Optimal Routing in Stochastic Networks with Reliability Guarantees

Wanzheng Zheng[1], Pranay Thangeda[2], Yagiz Savas[3], and Melkior Ornik[2]

Abstract— **Optimal routing in highly congested street networks where the travel times are often stochastic is a challenging problem with significant practical interest. While most approaches to this problem use minimizing the expected travel time as the sole objective, such a solution is not always desired, especially when the variance of travel time is high. In this work, we pose the problem of finding a routing policy that minimizes the expected travel time under the hard constraint of retaining a specified probability of on-time arrival. Our approach to this problem models the stochastic travel time on each segment in the road network as a discrete random variable, thus translating the model of interest into a Markov decision process. Such a setting enables us to interpret the problem as a linear program. Our work also includes a case study on the street of Manhattan, New York where we constructed the model of travel times using real-world data, and employed our approach to generate optimal routing policies.**

I. Introduction

Optimal routing in transportation networks serves an essential role in multiple fields including emergency response and service industry [1]. Travel times on a road network are often stochastic due to unforeseeable incidents such as accidents, congestion, and pedestrian movement patterns [2]. In recognition of this stochasticity, navigation services generate a route between origin and destination locations with the *least expected travel time* (LET) [3]. However, LET as a metric of route selection does not consider traveler's tolerance to delay. In particular, emergency services or mass transit may have little tolerance for arriving late to their destination. For such services, it may be desirable to minimize the expected travel time *while guaranteeing* certain probability of arriving at destination in given time budget, hereinafter referred to as *route reliability* [4], [5].

While solutions to the problem of LET routing with reliability as a hard constraint have not been widely considered in the transportation community, various similar problems have been discussed. Frank [5] studied the problem of finding the path that maximizes the probability of realizing a travel time smaller than a specified time budget, namely the on-time arrival in stochastic network (SOTA) problem. Samaranayake and Blandin [6] improved on existing solutions to the SOTA problem and present an algorithm that accommodates both time-varying traffic conditions and spatio-temporal correlations of travel time distributions. Niknami and Samaranayake [7] provide a computationally efficient technique for the SOTA problem where the travel times belong to a general class of probability distributions. Nie and Wu [8] proposed algorithms to determine the latest possible departure time and the associated route for a SOTA problem. Note that above problems and discussions intend solely to maximize probability of on-time arrival and do not minimize travel time expectation. Such an approach might result in an overly cautious policy that chooses reliable, but unnecessarily long routes as optimal solution. A variant where reliability is considered as a soft constraint and included in the objective function with appropriate weight has been considered by Loui [9] and Thangeda [4]. Such a trade-off approach does not treat route reliability as a hard constraint, and it is not clear how to find the weight that will result in a policy that would be generated by a hard-constrained optimization problem. Another approximation of the problem was discussed by Murthy and Sarkar [10] using linear piecewise objective function to capture the nonlinear reliability constraints.

Xu and Mannor [11] proposed an algorithm for the general problem of achieving the best expected performance with the hard constraint of achieving some level of performance with a given probability. However, this work considers general Markov decision processes (MDP), with no emphasis on applications to the optimal routing problem. As we will show, the problem of optimal routing with reliability constraints can be interpreted within the framework of [11]. Such an interpretation results in a computationally feasible way to obtain the solution to the problem of interest even for large-scale transportation networks.

We represent the road network as a graph with intersections as nodes and roads as edges, as common in the literature [4], [7]. We augment the graph by an additional dimension to consider elapsed time. Given that travel times on the road networks are stochastic, the resulting product space yields an MDP, which enables us to appropriately plan an optimal routing policy with reliability constraints. Namely, once in the MDP framework, we use the flow equations to reformulate our problem as a linear programming (LP) problem. LP programs are well-studied and can be solved easily used efficient commercial solvers [12], [13]. We are thus, as shown by our numerical study on the streets of Manhattan, able to obtain an optimal, provably reliable path even in large road networks; our model contains more than 1000 intersections and more than 2000 road segments.

This work was supported by NASA's Space Technology Research Grants program for Early Stage Innovations under the grant "Safety-Constrained and Efficient Learning for Resilient Autonomous Space Systems."

[1]Wanzheng Zheng is with the Department of Aerospace Engineering, University of Illinois Urbana-Champaign, Urbana, USA. wz10@illinois.edu

[2]Pranay Thangeda and Melkior Ornik are with the Department of Aerospace Engineering and the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, USA. pranayt2@illinois.edu, mornik@illinois.edu

[3]Yagiz Savas is with the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, USA. yagiz.savas@utexas.edu

II. Preliminaries and Problem Statement

In this section we introduce the notation used throughout this paper and formally state the considered problem.

We model the road network using a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ are the graph vertices, and $\mathcal{E}$ the graph edges. Let n_v and n_e denote the number of vertices and the number of edges in the graph, respectively. Each intersection in the road network is represented by a vertex $v_i \in \mathcal{V}, i \in \{1, 2, \ldots, n_v\}$, and a road segment between any two intersections is represented by a directed edge $e_j \in \mathcal{E}, j \in \{1, 2, \ldots, n_e\}$. We also use the notation (v_i, v_j) to denote the edge heading from vertex v_i to vertex v_j.

We represent the stochastic travel times on edges as discrete random variables using a discrete time model, where one time step represents an appropriate unit of time. Assume that the longest possible travel time between any pair of directly connected vertices in the graph $\mathcal{G}$ is $N \in \mathbb{N}$, i.e., upon leaving any vertex $v_1 \in \mathcal{V}$ at time $t \in \mathbb{N}$, the agent reaches a directly connected vertex $v_2 \in \mathcal{V} \setminus \{v_1\}$ no later than $t + N$ with probability 1. For an edge $e \in \mathcal{E}$, let T_e denote the random variable representing the edge travel time, and for any $k \in \{1, 2, \ldots, N\}$, let $F(e, k)$ denote the probability of the travel time on the edge e being k; $\sum_{k=1}^{N} F(e, k) = 1$ for all $e \in \mathcal{E}$. While we consider time-invariant edge travel time distributions, our model, described in the next section, can accommodate time-varying distributions.

A *path* θ on the graph $\mathcal{G}$ is defined as a sequence of vertices $\theta = \{v_1, v_2, \ldots, v_k\}$ where $k \leq n_v$. Let π denote a *policy* defined as a function that, based on current time and vertex, (possibly non-deterministically) determines the next vertex to visit. Let Θ and Π denote the set of all possible paths and the set of admissible policies, and for any $\theta \in \Theta$, denote the total travel time along a path by T_θ. The value T_θ can then be expressed as the sum of the random variables representing the travel time on the edges constituting the path, i.e., $T_\theta = T_{(v_1,v_2)} + T_{(v_2,v_3)} + \cdots T_{(v_{k-1},v_k)}$.

Given an origin vertex $v_o \in V$ and a destination vertex $v_d \in V \setminus \{v_o\}$, let T_{od} be the random variable denoting the travel time from v_o to v_d. Considering a *travel time budget* $\tau_b \in \mathbb{N}$ and a reliability level of $\gamma \in (0, 1)$, we aim to find a policy $\pi^\star \in \Pi$ such that

$$\pi^\star = \arg\min_{\pi \in \Pi} \mathbb{E}[T_{od}] \tag{1a}$$

$$\text{s.t. } \Pr(T_{od} \leqslant \tau_b) \geqslant \gamma. \tag{1b}$$

When the travel times are deterministic, standard shortest path algorithms can be used to minimize the travel time. For stochastic travel times, algorithms for solving LET problems can be used to minimize expected travel time [5]. However, to the best of the authors' knowledge, there has been no prior work on providing solutions to (1). An obvious solution is one that considers all possible policies, but because of its computational complexity such a solution cannot be implemented in dense urban areas with thousands of vertices and edges. In the next section, we provide a solution to (1) by converting it to a linear programming problem on a Markov decision process.

III. Solution Approach

In this section, we discuss our approach to finding the optimal path by modeling the road network and the stochastic travel times using a Markov decision process (MDP) and then obtaining a solution by solving its corresponding linear programming formulation. We begin by describing how we represent the direct graph $\mathcal{G}$ with stochastic edge travel times as an MDP with stochastic state transitions.

A. Markov Decision Process Formulation

A Markov decision process [14], [15] $\mathcal{M}$ is specified by the tuple $(\mathcal{S}, \mathcal{A}, P, s_0)$ where $\mathcal{S}$ represents a finite set of states, $\mathcal{A}$ represents a finite set of actions with $\mathcal{A}(s) \subset \mathcal{A}$ denoting the available actions at a state $s \in \mathcal{S}$. Function $P : \cup_{s \in \mathcal{S}} \{s\} \times \mathcal{A}(s) \times \mathcal{S} \to [0, 1]$ represents a transition probability function between states such that $\sum_{s' \in \mathcal{S}} P(s, a, s') = 1$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$. Finally, $s_0 \in \mathcal{S}$ denotes the initial state.

A *stationary stochastic policy* π for the MDP $\mathcal{M}$ is a mapping $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that specifies the probabilities of taking different actions at any state, i.e., formally π at any state s is a probability distribution over $\mathcal{A}(s)$, $\sum_{a \in \mathcal{A}(s)} \pi(a|s) = 1$. The directed graph $\mathcal{G}$ and the origin vertex v_o, where the agent starts, can be represented as an MDP where the set of vertices becomes the state space $\mathcal{S} = \mathcal{V}$, the origin vertex becomes the initial state $s_o = v_o$, and the edges become the actions $\mathcal{A} = \{a_{(s,s')} : s, s' \in \mathcal{S}\}$. In order to simplify notation, when the agent is at a particular state s, we use $a_{s'}$ to denote the action $a_{(s,s')}$, i.e., $\mathcal{A}(s) = \{a_{s'} : s' \in \mathcal{S}\}$. Intuitively, we associate an action $a_{s'}$ with each state (vertex) $s' \in \mathcal{S}$, and if the agent takes the action $a_{s'}$ from the state $s \in \mathcal{S}$, then it transitions from s to s' with a probability 1 if the graph has the edge (s, s'). Formally, an agent taking the action $a_{s'}$ in the state s will have

$$P(s, a_{s'}, s') = \begin{cases} 1 & \text{if } \ (s, s') \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

The transition dynamics in $\mathcal{M}$ are deterministic as the outcome of every action in $\mathcal{A}$ is deterministic. In order to incorporate the stochastic edge travel times in this model, we include time as an additional dimension to the MDP. Specifically, let $\mathcal{M}' = (\mathcal{S}', \mathcal{A}, P', s_0')$ be an *extended MDP* with $\mathcal{S}' = (\mathcal{S} \times \{0, 1, \ldots, \tau_b, \delta\})$. The array $\{1, 2, \ldots, \tau_b\}$ represents samples with an arbitrary time resolution that we discretize our travel time data in and δ denotes "budget exceeded time" such that all actions leading to some state s' in time beyond τ_b will arrive in state (s', δ). For $(v, t) \in \mathcal{S}'$, if $t \in \{\tau_b, \delta\}$ or $v = v_d$, $P'((v, t), a, (v, t)) = 1$ for all $a \in \mathcal{A}$, and otherwise,

$$P'((v, t), a_{v'}, (v', t')) = \begin{cases} F((v, v'), t' - t) & \text{if } \ t' \leq \tau_b \\ \sum_{k: k+t > \tau_b} F((v, v'), k) & \text{if } \ t' = \delta \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Intuitively, if the agent takes the action v' from state (v, t), it transitions to (v', t') with probability $F((v, v'), t' - t)$ since the travel time associated with the edge (v, v') is $t' - t$ with probability $F((v, v'), t' - t)$. If there is a nonzero probability that the agent exceeds the allowed travel time τ_b by taking the action v', then all the excess flow goes to the absorbing state (v', δ). Similarly, the states (v, τ_b) and (v_d, t) are absorbing so that we can interpret (1) as a linear program.

Fig. 1 provides a simple example to illustrate our joint representation of the directed graph and stochastic travel times as an MDP. We consider a graph with vertices v_1, v_2 and v_3 and two edges, $e_1 = (v_1, v_2)$ and $e_2 = (v_1, v_3)$. We assume that the agent starts at the vertex v_1. Let the travel time on edge e_1 be a deterministic value of 10 seconds. On edge e_2, let the travel time be 10 seconds with a probability of 0.6 and 20 seconds with a probability of 0.4. Time resolution and time budget are chosen as 10 seconds and 20 seconds, respectively, i.e., the state space in the extended MDP will then contain states $\{\{v_1, v_2, v_3\} \times \{0, 10, 20, \delta\}\}$ and $\mathcal{A}(v_1)$ will contain the actions $\{a_2, a_3\}$ corresponding to the two edges e_1 and e_2. Let π denote a stochastic stationary policy defined as follows: $\pi(a_3|(v_1, 0)) = 0.9$ and $\pi(a_2|(v_1, 0)) = 0.1$. If an agent starts in the vertex v_1 at time $t = 0$ and follows the policy π, it travels along edge e_1 to the vertex v_3 with a probability of 0.9. If it indeed travels along the edge e_1, it could reach v_3 at $t = 10$ with a probability of 0.6 and at $t = 20$ with a probability of 0.4. The exact relationship is captured by the MDP as shown in Fig. 1 which illustrated the states reachable from $(v_1, 0)$.

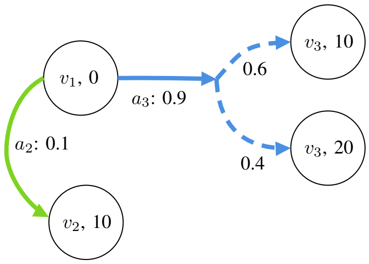

Fig. 1. Illustration of the MDP representation of a road network graph with three vertices and two directed edges.

With a better understanding of the extended MDP's structure, we define a cost function $C : \mathcal{S}' \times \mathcal{A} \rightarrow \mathbb{N}$ such that

$$C((v, t), a_{v'}) = \sum_{k=1}^{N} kF((v, v'), k). \tag{3}$$

Such a cost function associates each state-action pair in the extended MDP $\mathcal{M}'$ with an *expected travel time*. For $s \in \mathcal{S}'$, let $\Pr^{\pi}(\mathit{Reach}[s])$ denote the probability of reaching the state s from the initial state s'_0 under the policy π. By construction, the problem in (1) is equivalent to the following problem: find a policy $\pi^\star$ such that

$$\pi^\star = \arg\min_{\pi \in \Pi} \; \mathbb{E}\left[\sum_{(v,t)\in\theta} C((v, t), \pi(v, t)) \middle| (v, 0) = (v_o, 0) \right] \tag{4a}$$

$$\text{s.t.} \quad \sum_{t\in\{1,2,\ldots,\tau_b\}} \Pr^{\pi}\Big(\mathit{Reach}[(v_d, t)]\Big) \geq \gamma \tag{4b}$$

where θ is a path obtained from v_o to v_d by following the stochastic policy π. We now proceed to obtain the equivalent linear program formulation for (4).

B. Linear Programming

Let $\mathbf{s} = (v, t) \in \mathcal{S}'$ and let $\mathcal{B} = \{(v, t) : t \in \{\tau_b, \delta\}\}$ denote the set of absorbing states in the extended MDP $\mathcal{M}'$. The problem (4a)-(4b), and consequently our original problem stated in Section II, can be expressed as the following linear programming problem [16], [17]:

$$\min_{x(\mathbf{s},a)} \sum_{\mathbf{s}\in\mathcal{S}'\setminus\mathcal{B}} \sum_{a\in\mathcal{A}} x(\mathbf{s}, a)C(\mathbf{s}, a) \tag{5a}$$

$$\text{s.t.} \sum_{a\in\mathcal{A}} x(\mathbf{s}, a) - \sum_{\mathbf{s}\in\mathcal{S}'\setminus\mathcal{B}} \sum_{a\in\mathcal{A}} P'(\mathbf{s}', a, \mathbf{s})x(\mathbf{s}', a) = \alpha(\mathbf{s}) \quad \forall \mathbf{s} \in \mathcal{S}'\setminus\mathcal{B}, \tag{5b}$$

$$\sum_{t\in\{1,2,\ldots,\tau_b\}} \sum_{\mathbf{s}'\in\mathcal{S}'\setminus\mathcal{B}} \sum_{a\in\mathcal{A}} P'(\mathbf{s}', a, (v_d, t))x(\mathbf{s}', a) \geqslant \gamma \tag{5c}$$

$$x(\mathbf{s}, a) \geqslant 0 \quad \forall \mathbf{s} \in \mathcal{S}' \setminus \mathcal{B}, \quad a \in \mathcal{A}(\mathbf{s}). \tag{5d}$$

In the above problem, $\{x(\mathbf{s}, a) \geqslant 0 : \mathbf{s} \in \mathcal{S}'\setminus\mathcal{B}, a \in \mathcal{A}(\mathbf{s})\}$ are the decision variables. The variable $x(\mathbf{s}, a)$ represents the *expected number of times that the agent will visit a certain state* $\mathbf{s}$ *and takes an action* a. The function $\alpha : S' \rightarrow \{0, 1\}$ represents flow in each state such that $\alpha(\mathbf{s}'_0) = 1$ and $\alpha(\mathbf{s}) = 0$ otherwise. Note that problem (5) does not include any variables from absorbing state as path ends once the agent reaches $\mathbf{s} \in \mathcal{B}$.

The objective function in (5a) corresponds to the expected travel time until the agent reaches an absorbing state $\mathbf{s} \in \mathcal{B}$. The constraint in (5b), usually referred to as the *flow constraint* [18], ensures that for any state other than initial, $(v_o, 0)$, the number of times the agent enters a non-absorbing state is equal to the number of times the agent leaves that state. The constraint in equation (5c) ensures that the number of times the agent reaches any of the states $\{(v_d, t) : t \leq \tau_b\}$ is greater than or equal to reliability requirement γ. In general, the frequency of visiting a state is not necessarily equal to the probability of reaching that state. However, since the extended MDP has no loops, and the agent cannot return to states related to the same vertex once it leaves, the probability of reaching a non-absorbing state becomes equal to the expected number of times agent visits that state. Finally, the constraint in (5d) ensures that the expected number of visit to every state is non-negative. Linear programs such as (5) are well-studied and several commercial solvers exist that can efficiently solve problems with millions of variables in the matter of a few minutes [12] [13]. Once we obtain the LP formulation, we can solve for the optimal solution using commercial solvers such as Gurobi [12].

C. Policy Synthesis

Recall from (4) that our objective is to find an optimal policy $\pi^\star$. Given the optimal values of the decision variables in the linear program that represent the expected number of times an agent will take a certain action, the probability of taking an action under optimal policy $\pi^\star$ can therefore be given as

$$\pi^\star(a|\mathbf{s}) = \begin{cases} \frac{x(\mathbf{s},a)}{\sum_{a\in\mathcal{A}} x(\mathbf{s},a)} & \text{if} \quad \sum_{a\in\mathcal{A}} x(\mathbf{s},a) > 0 \\ \frac{1}{|\mathcal{A}(\mathbf{s})|} & \text{otherwise,} \end{cases} \tag{6}$$

where the notation $\pi^\star(a|\mathbf{s})$ denotes the probability with which the agent takes the action a from state $\mathbf{s}$.

IV. Network Model

This section discusses some of the steps involved in creating the extended MDP from road network geometry and traffic flow data. As described in Section III, we model the travel time on each edge as a discrete random variable. In order to obtain the probability mass function for these random variables, we use raw data commonly available in the form of vehicle travel times or mean travel speeds on that edge.

Given the maximum travel time N on an edge, we discretize the time duration of interest, 0 to N, into n_b bins, hereinafter referred to as *time buckets*, each with a size of $\varepsilon = N/n_b$. The time buckets $\{b_1, b_2, \ldots, b_{n_b}\}$ form the support of the travel time random variable for the edge. The available data on travel time is then organized into these buckets where all the samples with travel time between $(k-1)\varepsilon$ and $k\varepsilon$ are counted in the bucket b_k. The relative frequency of occurrences in each bucket is used to calculate the probability mass function $F(e,k)$. The values of ε and k are decided based on how the travel time is distributed on different edges and available memory and computational capabilities.

Given the probability mass function F, the transition probability function P' in the extended MDP and the cost function C can be calculated using (2) and (3). Using the extended MDP and the cost function, we can formulate the linear program as specified in (5) and obtain the optimal values of the decision variables to calculate the optimal stochastic policy using (6).

Fig. 2 illustrates an example of a road network with five intersections where travel times between intersections are assumed to be same in both directions. In this example, road segments between intersections 3 and 4, 3 and 5, and 4 and 5 are within a region of construction and will occasionally be blocked by heavy machinery. Travel times on all other edges are set to be deterministic for simplicity. Travel times and associated probabilities of each edge are given in Fig. 2, written as $(t_e, Pr(t_e))$ where t_e is a possible edge travel time. To be consistent with the assumption that the realized travel times on different edges are independent, the unexpected incidents are assumed to be independent: for instance, a road blockage due to heavy machinery operating in road segment between 3 and 4 does not correlate to road blockage in road segments between 3 and 5 or 4 and 5. If an agent wants to travel from intersection 1 to 5, a list of some attainable states and their corresponding available actions are visualized in Fig. 3. We will revisit this example in the following section to demonstrate the optimal solution.

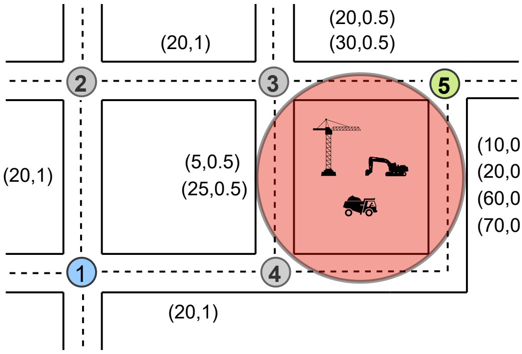

Fig. 2. An illustration of the considered construction site example.

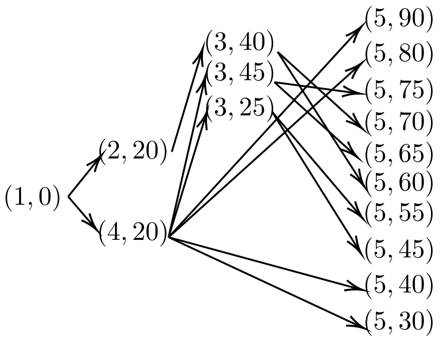

Fig. 3. Attainable state-action pairs in the simple example.

V. Examples and Verification

In this section, we present two examples that serve the purpose of verifying our approach and validating its utility on real-world data. First, we continue with the same example as above to validate the proposed solution approach. Once validated, we use our approach to solve the same problem the traffic network of Manhattan, New York City. This example serves to benchmark the performance of proposed approach.

In the algorithm implementation, the state-action pairs, as well as the selected origin, destination, and time bucket sizes are built into an extended MDP. Optimization problem (5) is constructed using this MDP and passed to Gurobi solver using its Python API. The solver then returns optimal values of decision variables $x^\star(\mathbf{s}, a)$. The optimal policy $\pi^\star$ is synthesized from the optimal values of decision variables returned by the solver using (6).

A. Construction Site Example Validation

We consider the following scenario for validating our approach: an agent wishes to travel from intersection 1 to intersection 5 in the graph illustrated in Fig. 2, with at least a 75% probability of arriving within a 70-unit time budget.

By examining the resulting extended MDP, we note that path 1-4-5 has the least expected travel time. Indeed, this is the path chosen using shortest path algorithms such as Dijkstra's algorithm using expected travel time values. However, such a path does not satisfy reliability constraint, as the agent only has 60% chance of arriving within 70 units of time. SOTA-py [7] generates a path that is guaranteed to satisfy the reliability requirement without any guarantees on the expected travel time. This approach yields the path 1-2-3-5 which guarantees that the agent will arrive to its destination within the time budget 100% of the time but is sub-optimal in terms of the travel time.

Unlike the LET path or the path produced SOTA-py, path 1-4-3-5 satisfies the specified reliability requirement *and* has a lower expected travel time than SOTA-py approach. This example shows that both LET and highest-reliability approaches are not best suited to solve the problem of finding path with minimum expected time under hard reliability constraints. Our approach indeed yields this path.

To compare our approach with approaches that optimize a weighted sum of different objectives, we solve the example using a variant of PROTRIP proposed in [4]. PROTRIP considers an objective function that uses a weighting factor α to combine the objectives of minimizing expected travel time and maximizing the reliability. A high value of α will give more importance to the objective of minimizing the expected travel time and vice versa. The results obtained for different values of α are presented in Table I. We observe that while one can obtain the desired optimal path by tweaking the hyperparameter α, it is often not obvious what the correct choice of α is to obtain the desired result. On the other hand, our approach does not need any tuning to obtain the optimal solution. Table I summarizes the solutions obtained from the different possible approaches discussed so far.

TABLE I
COMPARISON OF SOLUTIONS USING DIFFERENT ALGORITHMS

Algorithms	Solution	Travel Time	Reliability
LET approach	1-4-5	55	60%
SOTA-py	1-2-3-5	65	100%
PROTRIP $\alpha = 0.3$	1-2-3-5	65	100%
PROTRIP $\alpha = 0.4$	1-4-3-5	60	75%
Our approach	1-4-3-5	60	75%

We now move on to a more realistic and computationally intensive example.

B. Case Study: Navigation in Manhattan

In this section we consider the road network in a section of Manhattan, New York City, an urban area known for its high vehicle density and traffic variability [19]. The streets in Manhattan area follow a grid structure with most streets allowing only one-directional flow of traffic. We consider the area of Manhattan bounded by 2nd Avenue from the east, 11th Avenue/ West End Avenue from the west, 42nd Street from the south, and 116th Street from the north. The graph representing this area consists of 1024 vertices and 2118 edges.

The travel time on each edge is stochastic due to the high variance of travel time in the congested streets of Manhattan, especially during rush hour. In order to obtain a realistic model of the true travel time distribution on each edge, we use real-world speed data on each edge obtained from open source databases [20]. The data, available in the form of mean and variance of speed on each edge, is used to generate a speed distribution on each edge by assuming that the vehicular speed follows a lognormal distribution. This assumption is supported by the literature [21], which suggests that the vehicular speed in highly dense urban streets indeed follows a lognormal distribution. Using the speed distributions and data on lengths of road segments, we find the corresponding parameters of the lognormal travel time distribution on each edge. Fig. 4 shows the spatial distribution of travel time mean and standard deviation on different edges in the area of interest. While we consider static speed data in this example, our approach can also accommodate time-varying speed data.

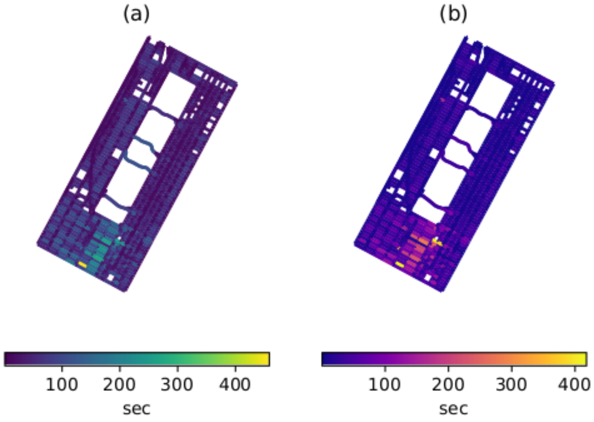

Fig. 4. Mean and standard deviation of travel times on different road segments in Manhattan, New York City.

From Fig. 4, it is clear that the travel time variance is highest in Midtown Manhattan, near the bottom of the map. We consider origin and destination vertices located on either side of this region for our analysis. We simulate two scenarios, one with 60% probability and another with 90% probability of arriving within the time budget, which correspond to $\gamma = 0.6$ and $\gamma = 0.9$ respectively. Using the proposed methodology, we obtain optimal policies for each of these scenarios. Note that the optimal policy is not deterministic, i.e., when an agent arrives at a particular intersection at some point in time, the policy does not specify a unique next vertex to reach. Instead, the policy provides a probability distribution over the next possible vertices, and the agent samples from this distribution to decide the next vertex. Figure 5 shows the origin and destination vertices and two instances of routes obtained by using the synthesized optimal policies for different reliability requirements.

Table II shows the mean travel time and reliability data for simulated trajectories obtained by following the optimal policies generated by our algorithm for different reliability

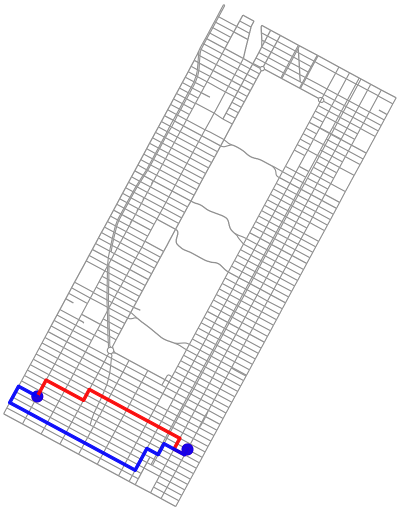

Fig. 5. Map showing the instances of routes obtained using optimal policies obtained for different reliability requirements for the same origin-destination pair. The red line represents one route obtained with $\gamma = 0.6$ and the blue line represents one route obtained with $\gamma = 0.9$.

TABLE II

COMPARISON OF ROUTES WITH DIFFERENT RELIABILITY REQUIREMENTS

Scenario	Travel Time	Reliability
$\gamma = 0.6$	620 sec	0.65
$\gamma = 0.9$	1460 sec	0.92

requirements with a time budget of 1800 seconds, i.e, 30 minutes. Clearly, the path with a lower reliability requirement ($\gamma = 0.6$) has a shorter expected travel time in this case. However, when a higher reliability requirement is enforced ($\gamma = 0.9$), the policy tries to take a path with more predictable travel time, even if it results in a higher expected travel time.

VI. Conclusion and Future Work

In this paper we presented a solution approach for finding the policy that results in minimal expected travel time while also satisfying a user-defined on-time arrival probability requirement for a given time budget. In order to solve the problem of interest, we interpreted the road network and associated travel times as an extended Markov decision process (MDP). The translation is founded on understanding node-time information as abstract states, and road selection between states as actions. The problem of optimal routing can then be framed as a linear programming problem which can be solved using an off-the-shelf optimizer. We validated and illustrated our approach on two examples: a simple five-node graph and the road network of Manhattan, the later demonstrating the versatility and performance of the solver in real life scenarios.

Motivated by the application of the proposed approach to emergency and delivery services, where reliability plays a significant role, a natural avenue of future work includes proposing and solving a facility location problem that minimizes the expected time of service while maintaining confidence in maximum service time. Another possible area of future work is optimal planning of fixed routes for mass transit to conveniently, efficiently, and reliably serve the population. Relevant further study also includes investigating the impact of road closures and other significant unexpected changes in network geometry and flow on route reliability.

Acknowledgment

The data for the case study was retrieved from Uber Movement, (c) 2021 Uber Technologies, Inc.

References

[1] G. Feng, G. Su, and Z. Sun, "Optimal route of emergency resource scheduling based on GIS," *3rd ACM SIGSPATIAL Workshop*, no. 6, 2017.

[2] R. J. Javid and R. J. Javid, "A framework for travel time variability analysis using urban traffic incident data," *IATSS Research*, vol. 42, no. 1, pp. 30–38, 2018.

[3] R. W. Hall, "The fastest path through a network with random time dependent travel times," *Transportation Science*, vol. 20, no. 3, pp. 143–221, 1986.

[4] P. Thangeda and M. Ornik, "PROTRIP: Probabilistic risk-aware optimal transit planner," *23rd International Conference on Intelligent Transportation Systems*, 2020.

[5] H. Frank, "Shortest paths in probabilistic graphs," *Operation Research*, vol. 17, no. 4, pp. 565–759, 1969.

[6] S. Samaranayake, S. Blandin, and A. Bayen, "A tractable class of algorithms for reliable routing in stochastic networks," *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 199–217, 2021.

[7] M. Niknami and S. Samaranayake, "Tractable pathfinding for the stochastic on-time arrival problem," *15th International Symposium on Experimental Algorithms*, pp. 231–245, 2016.

[8] Y. Nie and X. Wu, "Shortest path problem considering on-time arrival probability," *Transportation Research Part B: Methodological*, vol. 43, no. 6, pp. 597–613, 2009.

[9] R. Loui, "Optimal paths in graphs with stochastic or multidimensional weights," *Communications of the ACM*, vol. 26, no. 9, pp. 670—676, 1983.

[10] I. Murthy and S. Sarkar, "Stochastic shortest path problems with piecewise-linear concave utility functions," *Management Science*, vol. 44, no. 11, pp. 125–136, 1998.

[11] H. Xu and S. Mannor, "Probabilistic goal Markov decision processes," in *22nd International Joint Conference on Artificial Intelligence*, 2011, pp. 2046–2052.

[12] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2021. [Online]. Available: http://www.gurobi.com

[13] IBM Corp., "CPLEX user's manual," 2021. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer

[14] R. Bellman, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954.

[15] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[16] K. Etessami, M. Kwiatkowska, M. Y. Vardi, and M. Yannakakis, "Multi-objective model checking of Markov decision processes," in *13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2007, pp. 50–65.

[17] E. Altman, *Constrained Markov Decision Process*. CRC Press, 1999.

[18] P. Toth and D. Vigo, *The Vehicle Routing Problem Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics, 2002.

[19] A. Yazici, E. E. Ozguven, and A. Kocatepe, "Urban travel time variability in New York City: A spatio-temporal analysis within congestion pricing context," *Transportation Research Board 96th Annual Meeting*, 2017.

[20] Uber Technologies, Inc., "Uber movement," 2021. [Online]. Available: https://movement.uber.com

[21] H. Rakha, I. El-Shawarby, and M. Arafeh, "Trip travel-time reliability: issues and proposed solutions," *Journal of Intelligent Transportation Systems*, vol. 14, no. 4, pp. 232–250, 2010.