Emergency Routing

Karthik Mohan, Amarpreet Singh, Nirmit Zinzuwadia

University of Toronto
27 King's College Circle, Toronto, Ontario M5S 1A1 Canada

karthik.mohan@mail.utoronto.ca
amarkbr.singh@mail.utoronto.ca
nirmit.zinzuwadia@mail.utoronto.ca

***Abstract*— The efficiency of Emergency Dispatch and Routing is a major public safety concern. It involves utilizing real-time traffic data and flexible dispatching strategy to meet the crucial response time. This paper describes a case study of a potential dispatch of firetrucks and ambulances to a current accident location. This project concentrates on developing an optimization model for developing flexible dispatching strategies that take advantage of available real-time travel accident data. The objectives considered include minimizing the total travel time to a dispatch location while eliminating busy areas, which is what the public is concerned with the most. The major modules of this simulation model include travel time prediction, dynamic shortest path, incident/vehicles tracking, and dispatch optimization. The study includes a comparison between A*, a deterministic informed search algorithm, and Particle Swarm Optimization (PSO), a stochastic search algorithm, with modified heuristics to see which kind of algorithm elevates the performance. The real-time dispatch strategy, a key component of this simulation model, is validated by various samples.**

***Keywords*— dispatch routing; emergency response; optimization; generic algorithm**

I. Introduction

The public's concern for safety has generated the motivation to improve the coordination and information sharing of public transportation agencies. Transportation agencies, mostly run by governments, have to provide a 24-hour service at the public's disposal since emergencies can occur at any time. There is an incentive to design a platform where the public is served in the case of an emergency without any hurdles along the transportation route since, in most cases, it's a question about life or death. Hence, it is important to design an integrated decision-making platform for public safety agencies (e.g. fire trucks, ambulances, etc.)

The project aims to develop and evaluate a real-time emergency response system that uses Bing live traffic incidents [1] along with heuristics based on the past history of accidents and construction sites to guide optimal routes for emergency vehicles. The model is tested on real and made-up scenarios to outline the performance metrics. The proposed solution enables re-route diversion from crowded or busy locations. Emergency service vehicles cannot simply rely on sirens to get the right-of way on a busy/crowded road. It is important to re-route and choose a path that saves time through other means.

We have utilized A* and PSO algorithm with extra flavors of genetic algorithm and randomness to compare the performance of the algorithm we have patched. The end result integrates the mathematical programming formulation and the dynamic shortest path algorithm into the proposed emergency routing system for the city of North York, Ontario, Canada.

The following paper first outlines the problem metrics in Section II; followed by a math formulation and algorithm insights in Section III. It then presents a few demos to outline and compare the performance of our algorithm. At last, we conclude the paper with some of the related work in this area and how it relates/differentiates from our research.

II. Problem Characterization

A. Target Users

The primary target users of our Emergency Routing problem include emergency responders. We consider fire fighters, paramedics, and police officers to be the entities that form the emergency responders for our problem. Another key target users are the bystanders that are involved in the emergency. We split bystanders into two categories: (1) individuals directly involved in the emergency, and (2) individuals indirectly affected by the emergency. For example, if we consider a severe car accident caused by two cars colliding, then the two cars that collided are the category (1) bystanders, and all of the other cars on the road that are delayed by the accident are the category (2) bystanders.

B. Challenges

The foremost challenge to overcome is to develop a robust real-time or live traffic data collection system, and then integrate the live traffic data to our mathematical solution in a meaningful manner. We concluded our preliminary investigation of the data-sets and traffic APIs available online by selecting the Bing live traffic incidents API. We are relying on this API because it is robust and has a simple user interface. We simply provide a geographic bounding box, which consists of four coordinates (north latitude, south latitude, west longitude, east longitude), and it returns a list of live traffic incidents occurring in the geographic bounding box. A clean data ingestion system is used to fetch, parse and transform this information so that we can integrate it with our mathematical solution.

Acquiring up-to-date data about emergency responders is also crucial for our problem. For instance, in the domain of fire stations, these details include the number of fire trucks that a specific fire station accommodates, and the serving capacity of each fire truck before it needs to return to its home fire station in order to replenish its supplies. We need similar details in the domain of hospitals.

C. Objectives

To evaluate our emergency routing solution we have established several goals that need to be satisfied at the same time. Firstly, we want to incorporate two traffic heuristics: (1) that uses live traffic incidents for our region of interest, and (2) that uses historical traffic incidents data for our region of interest. Both live traffic and historical traffic heuristics are used to suggest a route that exposes the least risk of delays. Secondly, we want to minimize the total number of emergency responders that are dispatched for any particular emergencies at a given time. This involves minimizing the number of fire trucks and ambulances required to serve the emergency completely. Finally, we want to minimize the total travel time of all the responders to the emergency site. We analyze the travel time for two scenarios: (1) for serving one emergency at a time, and (2) for serving multiple emergencies that occur concurrently.

D. Constraints

We have decided to constrain our problem to the geographic region of North York, Ontario, Canada. This means we evaluate our solution by applying it on the data of North York, which includes hospital, fire truck, traffic, and other map related data. Furthermore, we constrain the emergency responders to fire trucks and ambulances. We do not take into account other emergency responders like air ambulances or aerial firefighters.

III. Problem Formulation & Modelling

A. Data

We use osmnx API [2] to get the information about the nodes and edges contained in the map of North York. The locations of fire stations and hospitals are identified using the Overpass scripts which pinpoint the exact location of an amenity. We also use live and historical accident and construction regions to identify the optimal path for an emergency vehicle to meet a crucial response time. We utilize Bing Traffic API to capture this information. We used the kMeans algorithm on this data to help us capture zones of busy areas. In addition, live accident data is used to identify the busy streets. We designed an algorithm that integrates all these data to fine-tune the routing of emergency vehicles.

B. Mathematical Formulation

In formulating the mathematical model, the following notations are used in this section.

M = total number of emergencies
K = total number of emergency responders available
N = total number of emergency responder centers
D_i = number of emergency responders dispatched from location i ($i \in [1, N]$)
ER_i = location of emergency responder i ($i \in [1, K]$)
E_i = location of the emergency i ($i \in [1, M]$)
R_{ij} = travel time from responder i to emergency j

Objective 1: use live traffic data and historical traffic data to suggest a route that exposes the least risk of delays.

We use live traffic data to form subgraphs near each live traffic incident, and for each edge in the original maps that falls inside the subgraph, we adjust the risk score of the edges and the nodes. The risk score is a function of severity of the traffic incident and is penalized by the distance of the traffic incident to the edge. Since we know the distribution of severity from historical data. We can assume it's probability density function is known and hence treat the risk score at a point as a random number.

$$E(R) = \sum P(\text{Severity}) \text{ x } \text{Penality}$$

Objective 2: minimize the total number of emergency responders that are dispatched for any particular emergencies at a given time.

$$\min(\sum_{i=1}^{N} \mathrm{D_i}) \text{ s.t. } \mathrm{D_i} <= \mathrm{N} \ \forall \ \mathrm{D_i} \in \mathrm{D}$$

Objective 3: minimize the total travel time of all the responders to the emergency site.

The closeness metric between any two points takes into account the live traffic data and historical traffic data heuristics defined in Objective 1. This objective can be achieved by assigning the ER with the least risk score among all ER's to the emergency since each edge is penalised by the distance between that edge to the emergency.

$$\min(\sum_{j=1}^{K} \sum_{i=1}^{M} \mathrm{R_{ij}})$$

where R is the risk-score which is close to 1 if $\mathrm{ER_i}$ is far from $\mathrm{E_i}$ and close to zero if $\mathrm{ER_i}$is risk-free for the emergency.

The K-means algorithm is applied on the historical data to help us capture zones of busy areas in the past. We use this historical insight to further adjust the risk score of each edge on our graph.

C. Live Traffic Data and Risk Scores

Using Bing live traffic incident data, we assign a risk score to the nodes and edges of North York. We generate a circular subgraph around each accident and evaluate the risk of North York nodes and edges that lie in the subgraph. Risk of every node and edge is found based on two criteria: (1) severity of the traffic incident, and (2) distance from the traffic incident to the node or edge. Risk is proportional to the incident severity, and inversely proportional to the distance between the node or edge. Therefore, a node is riskier than other nodes if it is closer to a traffic incident that is very severe. Below, we present the pseudo-code for the logic that updates the risk of each node in North York based on the traffic incidents.

```
for incident in traffic_incidents do
  radius ← random_int_between(50, 100)
  subgraph ← generate_subgraph(incident.location, radius)
  for node in subgraph.nodes do
    severity ← incident.severity
    distance ← distance_in_meters(incident, node)
    risk ← severity / distance
    node.risk ← node.risk + risk
  end
end
```

We randomly select the subgraph radius (in meters) for each traffic incident. To achieve a reasonable runtime, we randomly selected a subgraph radius between 50 meters and 100 meters. Based on trial and error, we usually observe a few hundred live traffic incidents, and by tuning our subgraph radius hyperparameter in the above manner, we complete risk analysis within 30 minutes. An identical process is followed to find the risk of each edge. Finally, we transform the risk of each node and edge into probabilities using normalization.

D. Historical Traffic Data and K-Means

We obtained historical traffic data using the Bing Traffic API. Over a period of 2 weeks beginning in October 2020, we periodically called the Bing Traffic API every hour and collected North York traffic incident data. We then supplied the historical traffic incident data to our K-Means algorithm, which was configured to use 16 clusters. We chose 16 clusters because it captured 97% of the variability in the historical traffic incident data. We interpret each cluster center as risky locations based on historical traffic incident data.

After obtaining the coordinates of the K-Means clusters we treat it like the live traffic incidents and perform the same risk analysis on it as previously described. However, we give less priority to the historical traffic data by (1) lowering the severity for each cluster location, and (2) increasing the radius of the subgraph formed around each cluster by a factor of 2.

E. Search Algorithms

We tried A* and PSO to find the routes that the emergency responders would use to reach the emergency locations. We treated the 10 most severe accidents in North York returned by the Bing Traffic API as emergency locations.

The emergency locations are treated as sources, and the traffic responder locations are treated as destinations. We considered two cases: (1) when there are more emergency responders than emergencies, (2) when there are more emergencies than emergency responders. We analyzed case (2) by treating the location of the first emergency as the origin of the next available emergency responder. We assume fire trucks can hop from one accident location to the next, whereas ambulances need to return to the hospital in order to drop off patients before they can route to the next emergency location.

A*: We adjusted the A* cost heuristic by using the node risk scores generated from live and historical traffic data. We scale up the heuristic score for nodes that are more riskier than others.

PSO: We adjusted the PSO cost heuristic by using the edge risk scores generated from live and historical traffic data. We scale up the heuristic score for edges that are more riskier than others.

IV. Performance Evaluation

Case 1 routes for A* and PSO is shown below, where an ambulance (red) and a firetruck (white) is sent to the emergency incident.

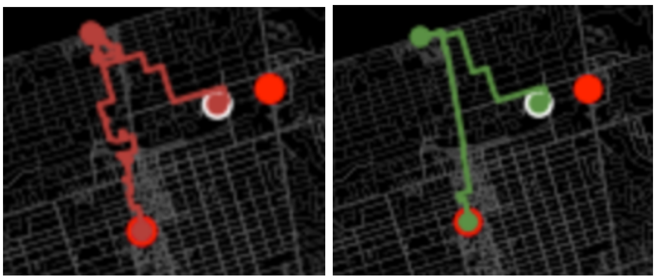

Figure 1: A* (left) and PSO (right) case 1 routes

For Case 2, when the Emergency responders are already operating at their maximum capacity, for the remaining emergency incidents we try to map it with the closest responder. Such points where the emergency responders travel from incident 1 to incident 2 are represented as yellow nodes.

Figure 2: A* case 2 routes

PSO is less performant than A*. Execution time of PSO was much higher than the A*. In the table below, case I considers a scenario with a single accident and case II considers a scenario with two accidents where there is only one ambulance and one fire truck.

Table 1: CPU times of a runtime for each case

	A*	**PSO**
Case I	1 min 37 sec	3 min 43 sec
Case II	3 min 24 sec	7 min 21 sec

Similar pattern in execution time was observed for Case 2 (ER < Accidents) in both the algorithms, with A* algorithm having a faster execution time. The reason for the longer execution time of PSO is because of the longer routes and less amenities, due to which the algorithm routes the responders available at the farthest places in the city to the emergency locations.

V. Related Work

The problem of emergency vehicle routing is of course not a novel problem. People have already looked at this in the past. In fact, every city has public safety agencies that utilize various algorithms to reach the destination in question in the fast way possible. Our goal here is to apply new and novel heuristics to an existing algorithm to differentiate how the end results compare. Others have tackled and tried to improve this problem in a similar manner. Table 2 shows the techniques utilized by other people within this research community.

Table 2: Related work and algorithms

Paper	**Algorithm**	**Notes**
[3]	Discrete Inspired Bat Algorithm	❑ Considers a case where multiple fire trucks need to be deployed to handle a fire at a large location ❑ Considers CO2 emissions of the fire trucks in the cost function of the

		routing algorithm
[4]	Genetic Algorithms	❏ Provides different parameters and determine configurations that provide best results for the path panning of emergency vehicles ❏ Uses team selection mechanism along with crossover and mutation for the next generation to find the best fit. Authors try multiple trials to attain the perfect result.
[5]	Markov Chain and heuristic based on a queueing approximation	❏ Here, the authors show that driving-time correlation has a non-negligible impact, hence authors focus on dispatching heuristic to lower the travel times. Dispatching simply from the nearest location may not be an optimal solution for later routes. ❏ New heuristic seems to be as optimal and requires less computational effort.
[6]	Dynamic shortest path algorithm	❏ Simulated model presents a conceptual design of a real-time EMS model ❏ Utilizes integer programing with few changes in various parameters
[7]	Hybrid algorithmic approach: particle swarm optimization algorithm and path relinking strategy	❏ Deals with emergency vehicle routing during disasters where supplies are limited ❏ Improves not only timeliness but also "utilization and sustainability of emergency service, the allocation of the emergency supplies and the emergency vehicle routes"

Table 1: Related work findings

As can be seen from Table 1, researches have incorporated various algorithms, however, the most important aspect which improves upon the use case is the heuristics and cost function authors have utilized. Some take CO2 emissions and time into considerations, while others use dispatching strategies to optimize the problem even further. All these solutions present unique approaches. There is no one-go to answer to this complex problem. Similarly, we present our own heuristics of live traffic data and mutations to the algorithm to see how well it bolds.

VI. Conclusion And Future Work

In this paper, we explored the emergency routing problem using live traffic incident data, fire stations, and ambulances in the region of North York, Ontario, Canada. We relied on the Bing Traffic API to obtain live traffic data. We explored solving the problem using the deterministic A* algorithm, and the swarm intelligence-based PSO algorithm. Our experiments to use simulated annealing were unsuccessful because of the large runtime associated with our problem size of North-York. In all cases, we used live and historical traffic data to guide emergency dispatch vehicles by using the idea of risk.

Future work can focus on adapting the subgraph logic for node and edge risk generation based on the number of live traffic accidents. Furthermore, we can focus on reducing the runtime of our algorithm by caching edge coordinate data that is retrieved using OsmAPI. Alternatively, we can also explore caching the map.

Non-deterministic algorithms give us a lot of control over tweaking the algorithm. However, when the size of the map increases, we find that the algorithm becomes computationally expensive. Whereas deterministic algorithms like A* provides faster results. For our objective, we find that both the algorithms comparatively provide similar paths, with PSO providing opportunities for explorability.

References

[1] "Traffic API - Bing Maps", Docs.microsoft.com, 2020. [Online]. Available: https://docs.microsoft.com/en-us/bingmaps/rest-services/traffic/. [Accessed: 17-Oct-2020].

[2] "osMnx API package", 2020. [Online]. Available: https://osmnx.readthedocs.io/en/stable/osmnx.html. [Accessed: 15-Oct-2020].

[3] Trachanatzi, D., Rigakis, M., Marinaki, M., & Marinakis, Y. (2020). A Discrete Inspired Bat Algorithm for Fire Truck Dispatch in Emergency. Natural Risk Management and Engineering: NatRisk Project, 203.

[4] Constantinescu, Vlad & Patrascu, Monica. (2020). PERFORMANCE ANALYSIS OF GENETIC ALGORITHMS FOR ROUTE COMPUTATION APPLIED TO EMERGENCY VEHICLES IN UNCERTAIN TRAFFIC.

[5] Usanov, D., van de Ven, P. M., & van der Mei, R. D. (2020). Dispatching fire trucks under stochastic driving times. Computers & Operations Research, 114, 104829.

[6] A. Haghani, H. Hu, Q. Tian. "An Optimization Model for Real-Time Emergency Vehicle Dispatching and Routing". In Transportation Research Board. 2013.

[7] Qin, Jin & Ye, Yong & Cheng, Bi-rong & Zhao, Xiaobo & Ni, Linling. (2017). The Emergency Vehicle Routing Problem with Uncertain Demand under Sustainability Environments. Sustainability. 9. 288. 10.3390/su9020288.