Civilian Evacuation and Route Planning Coordinated with Emergency Response

Double Blind Proposal
Georgia Institute of Technology, Atlanta, GA, 30332 USA

***Abstract*—Evacuation Route Planning (ERP) has grown ever more imperative due to the increasing impact of natural disasters and threats of terrorism on modern infrastructure. To help mitigate loss due to these and similar large-scale situations such as nuclear disasters [1], ERP tasks are becoming more complex. Other complicating elements include increasing population densities, larger structures within urban environments served by multifaceted transportation networks. Because of the multitude of factors, all driving the problem toward greater scales, the ability to handcraft ideal ERPs has become nearly infeasible. This has led emergency planners to turn to artificial intelligence algorithms to quickly generate optimal paths for large-scale civilian evacuations under multiple scenarios. The authors propose extending the method of Capacity Constrained Route Planning (CCRP) [2] to include optimal paths for incoming emergency responders as well as for evacuees.**

Keywords—Evacuation Route Planning, Emergency Response, Artificial Intelligence, Multi Agent System.

I. Related Work

RECENT work within ERP falls into three methods: network flow methods, simulation methods and heuristic methods [2]. Network flow methods consist of minimizing total evacuation time under constraints through time expanded graphs (TEG) or using iterative algorithms optimizing cost functions [3], [4]. Simulation methods focus on individual movements to such a labor-intensive extent that the effort attenuates into the game of "herding cats." Heuristic methods have proved recently to be most promising, including CCRP, which uses quick and dirty time-aggregated graphs with the capacity constraints to rapidly and repetitively identify workable routes [4], [5].

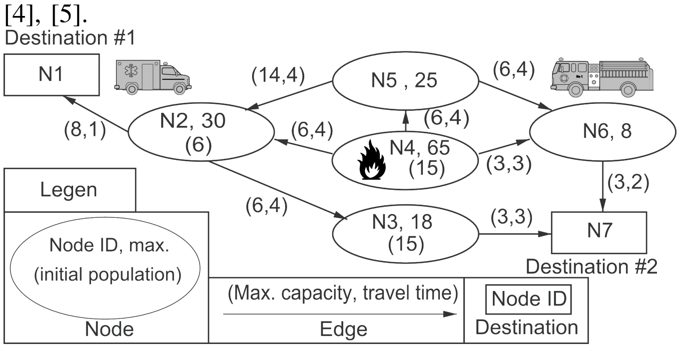

Fig. 1: ERP problem: Network Model for simple ERP scenario. Adapted from S. Shekhar et al [2]

II. Approach

The authors approached the problem by first creating the evacuation scenario as described by Shekhar et al [2].

Such a scenario could include a capacity constrained evacuation network of tremendous scales such as citywide transportation infrastructures with huge populations. An ERP algorithm need to be scalable [2] to include any or all transportation networks including but not limited to roads, subways, helipads, or even a building's exit routes such as rooms, corridors and stairwells. In principle, the algorithm proposed is capable of providing evacuation plans for large scenarios that can be modeled as a capacity constrained network.

Each Node within the evacuation network traditionally possesses two important parameters: current population, and maximum capacity. As the current population can vary over time, maximum capacity is constant and remains a limiting factor of the throughput or the maximum population for any given vertex.

Each Edge within the network will also commonly include two parameters: travel time and edge capacity. Travel time simply describes the duration of time steps to traverse the entire edge connected between two nodes. Edge capacity however, unlike a nodes maximum capacity, does not describe the maximum population limit of a given edge, but rather the maximum rate at which a populace may enter the edge. This distinction is quite subtle, but is however more akin to limitations of flow rates and evacuation times within reality. This distinction will also help simplify our implementation of CCRP, and can be considered one of the reasons this algorithm is less computationally expensive and more scalable when compared to other alternatives [2]. The growth in complexity of the algorithm is $O = nplog(n)$ where n is the number of nodes and p is the population.

For any type of evacuation agenda, zones defining both hazards and havens are defined, giving an evacuee route planner a set of endangered sources and refuge destinations. From there, we would like to generate evacuation plans for all the evacuees while minimizing total evacuation time. CCRP does this by calculating the quickest possible route at the soonest possible time given a list of sources and destinations. This is done by generating a single pair fictitious super-source and super-destination vertices that connect to the list of sources and destinations respectively using edges of zero travel time and infinite edge capacity.

This allows for a simple search such as Dijkstra's algorithm to solve for the quickest path between the two super vertices using edge travel time as the weight criteria. Once this path R is found, the maximum flow along that path is calculated by determining the minimum of the three parameters i.e. current source population, minimum available edge and node capacity along the path. The bookkeeping for available capacities within the graph is done using a time-series dictionary that allows the algorithm to check and update reservations along the path. If the quickest path at the current start time has no more available capacity, it repetitively looks into the future by a time step and checks for available capacity along same path again.

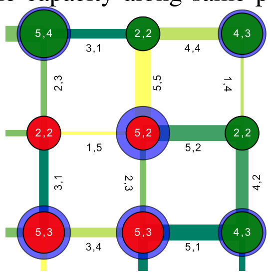

Fig. 2: Node numbers represent maximum capacity, denoted by the size of the shaded blue outline and population colored red for evacuation zone. The edge numbers represent edge capacity denoted by edge width and travel time denoted by green to yellow for fast to slow. (Best Viewed in Color)

This calculation of the flow ensures that the capacity of any given path is not exceeded while simultaneously maximizing evacuation times. What this does not yet achieve is the ability to plan for multiple parties with different objectives. Evacuees may start from a source in danger and migrate to a destination for refuge while inversely emergency responders may start anywhere with the goal to migrate to a destination in danger. This list of separate sources and destinations for separate parties results in additional complexities and optimizations in counter flow. A simple strategy would be to apply CCRP consecutively for emergency responders after completion of the algorithm for evacuees. While this would not hamper evacuation time, this remains non-optimal for response time, essentially under-utilizing the capacities that become available during the gradual evacuation.

As explained earlier, the authors proposed to implement an algorithm that would provide improved response time by allowing a slight trade off in evacuation time, thus benefiting the survival rates of victims who are trapped or left behind and are the most in danger.

By overhauling the majority of the original CCRP algorithm for N parties, major planning alterations included expanding the time-series bookkeeping to reflect multiple party populations with separate time-series while maintaining a shared reservation list for the entire network. Another challenge addressed was the modification necessary to the original Dijkstra algorithm in order to properly solve for the quickest paths between matching party sources and destinations.

Because almost all arrangements of evacuee and responder source and destination overlaps are possible, such as rescuers and evacuees sharing the same source location, or evacuees having the same destination as a responder's source and vice versa. A parallelized approach was developed by generating party-specific sub-sources and sub-destinations that were connected to their respective locations, while also sharing a common super-source or super-destination parent. Simply put, our modified search algorithm verifies that the current quickest path found includes a source and destination for a common party. Additional precautions are necessary to ensure that fictitious paths that backtrack down one party sub-source and up another that provide a teleportation-like behavior across the real transportation network are not utilized.

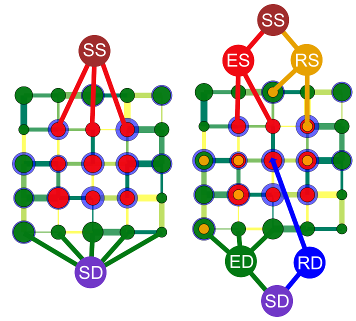

Fig. 3: Left: Original CCRP with Super Source and Destination Nodes. Right: New CCRP with sub Source and Destination Nodes for Evacuees and Responders

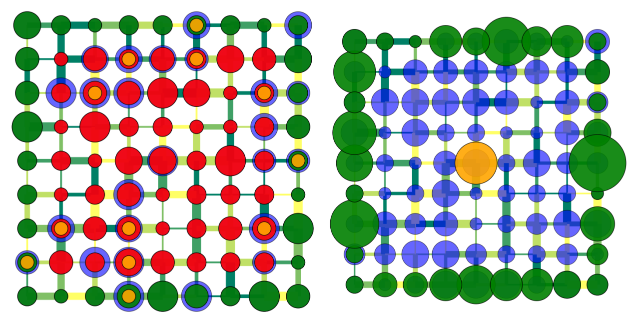

Fig. 4: The graphs right to left shows the population before and after evacuation and emergency response. Responder populations are assigned the color orange.

III. Evaluation

For the evaluation for our multiparty CCRP algorithm, a basic comparison between the new and original CCRP with respect to evacuation time is perhaps not the best descriptor in performance. As the evacuation start and end time for evacuees will be arbitrary within the time sequence of the new evacuation plan. Thus, a more curious inquiry could be made about whether it is better to evacuate sooner after the disaster, or over a shorter duration of time. This is similar to whether the risk is incurred in time spent exposed when moving or time remaining stationary, analogous to situations involving hurricanes or forest fires.

We propose a more general comparison by evaluating both algorithms within multiparty emergency scenarios. Because the original CCRP algorithm does not inherently support multiple parties, our simple strategy as first mentioned within the beginning of our approach, the original algorithm will simply be run twice; once for each party, sequentially one after another. Both algorithms can then be compared against one another by means of total completion time.

For the actual test trials, the two CCRP approaches are conducted and repeated 1000 times on a 9 by 9 Manhattan grid network, with the center vetex the disaster location with all but the edge vertices within the danger zone. See Figure 4. A random capacity constrained graph generated each trial was for both approaches, this graph includes random maximum node capacity, edge capacity and travel time between 1 and 5. Initial populations are also random from 1 to the maximum node capacity, with each node having a 50% chance of including a responder if allowed by the maximum node capacity.

As seen in Table 1, we saved an average of 9 time steps on 1000 iterations with a minimum time saved of 1 time step and the maximum time saved of 13. As seen within the resulting histograms, our new CCRP provides a more responsive and realistic route planner for multiple parties. Even though moving both parties simultaneously now takes longer that moving any individual party, the overall time is still less than the cascading time to move one after the other. Rather than waiting upon the completion of one partys migration prior to the next, both parties travel as soon as they are able while simultaneously sharing the same capacity constrained transportation network. This is better akin to how real emergency resolutions are conducted where all personnel need to be deployed in a rapid manner [1]. What we have shown is how to coordinate all of those involved to optimize total completion time.

IV. Discussion

Among the things we learned were some of the drawbacks to the CCRP, as touched upon in the beginning of our approach. CCRP will only search for the quickest route with respect to only the edge travel times, but then adjusts the flow rate to compensate for the available capacity. This is to say, that our Multi-Party CCRP algorithm is complete, i.e if a solution exists, it will find it. However it is still sub-optimal in that paths that are examined may not have available capacity, and thus the wait for availability may result in a longer total time that an alternate detour may provide. A similar analogy could be a patron in a theme park who wishes to wait in a long line for the food vendor that is closest, rather than expending time walking to other vendors with shorter lines where food could be bought sooner. Methods such as time expansion, where the quickest path is found given the current time as well as future capacity constraints, can produce more optimal routes [4]. However the larger memory usage and computational cost are still prohibitively expensive for large-scale citywide evacuations. This makes Multi-Party CCRP a candid solution because the added complexity for party computation is still subdued by CCRP's overall simplicity.

Future work for this project could include an examination on N number of party interaction and route planning times. Specifically how the growth in the number of agents in unique parties can affect an individuals and all parties' completion time. Another feature valuable for real-life emergency coordinators would be an ability to apply weighted priorities to better address the non-uniform urgency and importance each party brings in regard to the larger scheme of things.

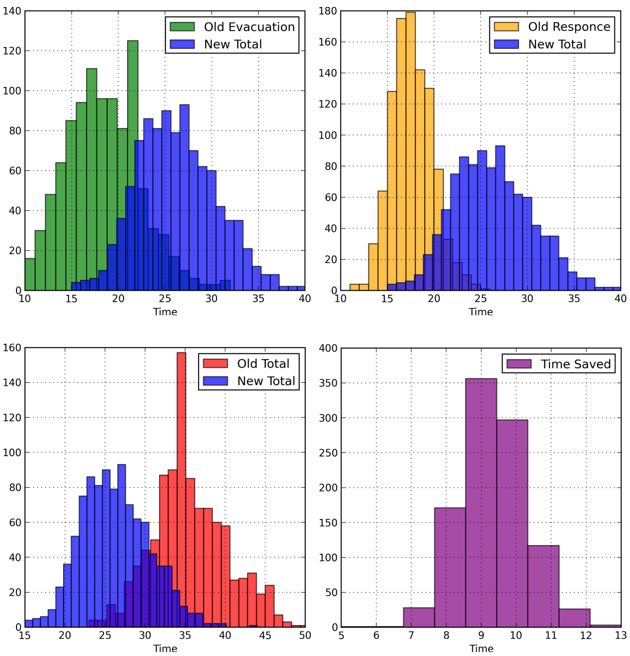

Fig. 5: Top Left: Histogram comparing Old Evacuation vs New Total Time. Top Right: Old Response vs New Total Time. Bottom Left: Old Total vs New Total Time. Bottom Right: Total Time Saved using new CCRP on a trial by trial basis.

	Mean	Std Dev	Min	Max
Evacuation Old	18.408	3.910	10	32
Response Old	17.168	2.223	11	26
Total Old	35.576	4.722	23	56
Total New	26.189	4.421	15	44
Time Saved	9.3870	1.098	5	13

TABLE I: Data Distribution of the results displayed in fig 5

References

[1] K. Khalil and M. Abdel-Aziz, "The Role of Artificial Intelligence Technologies in Crisis Response," *arXiv preprint arXiv: ...*, 2008. [Online]. Available: http://arxiv.org/abs/0806.1280

[2] S. Shekhar, K. Yang, V. M. Gunturi, L. Manikonda, D. Oliver, X. Zhou, B. George, S. Kim, J. M. Wolff, and Q. Lu, "Experiences with evacuation route planning algorithms," *International Journal of Geographical Information Science*, vol. 26, no. 12, pp. 2253–2265, Dec. 2012. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/13658816.2012.719624

[3] F. Itwm, "Mathematical Modelling of Evacuation Problems : A State of Art," vol. 24, no. 24, 2001.

[4] B. George and S. Shekhar, "Time-Aggregated Graphs for Modeling," pp. 85–99, 2006.

[5] M. Ryan, "Constraint-based multi-robot path planning," *2010 IEEE International Conference on Robotics and Automation*, pp. 922–928, May 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5509582