Congestion-Aware Stochastic Path Planning and Its Applications in Real World Navigation

Kamilia Ahmadi and Vicki H. Allan
Computer Science Department, Utah State University, Logan, Utah, U.S.A.

Keywords: Stochastic Path Planning, Multi-Agent Systems, Congestion-Aware Modelling, Non-Linear Objective, Route Planning under Uncertainty.

Abstract: In the realm of path planning, algorithms use edge weights in order to select the best path from an origin point to a specific target. This research focuses on the case where the edge weights are not fixed. Depending on the time of day/week, edge weights may change due to the congestion through the network. The best path is the path with minimum expected cost. The interpretation of best path depends on the point of view of car drivers. We model two different goals: 1) drivers who look for the path with the highest probability of reaching the destination before the deadline and 2) the drivers who look for the best time slot to leave in order to have a smallest travel time while they meet the deadline. Both of the goals are modelled based on the cost of the path which is highly dependent on the level of congestion in the network. Minimizing the paths' cost helps in reducing traffic in the city, alleviates air pollution, and reduces fuel consumption. Findings show that using our proposed intelligent path planning algorithm which satisfies users' goals and picks the least congested path is more cost efficient than picking the shortest-length path. Also, we show how agents' goals and selection of cost function impacts paths' choice.

1 INTRODUCTION

Path planning finds a path from a specific origin to a destination over a network of road segments. Path planning algorithms use the road segment costs in order to come up with the best path. If the road segments' costs are fixed, planning the best path through the network is a well understood task via the algorithms like Dijkstra and A* algorithms (Dijkstra and W., 1959; Hart et al., 1968). However, in real world navigation problems, depending on the level of congestion on the road segments, the cost associated with the legs of the trip changes over time. Also, it is not feasible to use an adaptive algorithm in every step due to the urgency in having a quick response to the queries and hesitancy of drivers to change their route frequently.

In modeling a city scale graph, congestion changes throughout the day which results in having uncertain costs on the road segments (Nikolova and Karger, 2008; Rus, 2020; Geisberger et al., 2010; Yaoxin Wu et al., 2016). Congestion is highly affected by the path selection of drivers in the network. In addition, there are many factors that affect the congestion pattern such as road conditions, drivers' path choice, time of the day, weather conditions, and events throughout the city (Rus, 2020; Geisberger et al., 2010; Wilkie et al., 2011; Sigal et al., 1980; Pi and Qian, 2017; Niknami and Samaranayake, 2016).

We consider expected travel time on the road segments as the cost of that segment. The variability of congestion level on road segments makes it a stochastic network. Minimizing the paths costs, ultimately results in reducing the city scale congestion by picking less congested paths. Reducing congestion throughout the city has the benefits of decreased pollution, fewer accidents, less wasted time, and less fuel costs (Chiabaut et al., 2009; Fan et al., 2005; Rus, 2020; Pi and Qian, 2017; Yaoxin Wu et al., 2016).

This paper focuses on path planning over a stochastic network which is a graph of a city. The challenge is to find the best paths under uncertainty and the constraints of real world domain. Agents are car drivers which can pursue different goals: first, the ones who are not willing to take risk and look for the path with highest probability of reaching destination before a desired arrival time, even if it may take them longer. Second, the agents who are open to take a riskier decision if it helps them in having the smallest en-route time. These agents are flexible in leaving

Ahmadi, K. and Allan, V.
Congestion-Aware Stochastic Path Planning and Its Applications in Real World Navigation.
DOI: 10.5220/0010267009470956
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 947-956
ISBN: 978-989-758-484-8
Copyright © 2021 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

anytime while they still need to make the trip.

To make it clearer, one good example of these kind of agents' goals is in the context of a package delivery system. For example, suppose that we guarantee the delivery of a package by 4 PM, otherwise the customer doesn't accept the delivery and we lose the shipping costs. In that case, we are interested in picking a path that has the highest chance of reaching destination before the deadline to avoid losing the shipping cost. The other possible case is delivering perishable products. For example, if we promised the delivery of perishable products before 6 PM to the customers, we are interested to pick a path that has the smallest en-route time due to the nature of our package. In this case, we are flexible in leaving anytime, but we do need to have the smallest en-route path while still making the destination before 6 PM.

As mentioned earlier, the definition of best path differs based on the goal of the agents. For finding the best path, queries have an origin, a destination and the desired arrival time (deadline) along with the agents' goals. Our proposed path planning framework, models the city as a stochastic network, utilizes pruning techniques to reduce the size of search space, defines the path costs, aligns them with agents goals and picks the minimum cost path.

2 LITERATURE REVIEW AND CONTRIBUTION

Miller-Hooks and Mahmassani (Miller-Hooks and Mahmassani, 2000) consider travel costs as edge weights of a navigation graph in their model. Costs depend on travel times, and their goal is to find the least expected travel time in peak and non-peak time of the day. Then they solve an equivalent deterministic problem. The main concern with this framework is there has been little work on considering uncertainty, congestion awareness and time dependency of edge weights in finding the optimal path.

Fan, Kalaba and Moore (Fan et al., 2005) consider a special monotone increasing cost based on the probability of arriving late and suggests that the Gamma distribution is natural for modelling stochastic edge travel times. The probability calculation requires computing a continuous-time convolution product. Therefore, it makes the path planning a computationally expensive and time consuming task.

Niknami et al (Niknami and Samaranayake, 2016), present a model to compute the route that maximizes the probability of on-time arrival in stochastic networks. Their method uses a heuristic for the optimal path that chooses the direction at every intersection based on the current state by evaluating zero-delay convolution on the path probability and expected travel time. However, they assume that travel time distributions are exogenous (not impacted by individuals routing choices) which makes it not desirable as in realistic domain path choices are affected by other drivers' decisions as a major source of congestion on stochastic networks.

Zhiguang (Cao, 2017), proposed the Probability Tail model based on a cardinality minimization problem by directly utilizing travel time data on each road link. Then, the minimization problem is approximately solved via relaxing the cardinality by $L1$-norm, and formulating it as a mixed integer linear programming problem. For extracting the edge weights, it uses travel time samples on each arc as input and adopts some random distributions to generate the weights. As the result, this model doesn't consider traffic patterns for different times of a day.

Rus et al. (Rus, 2020) proposes stochastic path planning method where edge weights are represented as linear combination of mean and variance of travel time (mean+ λ * variance) controlled by a λ parameter. In their model, the key property is that the optimal path occurs among the extreme points of the convex hull containing all the path points. Then λ is used to prune the search regions and selects only a small number of λ values. The best path is found by Dijkstra (Dijkstra and W., 1959) based on minimizing the cost function of two modelled goals: a) probability tail model and b) mean risk model. Since, Rus's model uses exhaustive enumeration for path selection, it's run time in average is $O(n^2 log^4(n))$.

In our work, we propose a path planning model which uses Rus's work (Rus, 2020) as a conceptual framework but provides practical improvements on top of it. Firstly, we do not linearly combine mean and variance of travel time on edge weights. We consider travel costs in intervals of 10 minutes for each day of a week and extract the typical mean and variance on that edge for the specific time slots. Means and variances are as short as 10 minutes time segments to present the variation in any given point of day/time accurately. Also, we study three options of cost functions (linear, exponential, and step cost function) to have a better understanding of main classes of cost modelling and their impact in path selection. However, the model is general enough to include any cost function. (details of cost functions are explained in 3.5)

Our proposed path planning algorithm has two main steps: a) pruning search region to select few candidate paths among all possible paths, and b) Planning an optimal path from the candidate paths in step *a*. In pruning phase, a node is expanded if expected mean of the travel time of the approximate path through the node is less then the user's deadline. Proposed pruning algorithm utilizes graph clustering and approximation techniques (explained in detail in 3.4). Majority of pruning work utilizes pre-computation, which makes it very efficient and practical to real work domain. The second part of proposed path planning algorithm focuses on picking the path which has the minimum cost aligned with agents' goals.

The last contribution of this paper focuses on modelling agents' goals. First group of agents are looking for the path that maximizes the probability of reaching a destination before the deadline (highest probability path). Second group, look for the best departure time slot in order to have the least travel time and arrive at the destination before deadline, these agents are interested to take riskier decision if it provides them shortest en-route time. Details of agents' goal modelling is discussed in 3.6.

3 MODEL DESCRIPTION

3.1 Open Street Map Data

For building the city graph, we used Open Street Map data (Haklay and Weber, 2008). Open Street Map is a collaborative open source project which creates a free editable map that can be used widely. Open Street Map represents physical entities on the ground like buildings, roads, intersections, bridges and so on. It uses the basic data structure of entities and tags for describing the characteristics of that entity. The data structure includes nodes, ways, and relations. A *node* is a single point in space defined by its latitude, longitude, and node id. A *way* is a list of nodes used to represent linear features such as a series of roads. A *relation* is a multi-purpose data structure that relates two or more data elements like a route, turn restriction, traffic signal or an area. We used map matching techniques to match the OSM data to our logged traffic data.

3.2 City Graph Edge Weights

The city is modelled as a directed graph consisting a set of vertices, V, which represent road intersections and edges, E, that represent road segments between vertices. We consider the city graph to be planar (i.e., edges intersect only at their end points) (Rus, 2020; Nikolova and Karger, 2008). Associated with each edge of the graph is edge weight (W) which is not fixed, and it is represented by an expected travel time random variable in terms of mean and variance of the delay on that edge at the specific time shown in Equation 1.

$$W_{edge}(t) = (m_{edge}(t), v_{edge}(t)) \tag{1}$$

We compute time segments in the intervals of 10 minutes for each day of a week. For finding the mean and variance of each edge in time segments of a week, we summarized yearlong traffic data based on 10 minutes time segments for each day of a week. The target city in this model is Salt Lake City, Utah, and we use monitored traffic data from Utah Department of Transportation (UDOT) to extract edge weights of the city graph.

Travel time of each edge is an independent Gaussian random variable (Ahmadi and Allan, 2017; Rus, 2020; Nikolova and Karger, 2008; Long et al., 2006; Wilkie et al., 2011; Sigal et al., 1980; Fan et al., 2005). Since the sum of independent Gaussian random variables is also a Gaussian random variable, the travel time for the whole path is also Gaussian (shown in Equation 2).

$$t_{path} \sim Normal(m_{path}, v_{path}) \tag{2}$$

We consider edge weights to be independent from each other as the time dependent variance on edges represents the dependency of the congestion on adjacent edges (Rus, 2020; Nikolova and Karger, 2008; Wilkie et al., 2011; Sigal et al., 1980; Campbell et al., 2011; LAU et al., 2012; Niknami and Samaranayake, 2016). For example, suppose that edge e takes 30 percent longer than when congestion free in a specific time slot, an adjoining edge is likely to take 30 percent longer than when congestion free in the same time slot. Then for the specific edge and its adjoining edge, the variance reflects all of these changes throughout different time slots of the day. It is also possible to consider the stochastic dependency between edges by transforming the graph in a way to add a new edge between two dependent edges with mean equals to 0 and variance equals to covariance of the weights of two dependent edges (Rus, 2020; Nikolova and Karger, 2008; Long et al., 2006; Fan et al., 2005). However, since this makes the city graph even more interconnected and complex, we model edge weights as independent while variance on edges represents the dependencies.

The mean of a path is the sum of the means of all edges included in the path considering sliding time window (δ) (Equation 3).

$$m_{path}(t) = \sum_{e \in path} m_e(t+\delta) \tag{3}$$

Variance of the path is the sum of variance values of all edges included in the path from an origin O to destination D including sliding time window δ (Equation 4) (Rus, 2020; Nikolova and Karger, 2008; Chiabaut et al., 2009; Campbell et al., 2011; LAU et al., 2012).

$$v_{path}(t) = \sum_{v \in path} v_e(t+\delta) \tag{4}$$

If we consider each edge as an independent random variable, then the sum of variances is derived from (Equation 5). Since we assume edge weights are independent from each other, then $cov(X_i, X_j)$=0 for $i \neq j$ and Equation 6 is the result. Based on Equation 6, the variance of a path is the sum of variance of all edges included in the path shown in Equation 4.

$$var(\sum_{i=1}^{n} X_i) = E([\sum_{i=1}^{n} X_i]^2) - [E(\sum_{i=1}^{n} X_i)]^2 \tag{5}$$

$$var(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n}\sum_{j=1}^{n} cov(X_i, X_j) = \sum_{i=1}^{n} cov(X_i, X_j)$$
$$= \sum_{i=1}^{n} var(X_i) \tag{6}$$

For finding the mean and variance of a path, a sliding time window has been considered. A sliding time window implies that the cost of each edge in the path depends on the amount of time that took to reach it, not just the initial departure time. For example each $m_e(t)$ is actually $m_e(t+\delta)$ in which delta is the estimated arrival time from source node to edge e.

3.3 Agents

We consider drivers as agents. Agents get suggested directions from a central path planner by entering source, destination, deadline and their goal. Definition of best path may be different from the point of view of one agent to another. Having the origin (O), target (T), and deadline (D), here are the two main questions that clarifies agents' goals in this model.

- What is the path with the maximum probability of reaching destination before the deadline? (the most secure path, hence might be longer)
- What is the best time to leave in order to have the smallest travel time and reach the target before the deadline? (riskier decision, while getting smallest travel time path)

3.4 Pruning Heuristic in Path Finding

In a city scale graph with interconnected nodes, there are many possible paths between a source node (S) to a destination node (D). Considering all of those paths is computationally intractable and lots of them are not aligned with the agent's deadline and goal. Thus, we need to prune the search region in order to consider the paths with the closest characteristics to the desired path. For finding the candidate paths between a source (S) to a destination (D), we start from the source node and expand the connected nodes until we reach the destination. In expanding phase, we use a heuristic that for each node considers an approximate path from source to destination through that node, and if the mean of that path is greater than the provided deadline in query time, then the node is not expanded.

The path from source to destination through node N is the combination of the path from source to the node (P_{SN}) and the approximate shortest-length path from the node to destination (P_{ND}). For each node in expansion phase, P_{SN} is known from the history of previous expansion steps. For finding P_{ND}, we consider an approximate shortest-length path from that node to destination as finding the actual shortest-length path from N to destination is also computationally intractable due to the large branching factor in each step of the city scale graph.

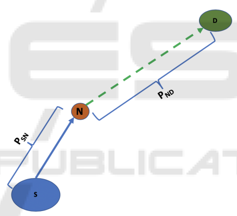


Figure 1: P_{SN} is the part of the path from source to node N and it is retrieved from the history of previous expansion steps. P_{ND} is the approximate shortest-length path from node N to destination. If the summation of mean of P_{SN} and P_{ND} is greater than the provided deadline, node N is not getting expanded.

For approximating the P_{ND}, we use city partitioning. Each partition includes a set of nodes and it is represented by its exemplars. Exemplar of each partition is one of the main nodes with highest traffic in that partition. For city partitioning, We tried few community detection methods on the graph of Salt Lake City (Infomap (Edler et al., 2017), Leading Eigenvector (Ruaridh Clark, 2018), Label propagation (Garza and Schaeffer, 2019), and Multilevel (Yang et al., 2016)) and among those Multilevel divides the city to 157 communities and in average each community includes 200 to 400 nodes in it. For the city like Salt Lake City, this distribution of nodes and number of partitions is reasonable. Figure 2 shows the distribution of communities of multilevel approach on the graph of Salt Lake City.

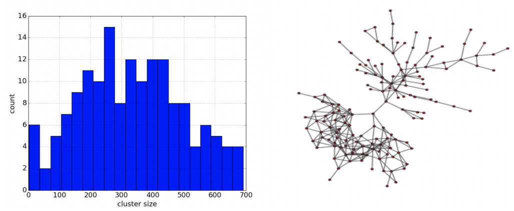


Figure 2: Left: Distribution of communities in Multilevel approach. Right: Visualization of communities on SLC graph. The dots represent communities.

Approximate shortest-length path (P_{ND}) is found by using A* algorithm on exemplars, i.e. instead of considering all the nodes from N to D, only exemplars are considered. In each step of A*, the next exemplar is picked based on the smallest $g(n)+h(n)$ value, where $g(n)$ is the shortest-length path from current exemplar to the neighboring exemplar and $h(n)$ is the direct path from the neighboring exemplar to the destination. Shortest-length paths between adjacent exemplars are pre-computed and they are retrieved to build the approximate shortest-length path.

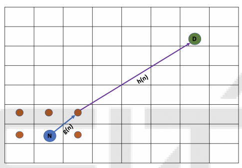


Figure 3: Finding an approximate shortest-length path from N to destination using $A*$ algorithm through exemplars.

As it can be seen from Figure 2, Salt Lake City has 157 partitions, therefore, pre-computing and storing the shortest-length paths between the adjacent exemplars is not a complex task. Also pre-computation of shortest-length path between exemplars is a one time task as the shortest-length path between exemplars doesn't change over time.

After finding the approximate shortest-length path, mean of both P_{SN} and P_{ND} are found considering the query time and if the summation of their means is greater than the provided deadline, the node is not expanded. This heuristic helps us to prune the path finding search region and find the potential paths with reasonable mean aligned with provided deadline.

As mentioned earlier, edge weights are represented based on mean and variance of the traffic flow on that edge at the query time. Also, each path is the finite sequence of edges. Therefore, paths from a specific source (C) to a destination (D) are presented as nodes (m_p,v_p) in the mean-variance plane (Figure 4). In the mean-variance plane, the horizontal axis represents the mean and the vertical axis represents the variance. Each small rectangle represents one of the candidate paths for a specific source, destination pair.

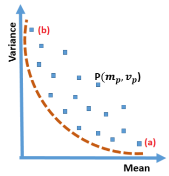


Figure 4: Paths from a specific origin (O) to a target (T) are presented as nodes (m_p,v_p) in the mean-variance plane.

Paths are in a convex hull and the best path is somewhere in the convex hull between the extreme points. Paths may vary from the one with highest variance and lowest mean (marked as b) to a path with highest mean and lowest variance (marked as a)in the mean variance domain shown in Figure 4. Convexity certifies that in the search region, there can be only one optimal solution which is globally optimal (Nikolova and Karger, 2008; Rus, 2020; Niknami and Samaranayake, 2016). Then based on the cost function and agent's goal, one of these paths is selected as the best path which we explain in further sections.

3.5 Cost Function

Since there may be more than one candidate path between two nodes and the main objective is to find a path with minimum expected cost, we need to have a function which models each path's expected cost which is found using Equation 7. Modelling cost function is extended from our previous work (Ahmadi and Allan, 2017)

$$ExpectedCost(t) = cost(t) * f_{path}(t) \tag{7}$$

For modelling paths' cost $Cost(t)$, we studied three main classes of cost functions: a) linear, b) exponential, and c) step cost functions and we discuss the characteristics of each one in the subsequent sections. Obviously, modelling paths' cost is not limited to the cost functions we discuss here and the model is general enough to handle any cost function of choice. Either by combining linear, exponential, and step function or by directly putting $Cost(t)$ in Equation 7.

Since the main cost on paths is travel time and travel time on edges is a continuous random variable which follows a normal distribution (Nikolova and Karger, 2008; Rus, 2020; Fan et al., 2005; Sigal et al., 1980), Probability Density Function (PDF) is used (Bachman et al., 2000) to define the probability of travel time random variable at each specific time (described in Equation 8). The parameter m_{path}, v_{path} are calculated based on Equation 3 and Equation 4.

$$f_{path}(t|m_{path}, v_{path}) = \frac{1}{\sqrt{2\pi v_{path}}} e^{-\frac{(t-m_{path})^2}{2v_{path}}} \tag{8}$$

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

3.5.1 Linear Cost Function

In the linear cost function model, the cost of the path increases linearly by travel time (Equation 9). The longer the travel time (t), the more expensive the path is. The expected cost is calculated using Equation 10.

$$cost(t) = t \tag{9}$$

$$ExpectedCost(t) = \int_{-\infty}^{+\infty} t f_{path}(t)dt = m_{path} \tag{10}$$

Therefore, if we model cost as linear, expected cost of the paths are equal to average travel time of those paths. In that case, neither deadline and nor agent's goal plays a role here. It even removes the effect of variance of travel time of paths.

3.5.2 Exponential Cost Function

Exponential cost function refers to the case where the cost of a path increases rapidly by travel time. Equation 11 shows the exponential cost model based on travel time (t), and Equation 12 is used for calculating the expected cost. In Equation 11, k is the steepness of the exponential cost increase.

$$cost(t) = e^{k*t} \tag{11}$$

$$ExpectedCost(t) = \int_{-\infty}^{+\infty} e^{k*t} f_{path}(t)dt = e^{k(m_{path}+\frac{v_{path}}{2})} \tag{12}$$

Based on the Equation 12, minimizing the expected cost of the exponential cost function depends on minimizing the linear combination of mean and variance in accordance with cost steepness parameter k.

Even though modelling cost as exponential considers the effect of variance in path planning, hence it always picks the path with minimum m_{path}, v_{path} at query time and other parameters such as deadline and agents' goals are not in the picture of decision making.

3.5.3 Step Cost Function

Another way of modeling the cost function is to penalize the paths which reach the destination after the deadline. In this case, a step function is used to model the cost (Equation 13 and Equation 14). In Equation 13, u represents a step function (Bachman et al., 2000), d stands for the desired arrival time, and t is travel time random variable. Then expected cost is found using Equation 15.

$$cost(t,d) = u(t-d) \tag{13}$$

$$u(t-d) = \begin{cases} 1 & if \quad t > d \\ 0 & if \quad t < d \end{cases} \tag{14}$$

$$ExpectedCost(t) = \int_{-\infty}^{+\infty} u(t-d) f_{path}(t)dt \tag{15}$$

Since the step function does not consider any penalty if the agent reaches the destination before deadline, the cost in the interval of $[-\infty, d]$ is zero and Equation 15 is re-written as Equation 16. Equation 16 is equal to Cumulative Density Function (CDF) of Standard Normal Distribution (Bachman et al., 2000). CDF generates a probability of the random variable (travel time in this case) when distribution is normal to be less than a specific value which is d (deadline) here. Based on Equation 16, when there is a set of paths from a specific origin to a destination, the path with minimum expected cost is the path that maximizes Equation 17.

$$ExpectedCost(t) = \int_{d}^{+\infty} f_{path}(t)dt = 1 - \Phi(\frac{d - m_p}{\sqrt{v_p}}) \tag{16}$$

$$\Phi(path) = \frac{deadline - m_{path}}{\sqrt{v_{path}}} \tag{17}$$

In the step cost model, in order to select the best path, we need to consider deadline, agents' goals and query time in the objective function as shown in Equation 17.

3.6 Modelling Agents' Goals

As mentioned in 3.3, two agents' goals have been considered in this work and per the discussion in 3.5, if we model paths' cost as linear and exponential, agents' goals are not considered in expected cost minimization. Therefore, we focus on step cost function as one of the possible cost functions which considers the agents' goals.

3.6.1 Highest Probability Path

If we model the cost as step function the expected cost can be found by Equation 16. Then, in order to minimize the expected cost we need to maximize Equation 17 which is the path with highest probability of reaching the destination before deadline. For finding the best path, we need to consider the set of all candidates paths from origin (O) to destination (D) in the mean-variance domain based on the approach explained in 3.4 in order to select the path which maximizes Equation 17.

3.6.2 Shortest En-route Time

For finding the shortest en-route time, cost function is modelled as the step cost. Then having the desired arrival time and the probability of making the trip before that, we are looking for the specific time t_G for departure which results in the shortest en-route time. Therefore, the departure time is not fixed and it is a specific time t_G bounded in the interval of query time $\tau 1$ and desired arrival time $\tau 2$. For simplicity of referral, we call the $[\tau 1, \tau 2]$ interval as the trip interval.

For this model, we modify the deadline variable in Equation 17 as the difference of desired arrival time and departure time. The goal is to find the departure time t_G in a way that the travel duration is minimized. In Equation 18, ϕ is the argument of the Cumulative Distribution Function (*CDF*) that makes the *CDF* equal to the given probability of making the trip before $\tau 2$ and it is fixed here.

$$\begin{aligned} desired\_arrival\_time - departure\_time = \\ m_{path} + \Phi(path) * \sqrt{v_{path}} \\ if \quad departure\_time \subset [\tau 1, \tau 2] \end{aligned} \quad (18)$$

Here is the summary of the steps need to be done for this scenario.

- Find the latest time (τ_L) that if agent departs it can still reaches destination before deadline.
- Divide trip interval $[\tau 1, \tau_L]$ to sub-intervals of $[t_1, t_2, t_3, t_4, t_5, \ldots, t_n]$ in accordance with time segment definition (10 minutes each).
- For each of the sub-intervals k which is $[t_{(k-1)}, t_k]$:
  - Find the paths from an origin (O) to destination (D) in a case that if they start their trip in $[t_{(k-1)}, t_k]$, they can make the trip before deadline.
  - From the set of paths found in last step, select the one which minimizes the expected cost of the objective function described in Equation 18.
- Now for each time segment k we have one path which is the best for that time segment. Then, select the interval which has the path with minimum travel time.

4 EXPERIMENTS AND RESULTS

4.1 Path Planning based on Agents' Goals

In this experiment, we study how agents' goals in path planning affect the paths selection in different times of the day. For this reason, we pick some source, destination pairs with the distance in the range of seven to ten miles.

4.1.1 Highest Probability Path

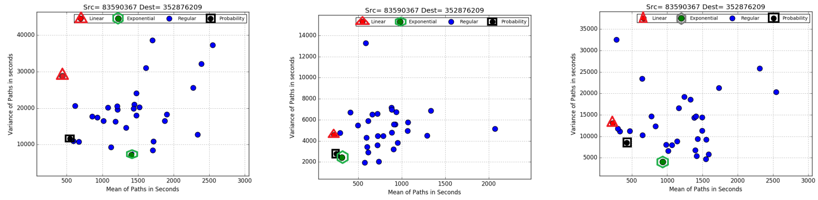
Given the deadline (set as the estimated travel time for the two selected nodes in this experiment) and the goals of selecting the path with highest probability of reaching destination, we aim to study the path selection for each of the introduced cost function (linear, exponential and step cost function).

Figure 5 and Figure 6 show the results of this experiment for three different time slots of Friday and Tuesday for two sets of randomly picked nodes. Each circle represents one of the possible paths between the source and destination. Considering the set of paths between a specific source and destination, a red triangle identifies the path that satisfies the linear cost function criteria. A green trapezoid is the best path based on the exponential cost model and a black rectangle identifies the path with step cost function.

The results in Figure 5 and Figure 6 show that the characteristics of paths for the same source and destination nodes changes in different times of the day depending on the impact of travel level throughout the city.

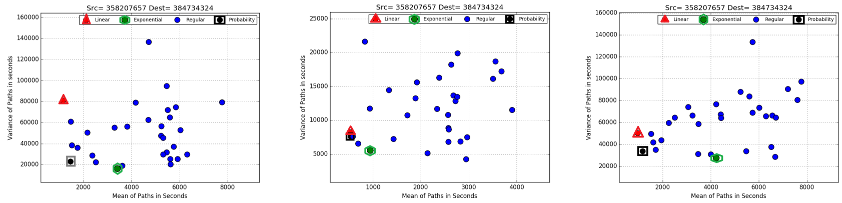
A pattern in both Figure 5 and Figure 6 shows that when we query for best paths in rush hours, the difference between linear, exponential and step cost function which models highest probability path is large. However, in non-rush hour times, there is not a significant difference between them. This means that having a realistic cost function model along with considering the deadline helps in finding better paths in rush hour. In non-peak times, since traffic is low, paths are similar to each other and navigation might not be that crucial. Having a good cost function modeling and a wise criteria of picking the best path is crucial when paths are congested.

Another interesting point is the way different models pick the best path. Linear model (red triangle) focuses on picking the path with minimum mean, while in some cases like Figure 5.(a) the path might have a high variance. Exponential model (green trapezoid) considers the mean and variance but it does not consider the deadline. Therefore, in Figure 5.(a), Figure 6.(a), and Figure 6.(c) the paths selected by exponential model all violate the deadline. Step cost function which demonstrates the highest probability path, pick the least risky path which makes the deadline without a high variance which is the desired outcome in this experiment.



(a) (b) (c)

Figure 5: Selected paths for times of Friday for Source node=83590367 and Destination node=352876209 for each cost function. Query times from left to right the times are: a) 8:00, b) 15:00, and c) 18:10 PM. Deadline is set as 1200 seconds.



(a) (b) (c)

Figure 6: Selected paths for different times of Tuesday for Source=358207657 and Destination=384734324 for each cost function. Query times from left to right are: a) 7:30, b) 11:40, and d) 17:45. Deadline is set as 1400 seconds.

4.1.2 Shortest En-route Time

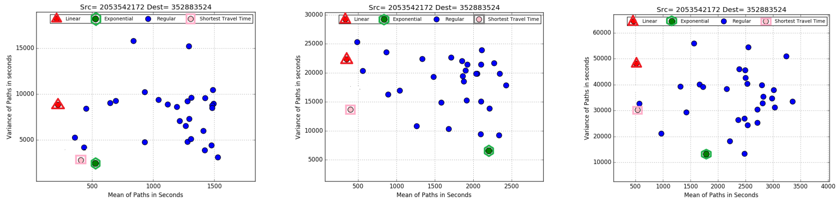
In this experiment, the agent wants to have the shortest en-route time within a desired arrival interval. We determine the best time to start the trip and which path yields the shortest travel time. Figure 7 and Figure 8, show the results of this experiment for three different time slots of Monday and Wednesday for two different sets of nodes with all three mentioned cost functions. Each circle represents one of the paths between the source and destination. In each figure, pink rectangle represents the path with smallest travel time with the step cost function, green trapezoid is the best path if the cost function is exponential, and red triangle is the best path based on linear cost function model.

Similar to the findings in the previous experiment, Figure 7 and Figure 8 shows that in rush hours selected paths for different models differ from each other significantly, while in non-peak hours they are almost the same. It emphasizes the effect of congestion in busy hours and how it changes the weights on edges of the graphs and impacts the path selection. As in the previous experiment, the linear model picks the path with smallest mean, while that path might have a high variance like Figure 7.(b). Exponential path does not consider deadline and it may pick a path which does not make the trip within the desired arrival. Step cost function which provides the shortest en-route time path considers mean, variance, and desired arrival time. Desired probability for this experiment is considered as 85 percent. This means that we are interested to find the paths that have the smallest travel time and within the chance of 85 percent can make the trip before deadline (85 percent is a number we picked to keep the experiments consistent here, it can be any probability).

Another finding indicates that picking the shortest en-route path sometimes is a risky decision as it has a high variance of reaching destination before deadline. For example, in Figure 7 and Figure 8 the smallest travel time path has the higher variance in comparison to the path with exponential cost function.

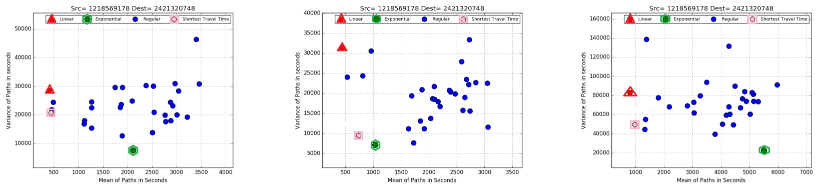
4.2 How Do Selected Paths Compare with Shortest-length Path?

In the realm of path planning, shortest-length path is always a practical option. Hence we used it as a baseline to see how our paths are different from the shortest-length path. In this experiment, we compare means and variances of shortest-length path with highest probability path and shortest en-route time path for 100 random pairs of source and destinations with the distance of 10 to 12 miles in different areas of Salt Lake City during morning rush hours. The results for each of the goals are averaged over all 100 pairs in weekdays (Monday through Friday). Desired travel time for the pairs is considered as 2200 seconds (based on the average time takes to get from a source to destination with the distance for 10 to 12 miles in rush hour) and desired probability for this experiment is considered as 85 percent. Figure 9 is the demonstration of means and variances for each day of the week. As it can be seen from the graphs, shortest-length path is not performing well as it just tries to pick the mini-



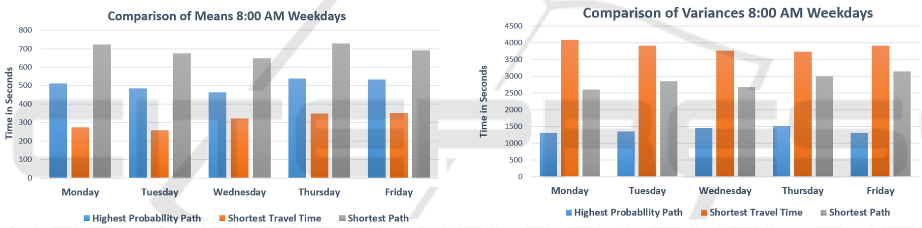
(a) (b) (c)

Figure 7: Selected paths for different times of Monday for Source=2053542172 and Destination=352883524. Times from left to right are: a) 6:40, b) 8:10, and d) 18:00. Desired arrival time is within the 1600 seconds after query time. Best time for start the trip to have the smallest travel time is as follow: a) 6:40, b) 8:23, and c) 18:21.



(a) (b) (c)

Figure 8: Selected paths for different times of Wednesday for Source=1218569178 and Destination=2421320748. Times from left to right are: a) 7:40, b) 11:10, and d) 17:30. Desired arrival time is within the 2000 seconds after query time. Best time for start the trip to have the smallest travel time is as follow: a) 8:01, b) 11:10, and c) 18:09.



(a) (b)

Figure 9: Comparison of average mean and variance of highest probability path, shortest en-route path and shortest-length path in 8:00 AM of Weekdays for 100 different source and destinations.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

mized length path even if it is congested. Shortest en-route time paths have smaller means in comparison to highest probability paths while they have higher variance which makes highest probability paths more secure options but longer travel time. Even though shortest-length paths have reasonable variance, their high mean value makes them not a good option to pick.

5 CONCLUSION AND FUTURE WORK

This research represents path planning in the real world domain in which edge weights are not fixed but are stochastically affected by the time of the day/week. Inspired by time-dependent traffic situation, we parameterize these distributions by time, which allows us to speak of time-dependent path costs and study the problems of reaching a goal by a deadline, and delaying departure to minimise traversal-to-goal time. The best path is the path with lowest cost and the cost is based on travel time which highly depends on the level of congestion in different time of the day/week. Since the graph is interconnected, optimizing for lowest cost on all possible paths is not feasible, therefore, we propose a pruning technique to shrink the search region. Agents can pursue two main goals: 1) picking the least risky path and 2) picking the smallest travel time along with awareness of when to start the trip.

Results show that during rush hour, utilizing an intelligent path planning approach is crucial. In addition, we demonstrate that a suitable path planning approach must consider path's mean, path's variance, agents' goals and the deadline to provide optimal options. We also compared the mean and variance of highest probability paths and shortest en-route paths with shortest-length paths at the same query time. This experiment proves that during rush hour shortest-length path is not a good option. Another finding indicates that, highest probability path has less variance as it takes the most secure path, while the smallest travel time has the lowest mean and might have a high variance.

Possible future work is to consider salable approaches that enables this framework to handle large number of path planning queries at the same time. Also, we can expand the model to offer paths that are optimal for alleviating the overall congestion of the city rather than just the best path for each agent (optimal decisions versus selfish decisions).

REFERENCES

Ahmadi, K. and Allan, V. H. (2017). Stochastic Path Finding under Congestion. In *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 135–140.

Bachman, G., Narici, L., and Beckenstein, E. (2000). *Fourier and Wavelet Analysis*. Universitext. Springer New York, New York, NY.

Campbell, A. M., Gendreau, M., and Thomas, B. W. (2011). The orienteering problem with stochastic travel and service times. *Annals of Operations Research*, 186(1):61–81.

Cao, Z. (2017). Maximizing the probability of arriving on time : a stochastic shortest path problem.

Chiabaut, N., Buisson, C., and Leclercq, L. (2009). Fundamental Diagram Estimation Through Passing Rate Measurements in Congestion. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):355–359.

Dijkstra, E. W. and W., E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Edler, D., Guedes, T., Zizka, A., Rosvall, M., and Antonelli, A. (2017). Infomap Bioregions: Interactive Mapping of Biogeographical Regions from Species Distributions. *Systematic Biology*, 66(2):197–204.

Fan, Y. Y., Kalaba, R. E., and Moore, J. E. (2005). Arriving on Time. *Journal of Optimization Theory and Applications*, 127(3):497–513.

Garza, S. E. and Schaeffer, S. E. (2019). Community detection with the Label Propagation Algorithm: A survey. *Physica A: Statistical Mechanics and its Applications*, 534:122058.

Geisberger, R., Kobitzsch, M., and Sanders, P. (2010). Route planning with flexible objective functions.

Haklay, M. and Weber, P. (2008). OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7(4):12–18.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

Lau, H. C., Yeoh, W., Varakantham, P., and Nguyen, D. T. (2012). Dynamic Stochastic Orienteering Problems for Risk-Aware Applications. *Uncertainty in Artificial Intelligence: Proceedings of the Twenty-Eighth Conference: August 15-17 2012, Catalina Island, United States*.

Long, D., ICAPS 2006 (16 2006.06.06-10 Windermere, U., and International Conference on Automated Planning and Scheduling (16 2006.06.06-10 Windermere, U. (2006). *Optimal route planning under uncertainty*. AAAI Press.

Miller-Hooks, E. D. and Mahmassani, H. S. (2000). Least Expected Time Paths in Stochastic, Time-Varying Transportation Networks. *Transportation Science*, 34(2):198–215.

Niknami, M. and Samaranayake, S. (2016). Tractable Pathfinding for the Stochastic On-Time Arrival Problem. pages 231–245. Springer, Cham.

Nikolova, E. and Karger, D. R. (2008). Route Planning under Uncertainty: The Canadian Traveller Problem.

Pi, X. and Qian, Z. S. (2017). A stochastic optimal control approach for real-time traffic routing considering demand uncertainties and travelers' choice heterogeneity. *Transportation Research Part B: Methodological*, 104(Supplement C):710 – 732.

Ruaridh Clark, M. M. (2018). Eigenvector-based community detection for identifying information hubs in neuronal networks | bioRxiv.

Rus, S. L. B. K. G. R. M. (2020). Method and apparatus for traffic-aware stochastic routing and navigation.

Sigal, C. E., Pritsker, A. A. B., and Solberg, J. J. (1980). The Stochastic Shortest Route Problem. *Operations Research*, 28:1122–1129.

Wilkie, D., Van Den Berg, J., Lin, M., and Manocha, D. (2011). Self-Aware Traffic Route Planning.

Yang, Z., Algesheimer, R., and Tessone, C. (2016). A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports*, 6.

Yaoxin Wu, Wei Chen, Xuexi Zhang, and Guangjun Liao (2016). Improving the performance of arrival on time in stochastic shortest path problem. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2346–2353. IEEE.