PODs' usage of the road network, the route from each census block group to its POD were first generated. Next, based on those routes, the number of PODs using each road segment were counted. This was used to determine the number of road segments that were unused, used by one or more PODs, and used by more than one POD.

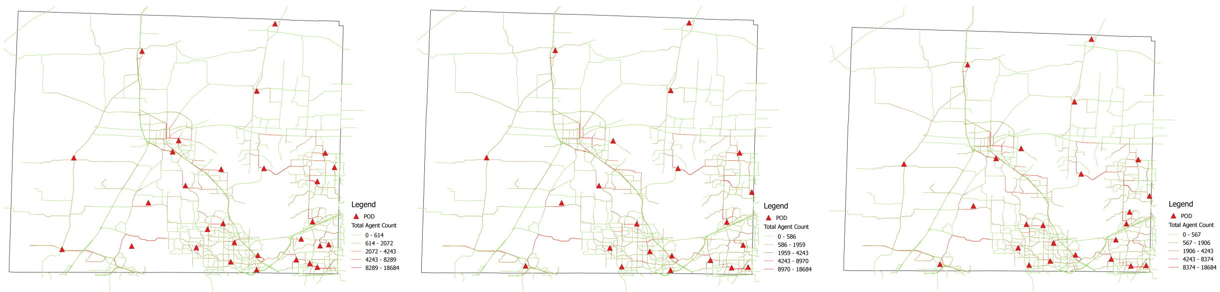

Figure 6: The load on the road network for Denton county during plan activation for the plans denton_1, denton_2, and denton_3 respectively.

Figure 7 shows the outcomes for the three Denton county plans. Less than half of the total road network for Denton was used for routing. Additionally, only roughly 6% of the road segments for denton_1, roughly 5% of the road segments for denton_2, and roughly 6% of the road segments for denton_3 were used by census block groups from more than one POD. Again, it can be concluded that overlap on the road network was minimal between the PODs in these plans. This analysis supports the strategy of executing independent simulations for catchment areas in parallel to reduce the load on the simulation engine.

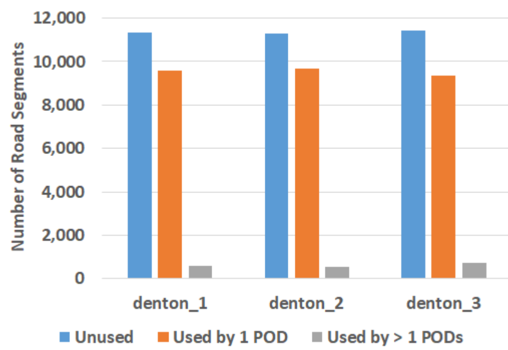

Figure 7: The number of road segments used for routing denton_1, denton_2, and denton_3.

4.2 Simulated Time Limit

Due to the dynamics caused by runaway congestion at the PODs, it was inferred that the simulation could be halted early and the completion time extrapolated. If a standard time limit could be identified, then plans could quickly be run and then their completion time estimated. However, because travel time to the POD differs between census block groups, this standard time limit must be experimentally determined.

For this experiment, using SUMO's GUI, a visual examination of each census block group in a plan was performed. The goal was to determine the time (in simulated seconds) it would take for a POD to reach runaway congestion. It was assumed a POD had reached runaway congestion when all of its booths were full, and a number of agents equal to the number of booths plus one, had queued up in front of the POD waiting for service. Each plan was simulated with a POD processing time of 180 seconds, and at a 1:1 scaling factor using the plan denton_1.

Different vehicle release intervals can affect the time to runaway congestion, therefore, three different intervals were used. The first interval test was releasing all agents into the simulation at the same time. This has an effective release rate of one agent every 3 seconds, but may decrease performance as all agents are in the insertion-backlog. The effects of releasing one agent every 30 seconds and one agent every 60 seconds were also examined. Figure 8 shows the average and maximum time to runaway congestion.

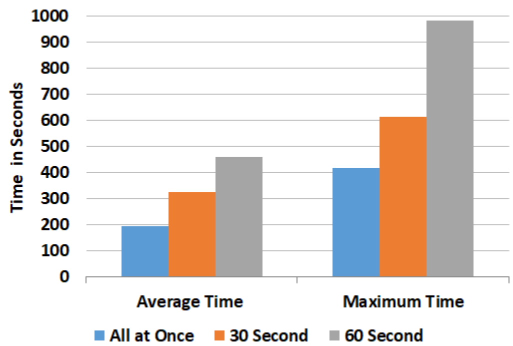

Figure 8: The average and maximum time in seconds, to runaway congestion.

Based on the results of these experiments, 3600 seconds or one hour, was chosen as a safe time limit. In addition to determining a time limit, the effects of using that limit were examined. Each of the three plans was simulated with a processing time of 180 seconds, a release rate of 30 seconds. Figure 9 shows the results of these simulations and it was seen that there was no variation about the mean for each of the experiments. As the results show, the completion times for each of these experiments was stable as the variation was almost negligible.

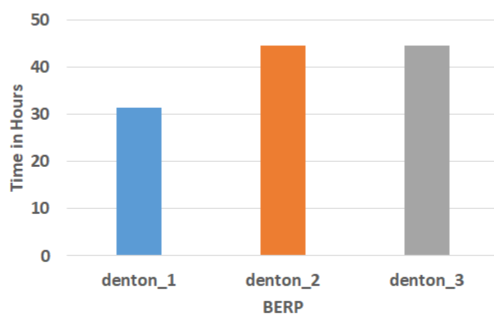

Figure 9: The average time, in hours, to plan completion using a one hour time limit.

4.3 Simulation vs Rate-based Validation

To determine the effectiveness of using a time limit on simulation it was compared to the rate-based time results from RE-PLAN. The simulated completion time is compared with the theoretical minimum completion time or the total processing time (Equation 4), and the estimated completion time from RE-PLAN, which is the maximum time taken by a booth to serve its demand. The completion time for the parallel catchment area strategy is the maximum time across all catchment areas. The theoretical completion time assumes an equal distribution of the population assigned to each booth. However, catchment areas created by Voronoi Partitioning does not ensure an equal distribution of the population across PODs. Figure 10 shows the results of these experiments.

5 DISCUSSION

Based on these experiments, it is shown that agent-based simulations of emergency response plan activation could be effectively performed. First, it is demonstrated that the results of simulating the activation of

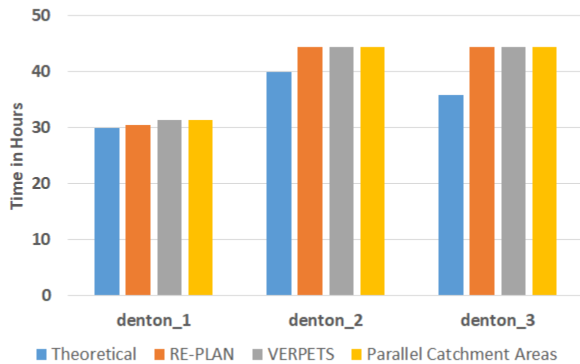

Figure 10: A comparison between the theoretical, RE-PLAN POD Manager, and VERPETS determined completion times.

each individual catchment area are comparable to the results of simulating the activation of an entire plan at once. This allows for a reduction in load on the simulator by allowing for multiple instances of SUMO to be run in parallel. This also improves performance as multiple agents can now be processed in parallel across multiple threads.

Next, the use of a time limit for the simulation to further decrease agent load was tested. It is shown that using a one-hour time limit is a safe limit as all examined catchment areas reached runaway congestion before one hour of simulated time. When using the time limit, the results are shown to be stable across all plan activation simulations.

Finally, simulating response plan activation using VERPETS is compared to RE-PLAN's rate-based analyses. Both RE-PLANs and VERPETS's analysis determined a longer plan activation than the theoretical analysis, because theoretical analysis ignores traffic time and booth distribution across PODs. As expected, VERPETS's plan completion time is greater than RE-PLAN's expected plan completion time because of travel time considerations. VERPETS can effectively validate emergency response plans and determine their feasibility in real time.

Biological emergency response planning plays a critical role in protecting the public from the possible devastating results of sudden disease outbreaks. Thus, the ability to validate that the activation of a plan will successfully provide service to an affected population within some time limit is crucial. While emergency response plan generation software tools such as RE-PLAN (O'Neill II, Mikler, and Schneider 2014) or RealOpt (Lee 2018) (Lee, Maheshwary, Mason, and Glisson 2006) do provide some rate-based validation analyses, an agent-based validation system is presented to allow for additional realistic traffic dynamics to be modeled and evaluated in this work.

Finally, VERPETS can be expanded to validate emergency response plans for other hazards. In the case of hurricanes, planning for contraflow lanes and evacuation routes must be planned and constructed before a hurricane arrives (Centers for Disease Control and Prevention 2016). This necessitates evaluation of these routes and plans in advance so that upon activation, potential problems have already been identified and mitigated. Additionally, in the wake of a hurricane, supplies such as fresh water, and medical supplies must be distributed in a timely and effective manner. Potentially, these supplies can be distributed via mobile distribution centers, and these resource distribution plans can be tested via simulation. Thus, VERPETS could be restructured to both test evacuation strategies and mobile resource distribution strategies.

REFERENCES

Balasubramanian, V., D. Massaguer, S. Mehrotra, and N. Venkatasubramanian. 2006. "DrillSim: a simulation framework for emergency response drills". In *Intelligence and Security Informatics*, 237–248: Springer Berlin Heidelberg.

Behrisch, M., L. Bieker, J. Erdmann, and D. Krajzewicz. 2011. "SUMO–simulation of urban mobility: an overview". In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.

Centers for Disease Control and Prevention 2016. *Notes on the Interim U.S. Guidance for Monitoring and Movement of Persons with Potential Ebola Virus Exposure*. http://www.cdc.gov/vhf/ebola/exposure/monitoring-and-movement-of-persons-with-exposure.html.

Environmental Tectonics Corporation 2016. *Advanced Disaster Management Simulator*. http://www.trainingfordisastermanagement.com/.

Gwalani, H., A. R. Mikler, S. Ramisetty-Mikler, and M. O'Neill. 2017. "Collection and Integration of Multi-spatial and Multi-type Data for Vulnerability Analysis in Emergency Response Plans". In *Advances and New Trends in Environmental Informatics*, 89–101: Springer International Publishing.

Jain, S. 2003. "An Integrating Framework for Modeling and Simulation for Emergency Response". In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, 1068–1076.

Jain, S., and C. McLean. 2003. "A framework for modeling and simulation for emergency response". In *Proceedings of the 2003 Winter Simulation Conference*, 1068–1076. IEEE.

Jenks, G., and C. F.. 1971. "Error on Choroplethic Maps: Definition, Measurement, Reduction". *Annals of the Association of American Geographers* 61(2):217–244.

Krajzewicz, D., and G. Hertkorn. 2002. "SUMO (Simulation of Urban MObility) An open-source traffic simulation". In *Symposium on Simulation*, 63–68.

Lee, E. 2018. *Overview of RealOpt Software Enterprise*. Georgia Institute of Technology. http://www2.isye.gatech.edu/medicalor/realopt/index.php.

Lee, E., S. Maheshwary, J. Mason, and W. Glisson. 2006. "Large-Scale Dispensing for Emergency Response to Bioterrorism and Infectious-Disease Outbreak". *Interfaces* 36(6):591–607.

Nelson, C., A. Parker, S. Shelton, E. Chan, and F. Pillemer. 2012. *Analysis of the Cities Readiness Initiative TR-1200*. RAND Corporation.

O'Neill II, M., A. R. Mikler, and T. Schneider. 2014. "RE-PLAN: An Extensible Software Architecture to Facilitate Disaster Response Planning.". *IEEE Trans Syst Man Cybern Syst* 44(12):1569–1583.

Tarjan, R. E. 1972. "Depth-First Search and Linear Graph Algorithms". *SIAM Journal on Computing* 1(2):146–160.

AUTHOR BIOGRAPHIES

SULTANAH M. ALSHAMMARI is a faculty member in the Computer Science Department at King Abdulaziz University. She completed her PhD in Computer Science and Engineering from the University of North Texas. Her research interests include computational epidemiology, modeling and simulation, and data science. Her email address is sshammari@kau.edu.sa.

HARSHA GWALANI is a PhD student in the Department of Computer Science and Engineering at the University of North Texas. She works as a research assistant at the Center of Computational Epidemiology and Response Analysis (CeCERA). Her research interests include optimization and partitioning algorithms, modeling and simulation, graph theory and computer graphics. Her email address is harshagwalani@my.unt.edu.

JOSEPH E. HELSING is a lecturer in the Department of Computer Science at the University of North Texas and a member of the Center of Computational Epidemiology and Response Analysis (CeCERA). His research focus is on model and simulation validation, specifically in the area of computational epidemiology. His email address is joseph.helsing@unt.edu.

ARMIN R. MIKLER is a professor and associate department chair in the Department of Computer Science at the University of North Texas. He is the director of the Center for Computational Epidemiology and Response Analysis (CeCERA). He established the Computational Epidemiology Research Laboratory (CERL, cerl.unt.edu) with focus on the development of a computational methodology to model and simulate the spread of diseases and the design and analysis of bio-emergency response plans. His email address is mikler@unt.edu.