Networks and Spatial Economics
https://doi.org/10.1007/s11067-021-09545-6

# Adaptive Park-and-ride Choice on Time-dependent Stochastic Multimodal Transportation Network

**Pramesh Kumar**[1] · **Alireza Khani**[1]

Accepted: 17 May 2021 Published online: 17 July 2021


## Abstract
In transportation networks with stochastic and dynamic travel times, park-and-ride decisions are often made adaptively considering the realized state of traffic. That is, users continue driving towards their destination if the congestion level is low, but may consider taking transit when the congestion level is high. This adaptive behavior determines whether and where people park-and-ride. We propose to use a Markov decision process to model the problem of commuters' adaptive park-and-ride choice behavior in a transportation network with time-dependent and stochastic link travel times. The model evaluates a routing policy by minimizing the expected cost of travel that leverages the online information about the travel time on outgoing links in making park-and-ride decisions. We provide a case study of park-and-ride facilities located on freeway I-394 in Twin Cities, Minnesota. The results show a significant improvement in the travel time by the use of park-and-ride during congested conditions. It also reveals the time of departure, the state of the traffic, and the location from where park-and-ride becomes an attractive option to the commuters. Finally, we show the benefit of using online routing in comparison to an offline routing algorithm.

**Keywords** Transit · Park-and-ride · Adaptive route choice · Stochastic shortest path · Recourse · Online shortest path

✉ Pramesh Kumar
kumar372@umn.edu

Alireza Khani
akhani@umn.edu

[1] Department of Civil, Environmental, and Geo-Engineering, University of Minnesota - Twin Cities, 500 Pillsbury Drive S.E., Minneapolis, MN, 55455, USA

## 1 Introduction

Due to a rise in the number of motor vehicles resulting from increasing travel demand, urban highways are facing an inescapable condition of traffic congestion. This increase in demand has led to a rise in the level of fuel consumption, greenhouse gas emissions, travel delays, accidents, and thereby a negative impact on the quality of life in cities (Shepardson and Carey 2018). It has been observed that the congestion problem is more dominant in the Central Business District (CBD) areas where people commute every day for work or school. To avoid this congestion, travelers often consider using park-and-ride mode i.e., they drive up to a certain stretch of the highway, and then take transit to reach their destination. The routing of park-and-ride users is different from transit and auto users as travelers need to decide where to park and take transit. These decisions are often made adaptively considering the realized state of traffic and transit availability.

Modeling park-and-ride location choice of commuters in a congested urban network is inherently a difficult problem. Various factors such as accessibility, parking availability at park-and-ride locations, travel time perception of users, or trip costs affect the decision process (Pang and Khani 2018; Webb and Khani 2020). Besides these, transportation networks are often subject to random arc travel times. This variability can be attributed to several factors such as congested road conditions (e.g., due to induced demand), traffic signals, accidents, bad weather, construction disruptions, and so on. Due to this variability in travel time, the park-and-ride location choice becomes even more complex. Recent advances in intelligent transportation systems such as Variable-message sign (VMS) on roads or Automatic Vehicle Location (AVL) technology employed in transit vehicles allow us to gather real-time information about the current state of the transportation network. However, most routing applications do not consider this uncertainty while providing routing policies. The adaptive routing policy can help users in making better choices and improve the overall travel cost by re-evaluating the auto route, park-and-ride location, and transit itinerary.

In the United States, several freeways with and without High-Occupancy Toll (HOT) lane connect the residential suburbs to CBD, where many people commute for work and education. Multiple express bus routes connect the suburban areas to the CBD along these facilities, with relatively high speed but low frequency. These routes are mostly supplemented by park-and-ride facilities alongside the freeways and give commuters the choice to transfer to transit mode at different points. On less congested days/times, one may continue driving to the destination, while on more congested days/times, parking at a park-and-ride location at a midway point and taking transit for the remaining part of the trip can lead to shorter travel time or cost. When real-time information on congestion level and waiting time of transit routes are provided to users using variable message signs along the freeway corridor and routing applications respectively, users can change their decisions *en route* considering current congestion level and short term traffic predictions. Under these conditions, understanding and modeling users' park-and-ride decisions become a difficult but interesting research question.

In this study, we model the adaptive routing of a park-and-ride user trip in a stochastic and time-varying (STV) transportation network. We assume that the arc travel time in the auto network and wait time for buses at any park-and-ride location is a time-dependent random variable. Travelers receive information *en-route* when arriving at each node. This information includes the travel time on the downstream arcs, and wait time of buses if they reach a park-and-ride node which helps them to assess the network conditions, and decide whether to park-and-ride or continue their journey on the freeway. We model this adaptive routing problem as a *stochastic shortest path* (SSP) problem. The states are defined by a tuple of node, time, and the online information received. The objective is to minimize the expected generalized cost of travel which includes travel time on road and parking cost if the auto option is chosen, otherwise, the road travel time, transit travel time, and transit fare if park-and-ride is chosen. Through this article, we make the following contributions:

- Modeling adaptive park-and-ride routing as a stochastic shortest path problem
- Formulating value iteration and label correcting solution methods for the current problem
- Evaluating optimal policy by minimizing expected cost of travel of a park-and-ride trip
- Implementing the current methodology to I-394 in Twin Cities, MN

The rest of the article is structured as follows. Section 2 presents the related work on adaptive routing in stochastic transportation networks. Then, Section 3 describes the problem with an example, which is followed by Section 4 introducing the problem definitions and various notations used in this article. Section 5 formulates the park-and-ride decision problem and describes various solution algorithms to solve it. Then, Section 6 presents a case study, and finally, the conclusions and directions for future research are presented in Sections 7 and 8 respectively. The proofs of various propositions are provided in Appendix B.

## 2 Related Work

There has been a considerable amount of work on the shortest route planning in the literature. It consists of route planning for different modes of transport such as auto (Ahuja et al. 1988), transit (Tong and Richardson 1984b; Khani et al. 2015; Kumar and Khani 2021), and park-and-ride (Khani et al. 2012). These route planning algorithms can be classified into two categories: deterministic and stochastic shortest path problems. In deterministic shortest path, using the historical average travel time on links, a path with minimum travel time is sought. On the other hand, a stochastic shortest path problem considers link travel time as a random variable and a path with minimum expected travel time between an origin and destination is sought. The stochastic shortest path problems can be further divided into two categories. The first category tries to find *a priori* solution that minimizes the expected cost (Mirchandani and Soroush 1985) or expectation-variance cost (Khani and Boyles 2015; Zhang and Khani 2019), while the second category finds an online optimal solution that allows decisions to be made at various stages (recourse or adaptive routing problem).

The recourse problem is an opportunity for a decision-maker to re-evaluate their remaining path based on the information obtained *en route*. Croucher (1978) seems to be the first to study this type of problem. Hall (1986) specified that the least expected travel time path between two nodes in a network with travel time both random and time-dependent cannot be found using standard shortest path algorithms (such as Dijkstra's algorithm), as the optimal route choice is not a simple path, but a strategy or hyperpath in which arcs are chosen based on an adaptive decision rule. Andreatta and Romeo (1988) described this problem on a network with general dependency in which different possible realizations of the network are considered. Psaraftis and Tsitsiklis (1993) presented the shortest path problem in acyclic networks in which the cost of an arc is a function of an environment variable at the head node of that arc. Each of these environment variables is assumed to evolve according to an independent Markov process. Polychronopoulos and Tsitsiklis (1996) presented solution methods for two models of the shortest path with recourse. The first model assumes a network with possible realizations of the arc costs whereas the second model assumes independent arc costs as a random variable. A label setting algorithm was developed by Miller-Hooks and Mahmassani (2003) to evaluate *a priori* least expected travel time path in a stochastic time-varying (STV) network by assuming independence among arc costs as well as time periods. They used a similar label setting algorithm to evaluate the least expected hyperpaths for adaptive route choice in STV networks (Miller-Hooks 2001). Gao and Chabini (2006) also developed an exact algorithm and several approximation algorithms such as certainty equivalence, no-online-information, and open-loop feedback algorithm to find the routing policy in STV networks.

As several authors (Polychronopoulos and Tsitsiklis 1996; Andreatta and Romeo 1988) considered a general spatial dependency, Waller and Ziliaskopoulos (2002) considered limited dependency in evaluating the shortest path with recourse. They presented two types of dependencies in a stochastic network. The one-step spatial dependency describes the transition of an arc state to another depending on the state of adjacent arcs. On the other hand, the temporal dependency reveals the state of downstream arcs when a traveler reaches a particular node. In both cases, a labeling algorithm is presented to evaluate the online shortest path having a minimum expected length. Provan (2003) classified the shortest path with recourse problem into two cases: *Reset* and *No reset*. In no reset case, if an arc is visited, then its cost becomes deterministically known upon further visits. On the other hand, in the reset case, each visit to a node is an independent stochastic trial. Provan proved that every instance of the No reset problem is an NP-Hard problem and provided a polynomial-time algorithm for the reset case. Recently, Boyles and Rambha (2016) formulated this problem as a total cost Markov Decision Process (MDP) to detect the unbounded instances in the presence of negative arc costs. The stochastic version of adaptive route choice was developed by Gao et al. (2008), Gao (1999), and Ding-Mastera (2016). Gao et al. (2008) proposed an adaptive path model and an adaptive routing policy choice model for STV networks and showed that the adaptive routing policy choice model can achieve lesser expected cost in comparison to the adaptive path model. Gao and Huang (2012) designed a heuristic algorithm for the adaptive routing problem for four different types of online information schemes, namely, perfect information, delayed global information, global pre-trip information, and up-to-date radio information on a subset of arcs.

The adaptive routing by passengers in the case of transit networks is also a common phenomenon. Due to several bus options, passengers often adopt varying strategies to board the bus (Chriqui and Robillard 1975). Spiess and Florian (1989) proposed that passengers take the first line among the set of attractive lines and formulated this problem as a linear program to solve the transit assignment problem. Nguyen and Pallottino (1988), formally defined hyperpaths to model this behavior which is a set of paths (a subnetwork) rather than just a single path. There are several studies such as Liu et al. (2019) and Nie and Wu (2009) which consider online routing based on the assumptions on the distribution of the headway. This problem is generally called the shortest path problem considering on-time arrival (SOTA) probability. Rambha et al. (2016) presented a pioneering effort in the case of the adaptive transit routing problem in a time-dependent stochastic network. They formulated a finite horizon MDP and presented several pre-processing ways to reduce the computational time of computing the least expected cost hyperpath. Khani (2019) developed a label setting algorithm to evaluate an online shortest path for reliable routing in case of schedule-based transit networks considering transfer failure probability.

The online shortest path has also been used for modeling route choice in various applications. A User Equilibrium with Recourse (UER) model was developed by Unnikrishnan and Waller (2009), which has been used for continuous network design (Unnikrishnan and Lin 2012), optimally locating the information sensors (Boyles and Waller 2011), and marginal cost pricing for system optimal traffic assignment with recourse (Rambha et al. 2018). Using a policy-size logit route choice, Gao (1999) developed a policy-based stochastic dynamic traffic assignment model. Jafari and Boyle presented *a priori* routing and online routing of an Electric Vehicle (EV) in a network with stochastic travel time in Jafari and Boyles (2017a) and Jafari and Boyles (2017b) respectively. It has also been used for online parking search by Tang et al. (2014) and Levin and Boyles (2019).

None of the above studies has considered adaptive routing in a multimodal framework, such as for park-and-ride mode which consists of routing in both auto and transit networks. In this research, we study the adaptive behavior of commuters in terms of park-and-ride decision and its location choice, considering two modes of transportation. We assume that the link travel times on the roadway network are stochastic and time-dependent and transit service has a schedule. We model this phenomenon as an infinite-horizon total cost MDP. A numerical example on a small network based on a real case study is presented. In nutshell, this research presents a method to evaluate efficient routing strategies for park-and-ride when online information is provided. This understanding will have a future impact on the quality of a trip and the satisfaction of travelers through improved planning, the location of park-and-ride facilities, tolling on freeways, and the scheduling of transit service.

## 3 Problem Description

Given the uncertainty in travel time on a freeway and bus arrival time at the park-and-ride locations, the problem is to navigate a traveler commuting from a suburban region to a downtown location in minimum expected travel cost by deciding:

1. Whether to continue driving towards the destination or use a park-and-ride location to take transit?
2. At which park-and-ride location should the traveler park their car?
3. At what times of the day, taking transit from a park-and-ride location minimizes the overall expected travel cost?

To better understand the problem, an illustrative example is provided in Fig. 1. The network given in this example consists of 4 nodes and 4 links, with travel time shown on the links. The path $(r, p, d)$ represents a freeway with $p$ as the park-and-ride node from where it is possible to take transit. The travel time and wait time on link $(p, d)$ and $(p, t)$ respectively are random variable (but not time-varying for simplicity) whose distribution is shown on the links. When a traveler arrives at node $p$, one of the possible realizations of cost on the outgoing links is revealed to them. Depending on the information and time of arrival, we need to compute an optimal policy that decides whether to take transit or continue driving on the freeway to minimize the expected cost to reach the destination. In this case, it is optimal to take freeway from $p$ if the travel time of 10 units is revealed to the traveler on it, otherwise it is optimal to take transit from there. The expected cost of using the adaptive routing policy is 42.5 units in comparison to an expected cost of 56 units by always taking the freeway and 45 units by always taking the park-and-ride option.

## 4 Preliminaries

In this section, we describe the network topology and various assumptions involved in modeling the stated problem. Let us start by considering an auto network represented by a directed graph $G_a(N_a, P_a, A_a)$, where $N_a$ represents the set of nodes, $A_a$ represents the set of links, and $P_a \subset N_a$ represents the set of nodes where park-and-ride facilities are located. Let $G_t(N_t, A_t)$ be the directed graph representing the time-dependent transit network, where $N_t$ and $A_t$ represent the time-dependent nodes and links in the transit network. The set $A_t = A_t^w \cup A_t^v$ can be partitioned into a set of

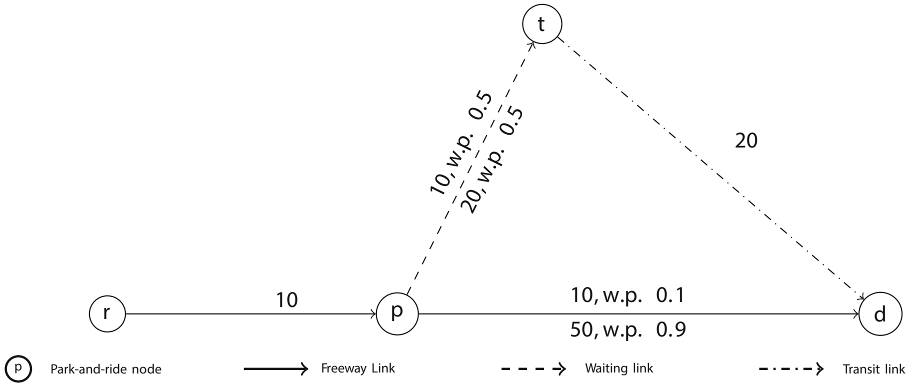


**Fig. 1** An illustrative example of a park-and-ride trip in a network with random travel time

walking (used for access and egress) or waiting links $A_t^w$ and a set of in-vehicle links $A_t^v$. The park-and-ride nodes in the auto network serve particular nodes in the transit network. Let $\mathcal{Z} : P_a \mapsto 2^{N_t}$ be the set-valued map that assigns a park-and-ride node to a set of transit nodes. Note that there can be no park-and-ride node connected to a transit node if the walking distance to the nearest park-and-ride node is more than a certain threshold (say 0.75mi). Let $M = \{(n_a, n_t) \in P_a \times N_t : n_t \in \mathcal{Z}(n_a)\}$ be the set of links created between park-and-ride nodes and transit nodes and vice-versa known as *Mode transfer links*. These links involve walking time to access the transit stop and waiting time before the arrival of the bus. Let $Z = \cup_{n \in P_a} \mathcal{Z}(n)$ be the collection of all transit network nodes which are connected to various park-and-ride facilities. Further, let us denote $\Gamma(i)$ and $\Gamma^{-1}(i)$ as the forward star $\{j : (i, j) \in N_a \cup N_t \cup M\}$ and backward star $\{j : (j, i) \in N_a \cup N_t \cup M\}$ of node $i$ respectively and $\mathcal{T}$ as the set of discretized time values with interval $\Delta t$. The value of $\Delta t$ should be less than the minimum travel time on any link. This is necessary to ensure the state transition from one time interval to another by taking an action in the MDP described later in Section 5.2. As the decision of taking transit at a park-and-ride node depends on the time the traveler arrives at that node, we need to expand the static road network into a time-dependent network. This is done by replicating the nodes $i \in N_a$ at different time intervals $t \in \mathcal{T}$ and connecting them with the corresponding cost of travel.

We assume that the travel time and wait time on the auto network links $A_a$ and mode transfer links $M$, respectively, are time-dependent discrete random variables with finite support, but the travel time on transit links ($A_t = A_t^w \cup A_t^v$) is constant. The random travel time on links $A_a \cup M$ results in a node-dependent stochasticity as when the traveler arrives at node $i \in N_a$, the information about the travel time on all the downstream links attached to it is revealed to them. Let $\Theta_i(t)$ denote the set of possible states at node $i \in N_a$, where probability of observing a particular state $\theta \in \Theta_i(t)$ at time $t$ is $p^\theta$. Each state $\theta \in \Theta_i(t)$ is a realization of travel time or wait time (if $i \in P_a$) on downstream links attached to node $i$. Let $S_{ij}(t)$ be the set of possible realizations of travel time (or wait time) on link $(i, j) \in A_a \cup M$ at time $t$. Clearly, $\Theta_i(t) = \times_{(i,j) \in A_a \cup M} S_{ij}(t)$ are the possible states at node $i$ and time $t$, where $\times$ represents the cross product of the sets. Moreover, each $\theta \in \Theta_i(t)$ represents an information that is revealed to a traveler when arriving at node $i$ at time $t$. This information consists of travel time or wait time (if arrived at a park-and-ride node) on outgoing links from node $i$ which is represented by $c_{ij}^\theta, \forall \theta \in \Theta_i(t), \forall j \in \Gamma(i)$. Let $\hat{c}_{ij}$ be the cost associated with link $(i, j) \in A_t$. A value of time parameter $\alpha$ is used to convert monetary costs into time units. The monetary costs may include parking cost $\tau_p$ at a particular node or transit fair $\tau_f$ to board a transit route. Before proceeding further, we make the following assumptions in this study, which help us in defining the current problem.

### Assumptions

1. The travel time on auto network links ($A_a$) and waiting time on mode transfer links ($M$) are modeled as time-dependent discrete random variables with finite support. We do not assume any stochasticity in the travel time experienced in the transit network. This is because the transit vehicles can use high occupancy

vehicle lane on freeways or dedicated infrastructure (such as light rail or heavy rail transit), which, in general, provide reliable travel times.

2. The arc travel times on the road network takes new independent cost values each time an arc is traversed. This condition is called *reset* condition and is necessary to develop a polynomial-time algorithm for this problem (Provan 2003).
3. There are no time-dependent correlations in travel time or wait time on links and are assumed to be independent.
4. The online information provided to the commuter is assumed to be one of the realizations of traffic state obtained from the historical data, and the proposed routing algorithm computes optimal policy for every such realization.
5. The online information is available at each node of the auto network. This information may consist of travel time or wait time depending on the type of link, which is accessed from online routing applications, Variable-message signs (VMS), or radio signals. Note that this is not a strict requirement, as the nodes with no online information can just have the deterministic cost for adjacent downstream links (with probability 1).
6. The passenger is assumed to be an expected-cost-minimizer. The word "optimal policy" used in this article minimizes the expected cost of travel (comprising of travel time, wait time, fare, and parking cost), and does not consider other attributes affecting passengers' utility.
7. The passenger has a preferred arrival time $PAT$ at the destination location.
8. There exists a transit stop near the destination of the traveler, which is accessible by walking. This is generally true for the destinations in CBD areas, which are well served by transit services.
9. There is sufficient capacity at park-and-ride facilities to park.
10. There is sufficient capacity to board the transit vehicle. We do not model the failure-to-board instances in this problem.
11. The auto and transit networks are connected.
12. We assume that the waiting time and travel time are the same cost to the traveler and use the same parameter to convert it into travel cost.

The relaxations of Assumptions 1, 6, 9, and 10 are research topics in their own right and a discussion on them is provided in Section 8.

## 5 Adaptive Routing of Park-and-ride Trip

This section presents the solution methodology for adaptive routing of the park-and-ride trip. The problem is to navigate a passenger starting at origin $r \in N_a$ to destination $d \in N_t$ in minimum expected travel cost. The computation of optimal routing consists of two steps. In the first step, we compute the latest departure time ($LDT$) label from every bus stop of park-and-ride locations ($Z$), so that the traveler reaches the destination before the preferred arrival time. This $LDT$ label is computed for each time-dependent transit node in $Z$, which is represented by a unique transit route and a trip departure time. The trip information is obtained from the transit schedule data, more specifically, General Transit Feed Specification (GTFS) data.

A label setting algorithm, known as *schedule-based transit shortest path* (SBTSP) is used to compute the $LDT$ labels, which is presented in Section 5.1. In the second step, routing in a network with uncertain travel time on the road and wait time for buses is considered. This is formulated as a stochastic shortest path problem, which in turn, will help in evaluating an optimal policy for routing as well as the park-and-ride decision. In the next subsection, we describe the first step of the procedure.

### 5.1 Schedule-Based Transit Shortest Path

This is a label setting algorithm that runs on a time-dependent transit network. The time-dependent transit network can be created using General Transit Feed Specification (GTFS) data, which is a standard format of schedule data released publicly by various transit agencies throughout the world. We use trip-based network representation for modeling the transit network. In this procedure, the main consideration is given to the fact that every deviation from a transit route cannot be considered as a transfer. Depending upon the acceptable walking time, waiting time, and direction of movement, transfers are created between two transit routes. We refer the interested reader to the article by Khani et al. (2014) for more details about the procedure to create such a network and its open-source implementation in R by Kumar and Khani (2019).

As a transit network consists of different types of links (access/egress, waiting, in-vehicle, and transfer links), the quickest path found on a transit network may not be an optimal path for a passenger (Tong and Richardson 1984a). For example, a quickest path may return a path that consists of a large number of transfers and walking components, which are considered more onerous than other components of a transit trip. Moreover, transfers are less reliable, therefore, an algorithm should avoid selecting such paths. To do that, we define weights associated with different types of links in $A_t$ for calculating a generalized cost. Let $\eta_{ij}$ be the weight associated with link $(i, j) \in A_t$ depending upon its type. For example, walking and waiting links can be assigned a higher weight in comparison to other types of links in the transit network. Given that a passenger departs from node $n \in N_t$, we find the labels $\gamma_n$ that specify the latest time a passenger should depart from $n$ to reach $d$ before $PAT$. Let $SEL$ be a scan eligible list, $\xi_i$ be the predecessor of node $i$ in the shortest path. Using the weighted sum of the cost of traversing different types of links in the transit network, a generalized cost label $\gamma_n^{gc}$ is maintained for each node $n \in N_t$, which satisfies the following principle of optimality:

*Bellman's principal of optimality*: For any node $i \in N_t$, $\gamma_i^{gc}$ should satisfy the following condition:

$$\gamma_j^{gc} = \min_{j \in \Gamma(i)} \{\gamma_i^{gc} + \hat{c}_{ij}\} \tag{1}$$

Pseudocode for finding an optimal transit path is given in Algorithm 1. The algorithm runs backward shortest path from $d$ at $PAT$. We initialize a scan eligible list (SEL) and maintain two different types of labels–time labels $\gamma$ and generalized cost labels $\gamma^{gc}$. The time labels are used to store information about the actual time of departure from that node while the generalized cost labels $\gamma^{gc}$ are used to maintain a minimum generalized cost for a rider which is calculated as the weighted sum of the cost of traversing different types of links in the transit network (line 9). Line 11 checks Bellman's principle of optimality (1). Finally, labels $\gamma_{pnr}$ and predecessor set $\xi$ are returned by the algorithm which can be used to retrieve $LDT$ from park-and-ride nodes and optimal transit itinerary respectively.

**Algorithm 1** Schedule-based transit shortest path.

1: **procedure** SBTSP($G_t, \eta, PAT$)
2: $\gamma_d \leftarrow PAT, \gamma_d^{gc} \leftarrow 0, \xi_d \leftarrow$ NULL
3: $\gamma_j \leftarrow -\infty, \gamma_j^{gc} \leftarrow \infty, \xi_j \leftarrow \phi \quad \forall j \neq d$ ▷ intializing node labels and SEL
4: SEL $\leftarrow \{d\}$
5: **while** SEL $\neq \phi$ **do**
6: $i \leftarrow \underset{j}{\text{argmin}}\{\gamma_k^{gc} \mid k \in \text{SEL}\}$
7: SEL $\leftarrow$ SEL$\setminus\{i\}$
8: **for** each $j \in \Gamma^{-1}(i)$ **do**
9: $\gamma_{new}^{gc} \leftarrow \gamma_i^{gc} + \eta_{ij} * \hat{c}_{ji}$ ▷ $\eta$ is user defined
10: $\gamma_{new} \leftarrow \gamma_i - \hat{c}_{ji}$
11: **if** $\gamma_{new}^{gc} < \gamma_j^{gc}$ **then**
12: $\gamma_j^{gc} \leftarrow \gamma_{new}^{gc}, \gamma_j \leftarrow \gamma_{new}, \xi_j \leftarrow i$ ▷ Updating labels
13: SEL $\leftarrow$ SEL $\cup\{j\}$
14: $\gamma_{pnr} \leftarrow \{\gamma_i \ \forall i \in Z\}$ **return** $\gamma_{pnr}, \xi$

**Proposition 1** *The worst case computational complexity of Algorithm 1 is* $\mathcal{O}(|A_t| \log |N_t| + |N_t| \log |N_t|)$.

### 5.2 Stochastic Shortest Path Problem

After computing the $LDT$ labels $\gamma_{pnr}$, we contract the transit network by connecting the nodes in $Z$ directly to the destination $d$ with the corresponding cost of travel obtained from these labels. With a little abuse of notation, let us assume that the set $A_a$ also consists of the mode transfer links $M$ as well as the contracted transit links. Similarly, let us assume $Z$ and $\{d\}$ are also part of $N_a$. After this contraction procedure, we obtain a single graph, $G_a(N_a, A_a)$ on which we are going to define the stochastic shortest path problem.

In this problem, the online information about the travel time or wait time on downstream links is provided when the traveler reaches the head node. This information can help the traveler in making a smarter choice and improving the overall cost of the trip by re-evaluating the route cost, park-and-ride facilities, and waiting for transit. Due to a schedule and current congestion condition, the transit trip consists of a waiting component depending on the available route. The waiting time on possible routes (represented by mode transfer links) is also modeled as a time-dependent random variable and is realized when the traveler reaches a park-and-ride node. The wait time information is accessible through routing applications (such as Transit App), which utilizes the Automatic Vehicle Location sensors installed in transit vehicles

(Webb et al. 2020; Kumar et al. 2018). With this setup, the objective is to minimize the expected cost to reach the destination. As the problem exhibits sequential decision making in a stochastic dynamic system, it can be formulated as a Markov Decision Process (MDP). The problem can be formulated as a finite horizon MDP where the time of arrival at various nodes are considered as different stages of the problem. However, we formulate it as an infinite horizon MDP by including the time of arrival at various nodes as part of the state of the problem. This is performed to enjoy various nice properties and algorithms designed for infinite horizon problems. The infinite horizon total cost MDPs are characterized by a set of states, the set of actions available at each state, the probability of transition from one state to another by taking a particular action at each stage, and finally, the cost incurred by taking such an action. These components are described below:

### 5.2.1 State Space

The state space $S \mapsto N_a \times \mathcal{T} \times \Theta$ defines the possible states in which a traveler can be present. Each state $s \in S$ is defined by tuple $s = (i, t, \theta)$, where, $i \in N_a$ represent the auto, park-and-ride, or the destination node. Here, $t \in \mathcal{T}$ is the time of arrival at node $i$, and $\theta \in \Theta_i(t)$ represents the information obtained at node $i \in N_a$ at time $t \in \mathcal{T}$. Any state corresponding to the destination node is considered as an absorbing state (once a traveler reaches there will remain there forever).

### 5.2.2 Action Space

Upon arrival at each node with no park-and-ride facility, the decision-maker considers the current travel cost, and the availability of the information about the future travel time on downstream arcs and then decide which arc to take next. On the other hand, when the traveler arrives at a node with a park-and-ride facility, she has two options available: whether to park and wait for transit or take one of the downstream auto links. Therefore, the actions available at state $(i, t, \theta)$ are denoted by $u(i, t, \theta) = \{j \in N_a : (i, j) \in A_a\}$ i.e., the set of nodes in the forward star of node $i$. Let $C$ be the set of such actions. A policy $\pi : S \mapsto C$ defines a stationary policy that specifies the action to be taken at any state. Here, $\pi(i, t, \theta) \in u(i, t, \theta), \forall (i, t, \theta) \in S$.

### 5.2.3 One-step Costs

If the decision maker chose to take an auto link, a cost equal to the travel time on the forward link is incurred. Similarly, if she decides to take a waiting link to board a transit route, a cost equal to the wait time is incurred. Let us denote the cost of choosing $\pi(i, t, \theta)$ at $(i, t, \theta)$ by $c^{\theta}_{i\pi(i,t,\theta)}$, where $\theta \in \Theta_i(t)$. The cost of transitioning from $(d, t, \theta)$ to itself is zero, $\forall \theta \in \Theta_d(t), \forall t \in \mathcal{T}$. Furthermore, the states associated with nodes in $Z$ has the only possibility of transitioning to the destination with probability 1.0 and cost equal to the transit travel time computed using Algorithm 1.

### 5.2.4 Transition Functions

A traveler at state $(i, t, \theta)$, following policy $\pi$, transitions to a new state $(\pi(i, t, \theta), t + c^{\theta}_{i\pi(i,t,\theta)}, \theta')$, by taking an action $\pi(i, t, \theta) \in u(i, t, \theta)$, the probability of which is denoted as $p^{\theta'}, \theta' \in \Theta_{\pi(i,t,\theta)}(t + c^{\theta}_{i\pi(i,t,\theta)})$. Note that by fixing the policy $\pi$, one can construct a transition diagram with corresponding states and transition probabilities. Let us denote the overall transition matrix for policy $\pi$ by $\mathbb{P}_{\pi}$. The probability of transitioning from $(d, t, \theta), \theta \in \Theta_d(t)$ to itself, by taking any action $j \in u(d, t, \theta)$ is 1.0.

### 5.2.5 Value Function

Let $J_{\pi}(i, t, \theta)$ at state $(i, t, \theta)$ denote the cost incurred by a traveler to reach the destination starting from node $i \in N_a$ at time $t \in \mathcal{T}$ receiving information $\theta \in \Theta_i(t)$ following policy $\pi$. Mathematically, one can write

$$J_{\pi}(i, t, \theta) = \lim_{K \to \infty} \mathbb{E} \left\{ \sum_{k=0}^{K-1} c^{\theta^k}_{i^k, t^k, \pi(i^k, t^k, \theta^k)} | i^0 = i, \theta^0 = \theta, t^0 = t, \pi \right\} \qquad (2)$$

where, $(i^k, t^k, \theta^k)$ represents the state at $k^{\text{th}}$ stage, which represents the decision points of the traveler. Given an initial state $(i^0, t^0, \theta^0)$, the task is to determine the least expected cost $\hat{J}^*(i, t) = \sum_{\theta \in \Theta_i(t)} p^{\theta} J^*(i, t, \theta)$ from every node $i$ at time $t$ to the destination $d$ as well as an optimal routing policy $\pi^*(i, t, \theta)$ which minimizes long-term cost function given by Eq. 2.

**Definition 1** (Bellman Operator). The Bellman optimality operator $T : S \to \mathbb{R}$ is defined as follows:

$$(TJ)(i, t, \theta) = \min_{j \in u(i,t,\theta)} \{c^{\theta}_{ij} + \sum_{\theta' \in \Theta_j(t + c^{\theta}_{ij})} p^{\theta'} J(j, t + c^{\theta}_{ij}, \theta')\} \qquad (3)$$

Similarly, for a stationary policy $\pi$, let us define the Bellman operator for policy $\pi$, $T_{\pi} : S \to \mathbb{R}$ as follows:

$$(T_{\pi} J)(i, t, \theta) = c^{\theta}_{i\pi(i,t,\theta)} + \sum_{\theta' \in \Theta_{\pi(i,t,\theta)}(t + c^{\theta}_{i\pi(i,t,\theta)})} p^{\theta'} J(\pi(i, t, \theta), t + c^{\theta}_{i\pi(i,t,\theta)}, \theta') \qquad (4)$$

In vector-matrix form, Eq. 4 can be written as:

$$T_{\pi} \boldsymbol{J} = \boldsymbol{c}_{\pi} + \mathbb{P}_{\pi} \boldsymbol{J} \qquad (5)$$

where, $\boldsymbol{J}$, $\mathbb{P}_{\pi}$, and $\boldsymbol{c}_{\pi}$ denote the vector of cost functions, transition matrix corresponding to the policy $\pi$, and the vector of the expected cost of transitions following policy $\pi$ respectively. Our objective is to determine the optimal cost function $J^*(i, t, \theta)$ and an optimal policy $\pi^*$. We next show that for any stationary policy $\pi$, the resulting transition graph is acyclic.

**Proposition 2** *For any stationary policy $\pi$, the associated transition graph is acyclic assuming that there do not exist self transition probabilities associated with any state except the destination.*

**Definition 2** (Proper Policy). A stationary policy $\pi$ is said to be *proper* if, when using this policy, there is a positive probability that the destination will be reached after at most $m \in \mathbb{Z}_+$ stages, regardless of the initial state, i.e., if $\phi_\pi = \max_{(i,t,\theta)\in S}\{\text{Prob}(x_m \neq (d, t, 0) \text{ for some } t \in \mathcal{T} \mid x_0 = (i, t, \theta), \pi)\} < 1$, where $x_k$ represents the state at stage $k$. A stationary policy which is not proper is called an *improper* policy.

With Assumption 11 and Proposition 2, we have an acyclic connected graph. Therefore, there exists at least one proper policy $\pi$ as for every arc $(i, j) \in A_a$, we have $\mathbb{P}_\pi(i, j) > 0$. The implication of the existence of a proper policy is that it becomes inevitable to reach the destination node.

For a proper policy $\pi$, one can repeatedly apply the Bellman operator $T_\pi$ to the cost function $\boldsymbol{J}_0 = \{0\}^{|S|}$ to evaluate the cost of the policy $\pi$ (see Eq. 6). Moreover, $\boldsymbol{J}_\pi$ is the unique solution of the system of Eq. 7 (Bertsekas 2012)

$$\lim_{k\to\infty} (T_\pi^k J)(i, t, \theta) = \lim_{k\to\infty} \left(T(T_\pi^{k-1} J)\right)(i, t, \theta) = J_\pi(i, t, \theta),\ \forall (i, t, \theta) \in S \quad (6)$$

$$\boldsymbol{J}_\pi = T_\pi \boldsymbol{J}_\pi \quad (7)$$

.

Similarly, one can evaluate the optimal cost function vector $\boldsymbol{J}$ by solving the Bellman Eq. 8. Since the solution of a high-dimensional system of equations can be time-consuming, we can also repeatedly apply mapping $T$ on vector $\boldsymbol{J}_0$ to evaluate $\boldsymbol{J}^*$ (see Eq. 9). This is the basis of the value iteration method (presented in the next section), which is a popular method to solve the Bellman Eq. 8.

$$\boldsymbol{J}^* = T\boldsymbol{J}^* \quad (8)$$

$$\lim_{k\to\infty} T^k J(i, t, \theta) = J^*(i, t, \theta), \forall (i, t, \theta) \in S \quad (9)$$

### 5.2.6 Solution Algorithm

The previous section described that the optimal cost functions and a stationary policy can be evaluated by solving the Bellman Eq. 8. However, it becomes difficult to solve it when the cardinality of the state space is a large number. This problem is commonly referred to as the “curse of dimensionality” in the MDP literature. It turns out that the state space of the current problem can be reduced by averaging the information vector $\theta \in \Theta$. This is because $\theta$ is an uncontrollable component of the state. It depends on node $i \in N_a$ at which the traveler is currently located and the time of arrival $t \in \mathcal{T}$ at that node but not on the control $u(i, t, \theta)$. In that case, we can reformulate the DP problem only on the controllable components (i.e., $(i, t)$) of the state space with the dependence on the uncontrollable component $\theta$ being “averaged out.” To do that, let us consider another cost function $\hat{J}(i,t)$ defined on the reduced state space $S' = N_a \times \mathcal{T}$.

$$\hat{J}^*(i,t) = \sum_{\theta \in \Theta_i(t)} p^\theta J^*(i,t,\theta) \tag{10}$$

$$= \sum_{\theta \in \Theta_i(t)} p^\theta \min_{j \in u(i,t,\theta)} \left\{ c_{ij}^\theta + \sum_{\theta' \in \Theta_j(t+c_{ij}^\theta)} p^{\theta'} J^*(j, t + c_{ij}^\theta, \theta') \right\} \tag{11}$$

$$\hat{J}^*(i,t) = \sum_{\theta \in \Theta_i(t)} p^\theta \min_{j \in u(i,t,\theta)} \{c_{ij}^\theta + \hat{J}^*(j, t + c_{ij}^\theta)\} \tag{12}$$

The Eq. 12 is the Bellman equation in the reduced state space. The cost functions $\hat{J}^*(i,t)$ can be viewed as the optimal expected cost-to-go from node $i$ and time $t$ before realizing the online information.

The above stochastic shortest path problem can be solved using the methods used for solving the infinite-horizon MDP. This includes value iteration, policy iteration, and linear programming. We describe various efficient methods for this problem below:

**Value Iteration Method** Generally, VI requires an infinite number of iterations to converge to an optimal solution. However, for the current problem, it converges to an optimal solution in finite number of iterations (Proposition 3).

**Algorithm 2** Value Iteration for adaptive park-and-ride routing.

**procedure** VI
  **for** $(i,t) \in S'$ **do**
    $\hat{J}_0(i,t) \leftarrow \infty$
  **for** $t \in \mathcal{T}$ **do**
    $\hat{J}_0(d,t) \leftarrow 0$
  **for** $k = 0$ to $|N_a|$ **do**
    **for** $t \in \mathcal{T}$ **do**
      **for** $i \in N_a$ **do**
        $\hat{J}_k(i,t) = 0$
        **for** $\theta \in \Theta_i(t)$ **do**
          $\hat{J}_k(i,t) \mathrel{+}= p^\theta \min_{j \in u(i,t,\theta)} \{c_{ij}^\theta + \hat{J}_{k-1}(j, t + c_{ij}^\theta)\}$
  **for** $t \in \mathcal{T}$ **do**
    **for** $i \in N_a$ **do**
      **for** $\theta \in \Theta_i(t)$ **do** ▷ Computing optimal policy
        $\pi(i,t,\theta) \leftarrow \operatorname{argmin}_{j \in u(i,t,\theta)} \{c_{ij}^\theta + \hat{J}^*(j, t + c_{ij}^\theta)\}$
  **return** $\hat{J}, \pi$

**Proposition 3** *Given a directed acyclic transition graph corresponding to any optimal stationary policy $\pi^*$, the value iteration will yield the optimal cost vector $\hat{\boldsymbol{J}}^*$ in at most $|N_a|$ iterations, when initialized as below:*

$$\hat{J}^*(i,t) = \infty,\ \forall (i,t) \in S',\ i \neq d$$
$$\hat{J}^*(d,t) = 0,\ \forall t \in \mathcal{T}$$

The overall value iteration method is summarized in Algorithm 2. If there exists more than one optimal policy, then the decision maker can simply pick one of the optimal policies as they all provide the same expected cost.

**Proposition 4** *The worst case computational complexity of Algorithm 2 is* $\mathcal{O}(|S||N_a||A_a|)$.

**Label Correcting Method** The dynamic stochastic shortest path problem can also be solved exactly using a label correcting algorithm proposed by Cheung (1998). The algorithm starts by initializing a scan eligible list $SE$ containing the neighbors of the destination node. Then, it scans elements in the backward direction updating the label of every node for every time interval. Unlike the value iteration algorithm, it updates the labels of sets $S'_0$, $S'_1$, etc. described in the proof of the Proposition 4 (see Appendix B) sequentially in various iterations rather than attempting to update labels of all the elements of set $S'$ in each iteration. The overall steps of finding the optimal policy using label correcting algorithm are summarized in Algorithm 3.

**Algorithm 3** Label correcting algorithm for adaptive park-and-ride routing.

**procedure** LC
  **for** $(i,t) \in S'$ **do**
    $\hat{J}(i,t) \leftarrow \infty$
  **for** $t \in \mathcal{T}$ **do**
    $\hat{J}(d,t) \leftarrow 0$
  $SE \leftarrow \Gamma^{-1}(d)$
  **while** $SE \neq \phi$ **do**
    Remove an element $i$ from $SE$
    **for** $t \in \mathcal{T}$ **do**
      $tempJ \leftarrow 0$
      **for** $\theta \in \Theta_i(t)$ **do**
        $tempJ \mathrel{+}= p^\theta \min_{j \in u(i,t,\theta)} \{c^\theta_{ij} + \hat{J}(j, t + c^\theta_{ij})\}$
      **if** $tempJ < \hat{J}(i,t)$ **then**
        $\hat{J}(i,t) \leftarrow tempJ$
        $SE \leftarrow SE \cup \Gamma^{-1}(i)$
  **for** $t \in \mathcal{T}$ **do**
    **for** $i \in N_a$ **do**
      **for** $\theta \in \Theta_i(t)$ **do** ▷ Computing optimal policy
        $\pi(i,t,\theta) \leftarrow \underset{j \in u(i,t,\theta)}{\operatorname{argmin}} \{c^\theta_{ij} + \hat{J}^*(j, t + c^\theta_{ij})\}$
  **return** $\hat{J}, \pi$

**Proposition 5** *The worst case computational complexity of Algorithm 3 is* $\mathcal{O}(|S||A_a|)$.

## 6 Case Study of I-394

In this article, we present a case study of the freeway I-394 in Twin Cities, MN to show the application of the method presented. I-394 is 9.8 miles long freeway (E-W) serving the Hennepin County of Minnesota. The major junctions along this freeway include I-494 and US-169 in Minnetonka, MN-100 in Golden Valley, and I-94 in Minneapolis. A High Occupancy Vehicle (HOV) lane was built on this freeway in May 2005 to maximize its capacity. The HOV lane can be used by buses and generally provide reliable service when general-purpose lanes are congested.

I-394 connects various sub-urban areas to Downtown Minneapolis. Due to the congestion during peak periods, five park-and-ride facilities are provided to the travelers (Table 1) along this freeway. These park-and-ride facilities are served by several bus routes, connecting them to various locations in Twin Cities such as Downtown Minneapolis, University of Minnesota campus, Downtown St. Paul, and so on. The park-and-rides facility locations are shown geographically in Fig. 2, and a list of bus routes serving these park-and-ride facilities are provided in Table 1.

### 6.1 Network and Calibration of Distribution of Travel Time and Wait Time

The Minnesota Department of Transportation (MnDOT) has located loop detectors every 0.5mi along I-394. They collect data about the travel speed of cars, which in turn gives us the travel time on different sections of the freeway. We used Google My Maps to create the network. More details about the network topology are given in Table 2.

It is assumed that the travel time recorded using a particular detector applies to half of the distance between the upstream detector and current location and similarly half of the distance between the downstream detector and the current detector location. We consider only two possible states of the freeway links, namely, “congested” and “uncongested”. Highway Capacity Manual defines the quality of any freeway on a

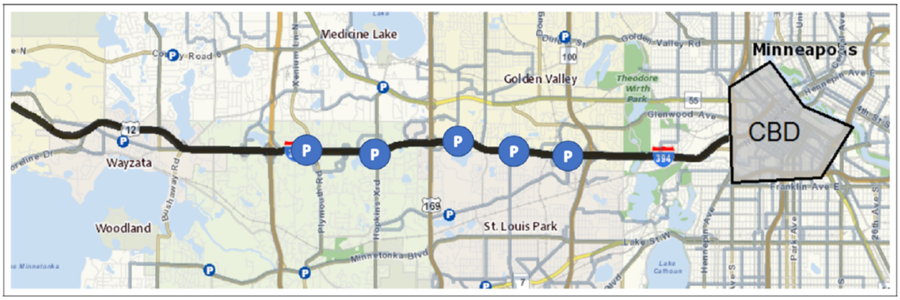


**Fig. 2** Minneapolis CBD and park-and-ride facilities along I-394 corridor

**Table 1** Park-and-ride locations along I-394

| Order | Name | Bus routes | Address |
|---|---|---|---|
| 1 | Plymouth Road Transit Center & Park & Ride | 652, 672, 645, 677 | 13126 Wayzata Blvd, Minnetonka, MN 55305 |
| 2 | I-394 & Co. Rd. 73 Park & Ride | 615, 652, 673, 645, 679 | 1100 Hopkins Crossroads, Minnetonka, MN 55305 |
| 3 | General Mills Blvd. & I-394 | 645, 652, 672, 756 | 8675 Wayzata Blvd. |
| 4 | Louisiana Avenue Transit Center Park & Ride | 9, 604, 643, 645, 652, 663, 672, 705, 756 | 1300 Louisiana Avenue, St, Louis Park, MN 55426 |
| 5* | Park & Ride | 645, 9 | Wayzata Blvd, Minneapolis, MN 55416 |

*5 being closest to Downtown Minneapolis

scale of A-F known as the level of service (LOS). Any level of service below C is considered as a "congested" state of the 1freeway, and "uncongested", otherwise. For a typical highway, the travel speed below 60mph is considered as the level of service C. This value of speed is used to determine the probability of a freeway link being congested or uncongested using the loop detector data collected in April 2017 (Minnesota Department of Transportation 2019). The probability distribution of the travel time is calibrated for every 30 seconds of the time horizon and the mean value of travel time in each time interval is used for a particular state of the link. To avoid inconsistency between the transition of states, each value of travel of time on every link was rounded to the nearest multiple of 30 seconds.

For this case study, the bus routes which serve the Downtown Minneapolis area are only considered. The destination node in our network is assumed to be the intersection of Hennepin Ave and 12th St, which is located in Downtown Minneapolis. The actual bus arrival time at various park-and-ride bus stops is obtained from the historical Automatic Vehicle Location (AVL) data collected by Metro Transit (transit agency in the Twin Cities region) over one year. Then, the difference between the actual and the scheduled arrival time at these bus stops is used for the calibration of the probability distribution of the random wait time of the mode transfer links. Note

**Table 2** I-394 network topology

| | |
|---|---|
| # of nodes | 32 |
| # of auto links | 23 |
| # of access or egress links | 16 |
| Time horizon | 6:00 A.M. - 10:00 A.M. |
| # of time steps | 480 |
| # of transit trips | 68 |

that we also considered the cases when the bus arrived early at the park-and-ride bus stops. For more applications of transit automated data, readers are referred to Kumar et al. (2019) and Kumar and Khani (2020), and Kumar et al. (2021). The value of time ($\alpha$) is assumed to be \$23/h as recommended by Belenky (2011).

### 6.2 Results

After creating the network, calibrating the required probability distributions and reducing the state space, we use both value iteration (Algorithm 2) and label correcting (Algorithm 3) methods to solve the stochastic shortest path problem presented in Section 5.2. We calculated the online shortest path during morning peak hours from 6:00 A.M. to 10:00 A.M., from different nodes of the freeway to the specified destination in Downtown Minneapolis. The number of states generated for this experiment after reducing the state space was 11,562. A typical value of transit fare $\tau_f = \$3$ and parking cost $\tau_p = \$12$ for one day is assumed for this experiment. Usually, the parking is free at park-and-ride locations, so we did not consider any cost associated with it. The value iteration method for this small network took 24 iterations and 18 minutes to converge to the optimal solution. On the other hand, the label correcting method outperformed the value iteration method and took only 1.42 minutes to converge.

Before delving into the results produced by the algorithm, let us first explore the congestion conditions on I-394. Figures 3 and 4 show heatmaps of the travel time during both congested and uncongested conditions. The links are shown on the horizontal axis in the E-W direction whereas the vertical axis shows the time of the day. The node numbers increase in the direction of travel. We can see that during congested conditions (Fig. 4), the overcrowding happens after 7:00 A.M. along three different stretches of the freeway. The first stretch is upstream before the first park-and-ride location. The second and third stretch of overcrowding is between Node 273 and 277 and after 279 up to central downtown respectively. Intuitively, taking park-and-ride between node 273 and 278 seems to be a reasonable choice.

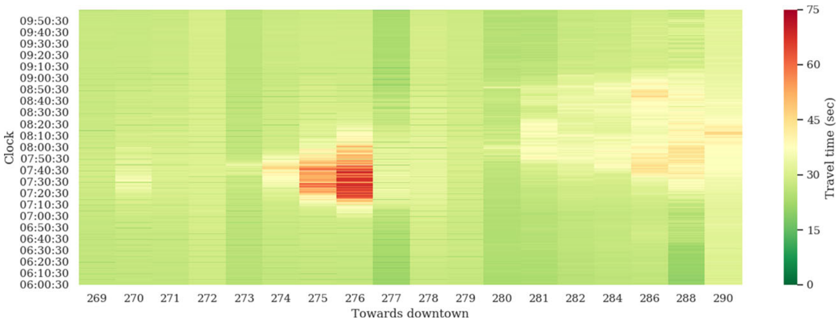


**Fig. 3** Travel time (sec) on links during uncongested conditions (For interpretation of colors in this figure, the reader is referred to the web version of this article)

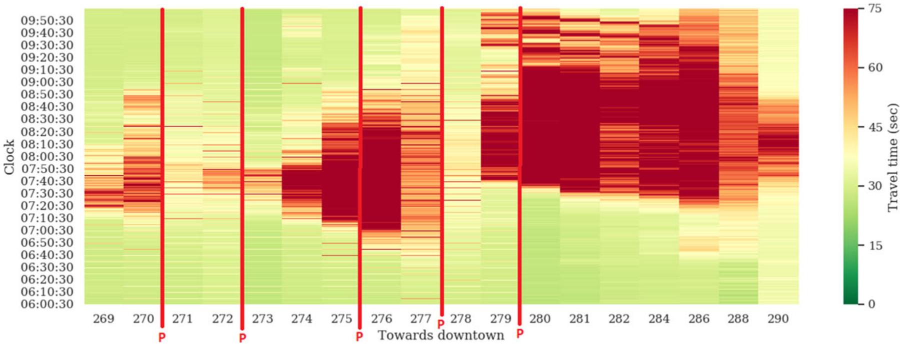


**Fig. 4** Travel time (sec) on links during congested conditions (For interpretation of colors in this figure, the reader is referred to the web version of this article)

#### 6.2.1 Expected cost

Figure 5 shows the results of the expected cost of travel (in seconds) from various nodes to node 291 (destination) computed using Algorithm 2. The four-digit nodes represent park-and-ride nodes along the freeway, which are also marked with the letter P in the figure. The figure shows an increase in the congestion as the time increases on the time scale with severe congestion between 7:00 A.M.-9:45 A.M. However, the availability of the bus at the park-and-ride nodes provides a reduction in the expected travel time during several periods. This is evident from the light yellow-colored stripes appearing within the red-colored region. This reduction in travel time is due to the policy of taking transit at a park-and-ride location. The figure clearly shows the time of the day when park-and-ride mode becomes more attractive in comparison to auto as this will provide faster access to the destination. We observe this significant reduction in the travel time when the buses are not frequent. In case if Metro Transit provides frequent service, we expect to see further improvement in the travel time of the commuters.

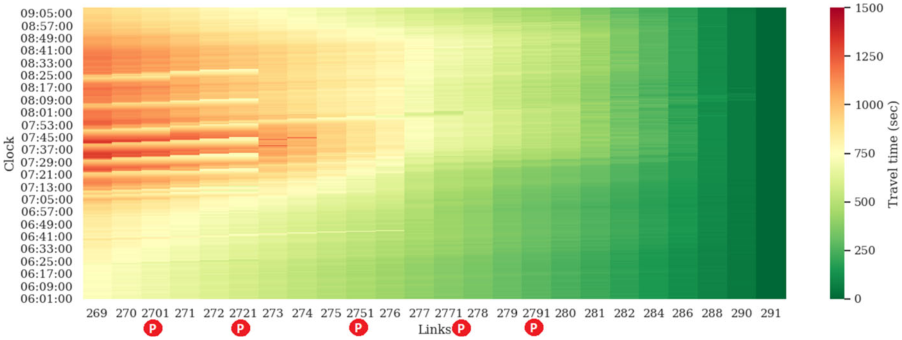


**Fig. 5** Travel cost (sec) of from different nodes to downtown (For interpretation of colors in this figure, the reader is referred to the web version of this article)

### 6.2.2 Optimal Policy

The output of the Algorithm 2 also yields the optimal policy for different node states. The policy will help a traveler to decide which downstream arc to take when arriving at any node. The point of interest in this research is when does taking the transit from any park-and-ride location becomes optimal. To observe this behavior, we plotted the optimal policy for a traveler at park-and-ride exit nodes, which are plotted on the vertical axis in Fig. 6, with the time of arrival at different nodes on the horizontal axis. Figure 6(a) and 6(b) depict the optimal policy when the road network is in the congested and the uncongested state respectively. The policy shows that the park-and-ride option becomes more attractive during the congested conditions than uncongested conditions. Moreover, it is interesting to note that taking transit from I-394 & Co. Rd. 73 Park & Ride and Plymouth Road Transit Center & Park & Ride (which are located upstream) is more frequent than the other park-and-ride facilities.

We use the Monte-Carlo simulation to evaluate the variability of the optimal policy $\pi^*$. For a given time interval, we generate 1000 random trajectories following policy $\pi^*$ starting from the farthest end (node '269' in Fig. 5) and ending at the specified destination. In particular, for every sample and node-time $(i, t)$ pair, we draw a random state $\theta \in \Theta_i(t)$ based on the discrete probability distribution $\{p^\theta\}_{\theta \in \Theta_i(t)}$.

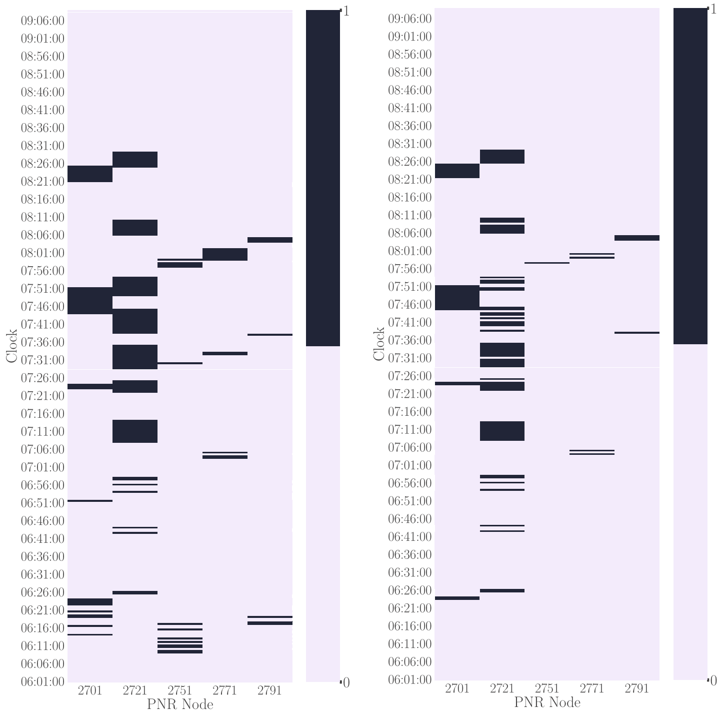


**Fig. 6** Optimal policy for park-and-ride nodes at different arrival times (Black (1) indicates park-and-ride and Blue (0) indicates auto mode)

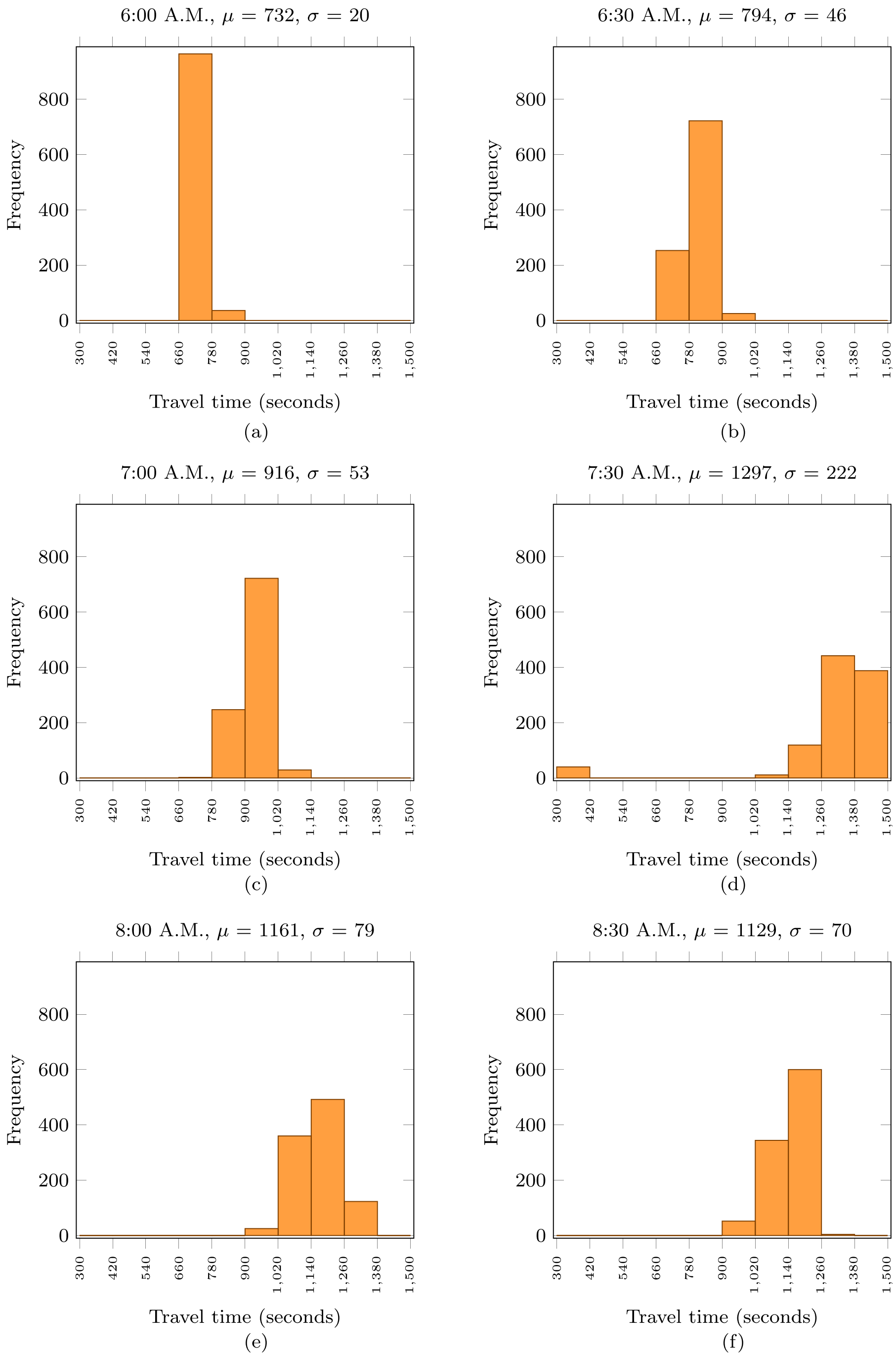


**Fig. 7** Mean and standard deviation of travel time (seconds) to the destination for different time of arrival at node ’269’ computed using 1000 sample trajectories

Figure 7 shows the results of the simulation that includes the distribution with mean and standard deviation of travel time at different times of arrival at node '269'. We can observe that the standard deviation ranges from 20 to 80 seconds for most time intervals except 7:30 A.M. at which it rises to 222 seconds. This is because of one particular trajectory with a travel time equal to 300 seconds.

We further use the same Monte-Carlo simulation to evaluate the attractiveness of various park-and-ride facilities. Figure 8 shows the frequency of use of different park-and-ride facilities at various times of arrival out of 10,000 sample trajectories. Through this analysis, we can make similar observations as made in Fig. 6. The park-and-rides I-394 & Co. Rd. 73 Park & Ride and Plymouth Road Transit Center & Park & Ride (which are located upstream) are more attractive than the other park-and-ride facilities.

### 6.2.3 Comparison with Offline Algorithm

To show the benefits of adaptive routing, we compare the expected cost computed using Algorithm 2 with the offline algorithm (Expected Value (EV) algorithm) proposed by Miller-Hooks and Mahmassani (2003). The offline algorithm computes the least expected time (LET) paths in the stochastic and time-varying transportation network. The output of the algorithm provides the least expected cost from each node to the destination node for possible departure times in the time horizon. A short description and a pseudo code of the algorithm is provided in Appendix A for reference. However, readers interested in more details can refer to the original article by Miller-Hooks and Mahmassani (2003).

Figure 9 shows the expected cost of travel (on the vertical axis) from origin to the destination for different departure times (on the horizontal axis). We can observe a difference in the expected cost computed by both algorithms. The adaptive routing algorithm proposed in this paper shows a reduction in travel cost at various departure times in comparison to the offline path computed by the EV algorithm. The savings using adaptive routing algorithm is as high as 120 seconds around 8 A.M.

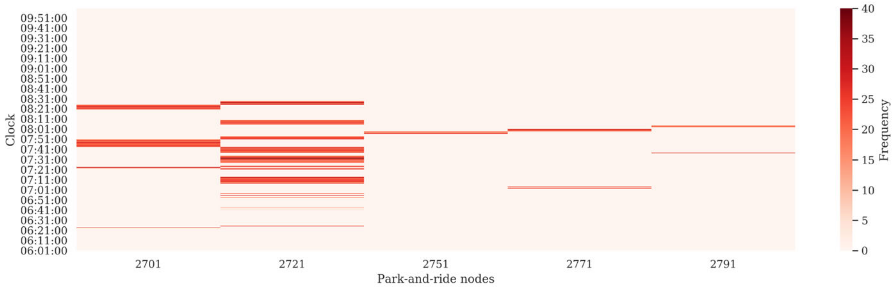


**Fig. 8** Frequency of use of different park-and-ride facilities out of 10,000 sample trajectories

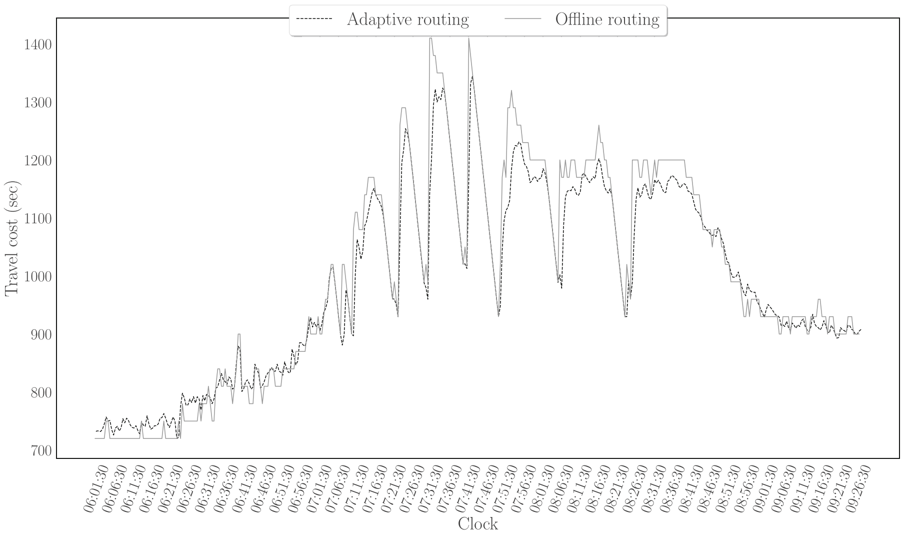


**Fig. 9** Comparison between adaptive and offline algorithm

### 6.2.4 Sensitivity Analysis

To see how parking cost $\tau_p$ at the destination location and transit fare $\tau_f$ affects the optimal policy, we performed a sensitivity analysis on these two parameters. The value of $\tau_f$ and $\tau_p$ was varied from \$1 − \$5 and \$0 − \$30 respectively. The transit fare cost is added twice as we also consider the cost of taking transit for the reverse commute. We calculated the percentage of times when park-and-option opted as a mode of travel in the optimal policy. The Figs. 10(a) and (b) depicts that the park-and-ride

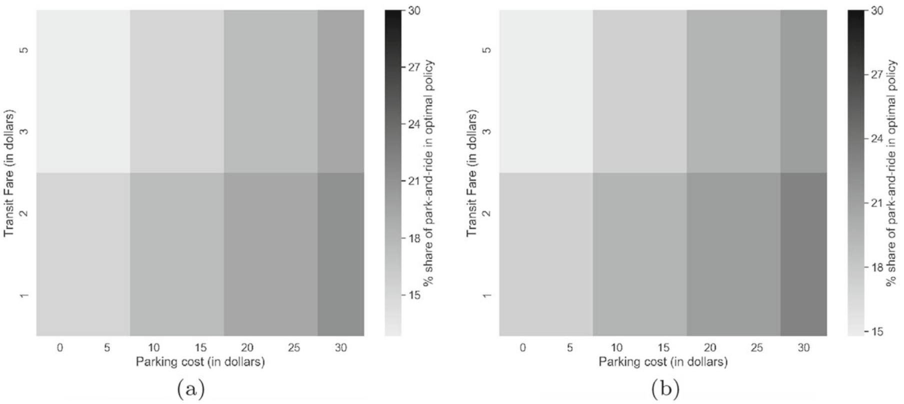


**Fig. 10** Sensitivity analysis on transit fare and parking cost

mode is attractive when the transit fare is low and parking cost is high. With no parking fee and $3-5 transit fare, we observe the lowest share of park-and-ride option in the optimal policy, i.e., 12.82% and 14.78% during uncongested and congested conditions respectively. On the other hand, with $30 parking fee and $1-2 transit fare, we observe the highest share of park-and-ride option in the optimal policy, i.e., 21.10% and 23.11% during uncongested and congested conditions respectively. We also performed a sensitivity analysis on the individual use of park-and-ride facilities for varying parking costs and transit fares. However, we did not find any significant change in the behavior of park-and-ride location choice.

## 7 Conclusions

The decision of park-and-ride is often made adaptively based on the realized state of traffic on a freeway. We modeled this phenomenon using a stochastic shortest path problem in a network with time-dependent and stochastic link travel times. The study was innovative as it not only provides the optimal policy in the auto network but also provides the optimal park-and-ride location choice, and transit itinerary to the traveler. The routing of the passenger in the transit network is performed using a schedule-based transit shortest path algorithm. This is a label setting algorithm that computes the optimal path for a passenger based on different weights assigned to different components of the transit network. The online shortest path problem, which is formulated as an MDP, evaluates the routing in the auto network. We proposed two different methods for solving the corresponding stochastic shortest path problem. The first method was the value iteration method that was proved to converge in the finite number of iterations for directed acyclic transition graphs for stationary policy obtained in this problem. The second method was the label correcting algorithm, which has a polynomial-time complexity and outperforms the value iteration method in terms of computational time.

The research also presents a case study of freeway I-394 in Minneapolis. Using historical loop detector data, the probability distribution of travel time on different links of the auto network was obtained. Based on the recommendations of the Highway Capacity Manual, the methodology was tested for two different states-congested and uncongested. The results computed the time and state of the realized traffic when park-and-ride mode becomes an attractive mode. For example, the park-and-ride mode provides faster access to the destination during severe congestion between 7:00-9:45 A.M. We expect to see further improvement in the travel time if more frequent service of buses is provided by Metro Transit on I-394. In terms of attractiveness among park-and-ride locations, Co. Rd. Park-and-Ride and Plymouth Road Transit Center were found to be more attractive parking locations than others. The standard deviation of travel time obtained by using the adaptive policy varies between 20-222 seconds. We also showed the benefit of online routing by comparing its expected cost with the offline algorithm. The online algorithm provides savings as high as 120 seconds in comparison to the offline algorithm. Finally, a sensitivity analysis on the parking cost at the destination location and transit fare was conducted. We found that the higher parking cost and lower transit fare make the park-and-ride mode more attractive to commuters. With \$2 increase in the transit fare at \$15 parking cost, we see a 2% decline in the share of park-and-ride option in the optimal policy. Similarly, with \$15 increase in the parking cost at \$2 transit fare, we see a 4% increase in the share of park-and-ride option in the optimal policy. We also found that by using adaptive policy, a commuter will save around 36 hours every year. Finally, the opportunity value of the research presented in this article will be high when this type of algorithm is implemented as an online cellphone application.

## 8 Limitations and Future Research

There are several assumptions made in this study, and the authors would like to throw some light on how to relax these assumptions in future research. First, the park-and-ride lot and bus capacity is assumed to be sufficient, but in congested systems, this may not be the case. The capacity of park-and-ride and transit vehicles can be incorporated into the current model. This requires augmentation of the state space by including the online information about finding a parking spot and availability of seats for a particular transit trip when arriving at a park-and-ride node. Depending upon the available capacity, there will be limited actions that will affect the following transitioning state. For example, if the parking lot is full, then the only action available is to take the freeway link further. Second, the travel time on the transit network is assumed to be reliable. This is because the possibility of missing transfers makes the transit online shortest path problem more complex. This complexity has been discussed in the literature by Rambha et al. (2016) and Khani (2019). Future research should look into ways of modeling this issue and efficiently solving it. Third, the routing passengers are assumed to be expected-cost-minimizers. However, they may consider other factors such as risk, preferences towards different modes, etc. as part of their utility. One way to address this issue is to consider an adaptive routing policy based on a discrete choice model. For example, Gao et al. (2008) studies adaptive route choice models for stochastic time-dependent auto networks. Fourth, we considered only one-way routing of a commuter, but in reality, nearly all commuters have to return home. The park-and-ride policy found for one-way routing may not provide an optimal tour. The problem can be addressed using the network transformation proposed by Nassir et al. (2012) for finding the multi-modal multi-destination tour. The development of an efficient algorithm for solving such a problem needs further research. For example, efficient reinforcement learning algorithms can be used to solve the problem with large state spaces. Finally, finding the optimal park-and-ride facility locations in the network considering adaptive routing will be another interesting research problem to investigate.

**Acknowledgements**

- This research is conducted at the University of Minnesota Transit Lab, currently supported by the following, but not limited to, projects:
    - National Science Foundation, award CMMI-1831140
    - Freight Mobility Research Institute (FMRI), Tier 1 Transportation Center, U.S. Department of Transportation: award RR-K78/FAU SP#16-532 AM2 and AM3
    - Minnesota Department of Transportation, Contract No. 1003325 Work Order No. 44 and 111
    - Minnesota Department of Transportation, Agreement No. 1044277.
- The authors are grateful to the Minnesota Department of Transportation and Metro Transit for sharing the data. Any limitation of this study remains the responsibility of the authors.
- The authors are grateful to anonymous TRB and NETS reviewers for their wonderful insights.

## Appendix A: EV Algorithm (Miller-Hooks and Mahmassani 2003)

The Expected Value (EV) algorithm generates *a priori* LET paths with their associated expected cost from all origins to a single destination for each possible departure time in a given time horizon. Let $S_{ij}(t) = \left\{s_{ij}^k\right\}_{k\geq 1}$ be the set of possible realizations of travel time on link $(i, j) \in N_a$ which happens with probability $\left\{p_{ij}^k\right\}_{k\geq 1}$ such that $\sum_k p_{ij}^k = 1$. Let us denote $\lambda_i^c(t)$ as the expected travel cost along path $c$ from node $i$ to the destination $d$ departing at time $t$. For each $t \in \mathcal{T}$, the minimum expected cost from each node is sought. In this label-correcting algorithm, we maintain a set of labels $\Lambda_i^c$. These labels are called pareto-optimal (or p-optimal) because each label is potentially optimal for one or more time intervals. Let $pathList[i]$ be the set of p-optimal paths from node $i$ to $d$. A scan eligible list $SE$ is maintained whose elements $(j, \mu)$ are characterized by a node $j \in N_a$ and path $\mu \in pathList[j]$. A set of path pointers $predNode$ and $predPath$ are also maintained to trace back the optimal path after the algorithm terminates.

At each iteration, an element $(j, \mu)$ is selected from $SE$ and for each neighboring node $i$ of $j \in N_a$, a temporary cost label $\{\kappa_i(t)\}_{t\in\mathcal{T}}$ is computed (Line 14). After this, we check for the Pareto-optimality of temporary labels (Line 16). If the new path is p-optimal, we add it to the $pathList[i]$ and update the optimal cost labels and path pointers. After the algorithm terminates, a single best path from each node and for each possible departure time is selected.

**Algorithm 4** EV Algorithm by Miller-Hooks and Mahmassani (2003).

1: **procedure** OFFLINEALGORITHM($G_a$)
2: **for** $t \in \mathcal{T}$ **do**
3: **for** $i \in N_a$ **do**
4: **for** $c \in \{1, 2, ..., M\}$ **do** ▷ M is large enough to store required p-optimal paths
5: $\lambda_i^c(t) = \infty$
6: $\lambda_d^1(t) = 0, \forall t \in \mathcal{T}$
7: $predNode[i, c] = \infty$, $predPath[i, c] = \infty, \forall i \in N_a, c \in \{1, 2, ..., M\}$
8: $pathList[d] = \{1\}$
9: $SE = \{(d, 1)\}$ ▷ Node-label pair
10: **while** SE **do**
11: Take first element $(j, \mu)$ out of $SE$
12: **for** $i \in \Gamma^{-1}(j)$ **do**
13: **for** $t \in \mathcal{T}$ **do**
14: $\kappa_i(t) = \sum_k [s_{ij}^k(t) + \lambda_j^\mu(t + s_{ij}^k(t))] p_{ij}^k(t)$
15: $predNode[i, temp] = j$, $predPath[i, temp] = \mu$
16: Check for pareto optimality. $\{\kappa_i(t)\}_{t\in\mathcal{T}}$ is pareto-optimal iff $\nexists c \in pathList[i]$ such that $\lambda_i^c(t) \leq \kappa_i(t), \forall t \in \mathcal{T}$ and $\exists t \in \mathcal{T}$ such that $\lambda_i^c(t) < \kappa_i(t)$.
17: **if** $\{\kappa_i(t)\}_{t\in\mathcal{T}}$ is pareto-optimal **then**
18: Add this path to $pathList[i]$
19: Fix the path pointers in step 15
20: Update the minimum cost label for each $t \in \mathcal{T}$ from node $i$.
21: **else**
22: Discard this path
23: Check if all the paths in $pathList[i]$ are still p-optimal, otherwise remove them
24: Evaluate the minimum cost label for each $t \in \mathcal{T}$ from node $i$ using the available paths in $pathList[i], \forall i \in N_a$

## Appendix B: Proofs of Various Propositions

*Proof (Proposition 1)* Lines 2-4 can be done in $\mathcal{O}(1)$ time. Assuming that the shortest path algorithm is implemented using Binary heap data structure, Line 5-13 consists of two major steps, finding the node $i$ with minimum label $\gamma^{gc}$ from SEL (Line 6) that can be done in $\mathcal{O}(|N_t|\log(|N_t|))$ and updating labels of new nodes which can be done in $\mathcal{O}(|A_t|\log(|N_t|))$ time. Therefore, the worst case computational complexity of Algorithm 1 is $\mathcal{O}(|A_t|\log|N_t| + |N_t|\log|N_t|))$. However, if $|A_t| = \Omega(|N_t|)$, then the worst case computational complexity of Algorithm 1 can be given as $\mathcal{O}(|N_t|\log|N_t|))$. □

*Proof (Proposition 2)* Without the loss of generality, let us assume that there exists a cyclic path of minimum length 2. Let $\mathcal{P}\{[(i,t,\theta),(j,t',\theta')],[(j,t',\theta'),(i,t,\theta)]\}$ be that path. Since, $t' = c_{ij}^{\theta} + t,\ \forall i, j \neq d$, implying that $t' + c_{ji}^{\theta'} = t$, which is a contradiction unless $c_{ij}^{\theta} = c_{ji}^{\theta'} = 0$, in which case there does not exist such transition. Hence, there does not exist a cyclic path of length 2. One can extend this argument for a cyclic path of any length. □

*Proof (Proposition 3)* To show this, let us consider various subsets of the state space created as below:

$$S'_0 = \{(d,t) : t \in \mathcal{T}\} \tag{13}$$

$$S'_{k+1} = \left\{ (i,t) : \sum_{\substack{\theta:\pi^*(i,t,\theta)=j,\\ \theta'\in\Theta_j\left(t+c_{ij}^{\theta}\right)}} \mathbb{P}_{\pi}\left[(i,t,\theta),(j,t+c_{ij}^{\theta},\theta')\right] = 0, \forall j \notin \cup_{m=0}^{k} S_m \right\}, k = 0,1,... \tag{14}$$

The above construction of sets adds various states in the backward direction of the destination node. For example, $S_1$ will contain all the states associated to the nodes in the sets $Z$ and $\Gamma^{-1}(d)$. Let $S_{\bar{k}}$ be the last of these sets that is non-empty. In view of the acyclicity and proper stationary optimal policy assumptions, we have $\bar{k} \leq |N_a|$ and $\cup_{m=0}^{\bar{k}} S_m = S'$. After this, one can show using induction that

$$(\hat{T}^k \hat{J})(i,t) = \hat{J}^*(i,t),\ \forall (i,t) \in \cup_{m=0}^{k} S_m, k = 1,...,\bar{k} \tag{15}$$

The mathematical induction part is same as the proof given in Bertsekas (2012). □

*Proof (Proposition 4)* Lines 2-5 can be done in $\mathcal{O}(1)$ time. The control computation in line 11 can be performed in $\mathcal{O}(|A_a|)$. The previous operation should be performed for every $(i,t,\theta) \in S$, repeatedly $|N_a|$ number of times. Therefore, lines 6-11 can be performed in $\mathcal{O}(|S||N_a||A_a|)$. Furthermore, lines 12-15 can be performed in $\mathcal{O}(|S||A_a|)$. Therefore, the overall complexity of the Algorithm 2 is equal to $\mathcal{O}(|S||N_a||A_a|)$. □

*Proof (Proposition 5)* Assuming that the elements of $SE$ are removed according to the FIFO rule, lines 1-15 are standard Bellman-Ford algorithm and can be performed in $\mathcal{O}(|S||A_a|)$ time. The computational complexity of finding the optimal policy (lines 16-19) can be performed in $\mathcal{O}(|S||A_a|)$. Therefore, the overall complexity of the Algorithm 3 is equal to $\mathcal{O}(|S||A_a|)$. □

## References

Ahuja RK, Magnanti TL, Orlin JB (1988) Network flows-theory, algorithms and applications. Pearson; 1st edition

Andreatta G, Romeo L (1988) Stochastic shortest paths with recourse. Networks 18(3):193–204

Belenky P (2011) Revised departmental guidance on valuation of travel time in economic analysis. Tech. rep., U.S. Department of Transportation. https://www.transportation.gov/sites/dot.dev/files/docs/vot%_guidance_092811c.pdf

Bertsekas DP (2012) Dynamic programming and optimal control, 4th edn. Athena Scientific, Belmont, Massachusetts. www.athenasc.com/dpbook.html

Boyles SD, Rambha T (2016) A note on detecting unbounded instances of the online shortest path problem. Networks 67(4):270–276

Boyles SD, Waller ST (2011) Optimal information location for adaptive routing. Netw Spatial Econ 11(2):233–254. https://doi.org/10.1007/s11067-009-9108-9

Cheung RK (1998) Iterative methods for dynamic stochastic shortest path problems. Naval Res Log 45(8):769–789. https://doi.org/10.1002/(sici)1520-6750(199812)45:8⟨769::aid-nav2⟩3.0.co;2-%23

Chriqui C, Robillard P (1975) Common bus lines. Trans Sci 9(2):115–121

Croucher JS (1978) A note on the stochastic shortest-route problem. Naval Res Log Quart 25(4):729–732

Ding-Mastera J (2016) ScholarWorks@UMass amherst adaptive route choice in stochastic time-dependent networks: routing algorithms and choice modeling (March). https://scholarworks.umass.edu/dissertations_2

Gao S (1999) Optimal adaptive routing and traffic assignment in stochastic time-dependent networks. October :1–237. http://www.ecs.umass.edu/\simsonggao/Gao_PhD.pdf

Gao S, Chabini I (2006) Optimal routing policy problems in stochastic time-dependent networks. Transp Res B Methodol 40(2):93–122. https://doi.org/10.1016/j.trb.2005.02.001

Gao S, Huang H (2012) Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks, vol 21. https://doi.org/10.1016/j.trc.2011.09.007

Gao S, Frejinger E, Ben-Akiva M (2008) Adaptive route choice models in stochastic time-dependent networks. Transportation Research Record (2085) 136–143. https://doi.org/10.3141/2085-15

Hall RW (1986) The fastest path through a network with random time-dependent travel times. Transp Sci 20(3):182–188. https://doi.org/10.1287/trsc.20.3.182

Jafari E, Boyles SD (2017a) Multicriteria stochastic shortest path problem for electric vehicles. Netw Spat Econ 17(3):1043–1070. https://doi.org/10.1007/s11067-017-9358-x

Jafari E, Boyles SD (2017b) Online charging and routing of electric vehicles in stochastic time-varying networks. Trans Res Record J Transp Res Board 2667(1):61–70. https://doi.org/10.3141/2667-07

Khani A (2019) An online shortest path algorithm for reliable routing in schedule-based transit networks considering transfer failure probability. Trans Res Part B Method 126:549–564

Khani A, Boyles SD (2015) An exact algorithm for the mean–standard deviation shortest path problem. Trans Res Part B Methodolog 81:252–266. https://doi.org/https://doi.org/10.1016/j.trb.2015.04.002. http://www.sciencedirect.com/science/article/pii/S0191261515000703

Khani A, Lee S, Hickman M, Noh H, Nassir N (2012) Intermodal path algorithm for time-dependent auto network and scheduled transit service. Transp Res Record J Transp Res Board 2284(1):40–46. https://doi.org/10.3141/2284-05

Khani A, Hickman M, Noh H (2014) Trip-based path algorithms using the transit network hierarchy. Netw Spat Econ 15(3):635–653. https://doi.org/10.1007/s11067-014-9249-3

Khani A, Hickman M, Noh H (2015) Trip-based path algorithms using the transit network hierarchy. Netw Spat Econ 15(3):635–653

Kumar P, Khani A (2019) prameshk/GTFS-to-schedule-based-transit-network: GTFS to schedule-based transit network. https://doi.org/10.5281/zenodo.2531085

Kumar P, Khani A (2020) Evaluating special event transit demand: A robust principal component analysis approach. IEEE Trans Intell Transp Syst pp 1–13. https://doi.org/10.1109/TITS.2020.3001470. https://ieeexplore.ieee.org/document/9130892/

Kumar P, Khani A (2021) An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access. Trans Res Part C Emerg Tech 122:102891. https://doi.org/10.1016/j.trc.2020.102891

Kumar P, Khani A, He Q (2018) A robust method for estimating transit passenger trajectories using automated data. Trans Res Part C Emerg Technol 95. https://doi.org/10.1016/j.trc.2018.08.006

Kumar P, Khani A, Davis GA (2019) Transit route origin–destination matrix estimation using compressed sensing. Transportation Research Record. https://doi.org/10.1177/0361198119845896

Kumar P, Khani A, Lind E, Levin J (2021) Estimation and mitigation of epidemic risk on a public transit route using automatic passenger count data. Transportation Research Record. https://doi.org/10.1177/03611981209851333

Levin MW, Boyles SD (2019) Optimal guidance algorithms for parking search with reservations. Netw Spat Eco https://doi.org/10.1007/s11067-019-09464-7

Liu Y, Blandin S, Samaranayake S (2019) Stochastic on-time arrival problem in transit networks. Transportation Research Part B: Methodological 119:122–138 https://doi.org/10.1016/j.trb.2018.11.013. https://www.sciencedirect.com/science/article/pii/S0191261518306702?dgcid=rss_sd_all

Miller-Hooks E (2001) Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. Networks 37(1):35–52. https://doi.org/10.1002/1097-0037(200101)37:1⟨35::AID-NET4⟩3.0.CO;2-G
Miller-Hooks ED, Mahmassani HS (2003) Least expected time paths in stochastic, time-varying transportation networks. Transp Sci 34(2):198–215. https://doi.org/10.1287/trsc.34.2.198.12304
Minnesota Department of Transportation (2019) Mn/DOT Traffic Data. http://data.dot.state.mn.us/datatools/
Mirchandani PB, Soroush H (1985) Optimal paths in probabilistic networks: A case with temporary preferences. Comput Oper Res 12(4):365–381
Nassir N, Khani A, Hickman M, Noh H (2012) Algorithm for Intermodal Optimal Multidestination Tour with Dynamic Travel Times. Trans Res Record J Transport Res Board 2283:57–66. https://doi.org/10.3141/2283-06. http://trrjournalonline.trb.org/doi/10.3141/2283-06
Nguyen S, Pallottino S (1988) Equilibrium traffic assignment for large scale transit networks. Eur J Oper Res 37(2):176–186. https://doi.org/10.1016/0377-2217(88)90327-X
Nie YM, Wu X (2009) Shortest path problem considering on-time arrival probability. Transp Res B Methodol 43(6):597–613
Pang H, Khani A (2018) Modeling park-and-ride location choice of heterogeneous commuters. Transportation 45(1):71–87. https://doi.org/10.1007/s11116-016-9723-5
Polychronopoulos GH, Tsitsiklis JN (1996) Stochastic shortest path problems with recourse. Netw Int J 27(2):133–143
Provan JS (2003) A polynomial-time algorithm to find shortest paths with recourse. Networks 41(2):115–125. https://doi.org/10.1002/net.10063
Psaraftis HN, Tsitsiklis JN (1993) Dynamic shortest paths in acyclic networks with Markovian arc costs. Oper Res 41(1):91–101
Rambha T, Boyles SD, Waller ST (2016) Adaptive transit routing in stochastic time-dependent networks, vol 50. https://doi.org/10.1287/trsc.2015.0613. http://pubsonline.informs.org/doi/10.1287/trsc.2015.0613
Rambha T, Boyles SD, Unnikrishnan A, Stone P (2018) Marginal cost pricing for system optimal traffic assignment with recourse under supply-side uncertainty. Transp Res B Methodol 110:104–121
Shepardson D, Carey N (2018) U.S. vehicle fuel economy rises to record 24.7 mpg: EPA. https://www.reuters.com/article/us-autos-emissions/u-s-vehicle-fuel-eco
Spiess H, Florian M (1989) Optimal strategies: A new assignment model for transit networks. Transp Res B 23(2):83–102. https://doi.org/10.1016/0191-2615(89)90034-9
Tang S, Rambha T, Hatridge R, Boyles SD (2014) Modeling parking search on a network by using stochastic shortest paths with history dependence. https://doi.org/10.3141/2467-08
Tong C, Richardson A (1984a) A computer model for finding the time-dependent minimum path in a transit system with fixed schedules. J Adv Transp 18:145–161
Tong CO, Richardson AJ (1984b) A computer model for finding the time-dependent minimum path in a transit system with fixed schedules. J Adv Transp 18(2):145–161
Unnikrishnan A, Lin DY (2012) User equilibrium with recourse: Continuous network design problem. Comput Aid Civil Infrast Eng 27(7):512–524. https://doi.org/10.1111/j.1467-8667.2011.00753.x
Unnikrishnan A, Waller ST (2009) User equilibrium with recourse. Netw Spat Econ 9(4):575–593. https://doi.org/10.1007/s11067-009-9114-y
Waller ST, Ziliaskopoulos AK (2002) On the online shortest path problem with limited arc cost dependencies. Networks 40(4):216–227. https://doi.org/10.1002/net.10049
Webb A, Khani A (2020) Park-and-ride choice behavior in a multimodal network with overlapping routes. Transp Res Rec 2674(3):150–160. https://doi.org/10.1177/0361198120908866
Webb A, Kumar P, Khani A (2020) Estimation of passenger waiting time using automatically collected transit data. Pub Trans 12(2):299–311. https://doi.org/10.1007/s12469-020-00229-x
Zhang Y, Khani A (2019) An algorithm for reliable shortest path problem with travel time correlations. Trans Res Part B Methodological 121:92–113. https://doi.org/10.1016/j.trb.2018.12.011. http://www.sciencedirect.com/science/article/pii/S0191261518302959