mathematics

MDPI

Article

Non-Aggressive Adaptive Routing in Traffic

Madhushini Narayana Prasad [1,*], Nedialko Dimitrov [1] and Evdokia Nikolova [2]

[1] Department of Operations Research and Industrial Engineering, The University of Texas at Austin, 204 E. Dean Keeton Street, C2200, Austin, TX 78712, USA; ned@austin.utexas.edu

[2] Department of Electrical and Computer Engineering, The University of Texas at Austin, 2501 Speedway, Austin, TX 78712, USA; nikolova@autsin.utexas.edu

* Correspondence: madhushini@utexas.edu

Abstract: Routing a person through a traffic road network presents a tension between selecting a fixed route that is easy to navigate and selecting an aggressively adaptive route that minimizes travel time. In this paper, we propose a novel routing framework that strikes a balance between adaptability and simplicity. Specifically, we propose to create non-aggressive adaptive routes that seek the best of both these extremes in the navigation world. These selected routes still adapt to changing traffic conditions, but we limit the number of adjustments made en route. This framework improves the driver experience by providing a continuum of options between saving travel time and reducing navigation stress. We design strategies to model single and multiple route adjustments, and investigate numerous techniques to solve these models for better route selection. To alleviate the intractability of handling real-life traffic data, we devise efficient algorithms with easily computable lower and upper bounds. We finally perform computational experiments on our algorithms to demonstrate the benefits of limited adaptability in terms of reducing the travel time.

Keywords: adaptive routing; dynamic programming; directed acyclic graphs; shortest paths

MSC: 90C39

check for updates

Citation: Prasad, M.N.; Dimitrov, N.; Nikolova, E. Non-Aggressive Adaptive Routing in Traffic. *Mathematics* **2023**, *11*, 3639. https://doi.org/10.3390/math11173639

Academic Editors: Shi Qiang Liu, Erhan Kozan, Felix T. S. Chan and Weidong Li

Received: 11 May 2023

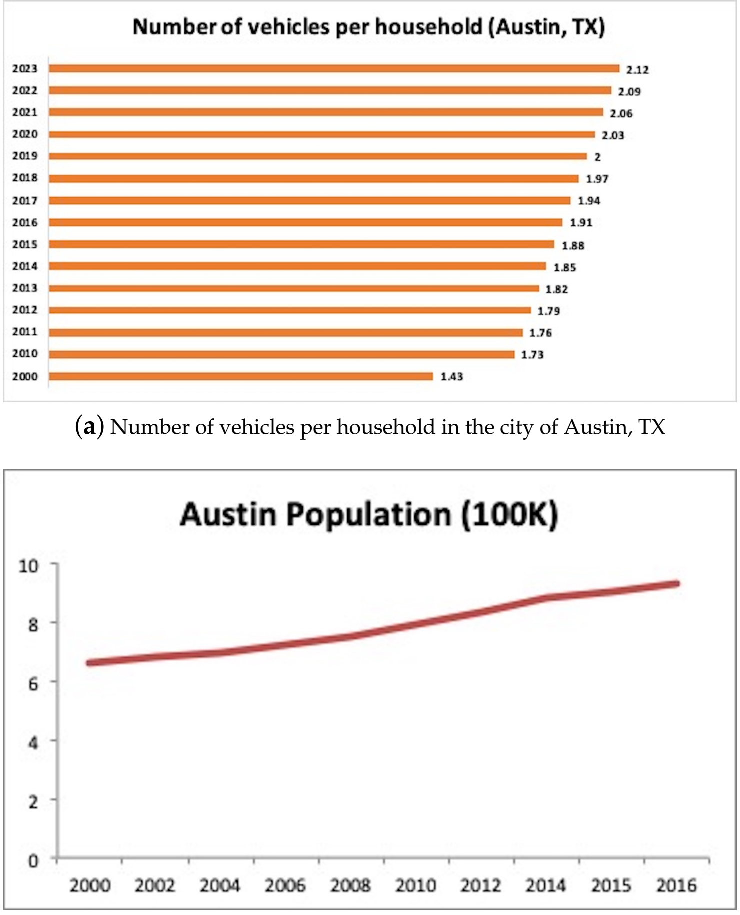
Revised: 16 August 2023

Accepted: 19 August 2023

Published: 23 August 2023

1. Introduction

Traffic congestion is an unprecedented problem in many major cities in the United States. The number of vehicles on road has increased significantly in recent years, leading to increased congestion. In year 2021, there were an estimated 282 million registered vehicles in the United States, up from 225 million in year 2000 [1]. According to data from the Census Bureau [2], there were about 1.8 vehicles available per US household in year 2016. These number of vehicles per household varies depending on the city and its population density. For example, in Austin, Texas, the population in year 2016 was 925 K [3] and there were 1.9 vehicles per household in year 2016 [2]. Figure 1a,b show that the number of vehicles per household and population density in Austin has been steadily increasing in recent years [1,3]. In year 2000, the average number of vehicles per household in Austin was 1.43. This number has increased by 48.2% in the past 23 years, up to 2.12 in year 2023. On the other hand, the population in Austin is also expected to increase by at least 30% by year 2030 [4]. This will lead to an even greater increase in the number of vehicles on road, and will further exacerbate the problem of traffic congestion. According to an INRIX report [5] and a congestion statistics report [6], due to poor traffic conditions, Austin-area commuters wasted about 41 h on average in traffic in year 2020 (3 hours more than in year 2012), the seventh-highest of any city in the country, thereby increasing the overall travel time by 22%. The problem of traffic congestion is expected to worsen due to future predicted population growth, which can lead to significant increase in travel times, decreased fuel efficiency, and increased air pollution. It can also have a negative impact on businesses and city's economy. One way to alleviate traffic congestion is to use efficient routing strategies. These strategies can help drivers find the fastest and most efficient routes to their destinations, and help drivers avoid congested areas.



(**a**) Number of vehicles per household in the city of Austin, TX

(**b**) Austin population growth

Figure 1. (**a**) Average vehicle density per household in Austin city. (**b**) Population history of Austin city.

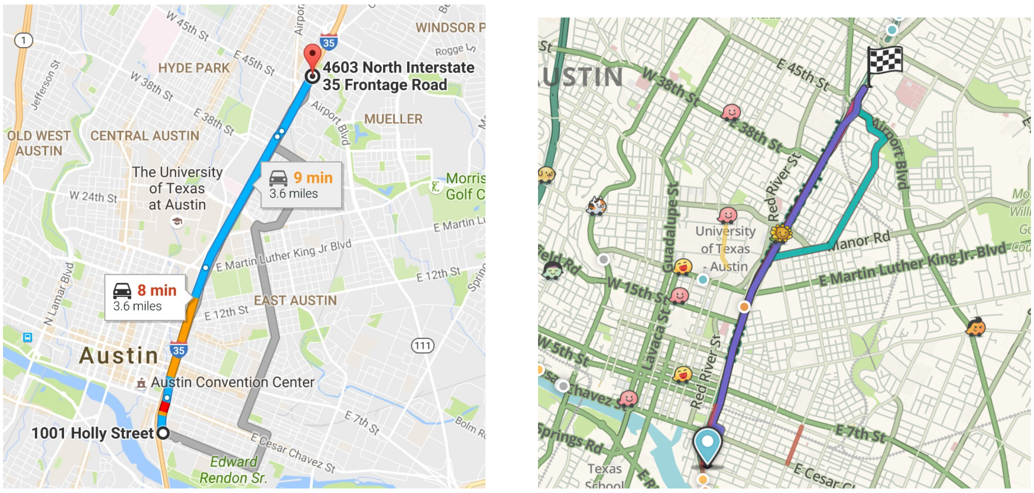
There are various strategies and tools currently available to develop optimized routes. For example, Google Maps and Waze route in different ways and serve a different clientele. Google Maps creates a fixed static route which is easier to navigate but could be potentially slow. On the other extreme, Waze provides an aggressive adaptive route. A snapshot of routes generated between a specific source–destination pair using Google Maps and Waze are shown in Figure 2a and Figure 2b, respectively. It is to be noted that the latter recommends a recourse path en route to the destination, which is depcited in Figure 2b with teal color. An (aggressive) adaptive route by Waze is a potentially faster route that dynamically changes and adapts to traffic conditions, but the frequent route changes may lead to high stress in navigation. To alleviate this issue and create a middle-ground that seeks the best of both extremes, in this work, we develop a methodology to compute non-aggressive adaptive routes.

A non-aggressive adaptive route adapts dynamically to changing traffic conditions but in a limited way—for example, by allowing only a certain number of route-shifts at critical junctures. These routes seek to provide both low travel times and low stress of navigation. At the start of the route, the conditions on the roads are only known through a probability distribution. As the driver approaches closer to individual intersections, specific road conditions are observed and the routes are adjusted to minimize the travel time.

The contributions of this paper can be summarized as follows:

1. We propose a novel routing framework for routing in road networks with limited adaptability. We call this *non-aggressive adaptive* routing, aiming to seek both low travel times and low navigation stress compared to traditional routing technologies like Google Maps and Waze;
2. We design several strategies to model and compute such non-aggressive adaptive routes, based on where and how the route adjustments are performed;
3. We develop exact mathematical methods such as complete enumeration and dynamic programming algorithms for each of the strategies. We also derive easily computable bounds that can be used to solve these algorithms efficiently for large networks;
4. We evaluate and analyze the performance of proposed algorithms using the Austin road network as an example and demonstrate the benefits of limited adaptability.

The remainder of the paper is organized as follows. In Section 2, we discuss the related work on adaptive routing. In Section 3, we present our proposed routing strategies and the respective solution methodologies. In Section 4, we develop efficient bounds to render our algorithms tractable for large sized networks, and also present the computational evaluation of our proposed tractable algorithms using Austin road network. Finally, Section 5 concludes our work by summarizing the key contributions and pointing to future research directions.



(**a**) Google maps: An example (**b**) Waze: An example

Figure 2. Sample routes generated using Google Maps and Waze on the Austin road network.

2. Related Work

Consider the case of routing a driver from point s to t in a traffic network. Adaptive routing is a stochastic shortest path problem where the edge costs are unknown until arriving at one of its endpoints. The decision to continue or change the route is based on the traffic condition at that edge. Croucher [7] appears to be the first to have studied a model of this type but in a fairly restricted setting. In that model, a first-choice arc is selected for every node, there is some probability that the arc fails, and, if it fails, a second outgoing arc is selected at random. Andretta and Romeo [8] considered a similar model with the choice of recourse computed in an optimal way. In their work, a recourse path to the destination is computed for every edge, assuming that the edge is inactive. In our work, if an edge has traffic congestion, it is still considered active with greater time delay for traversal. Additionally, if an edge is selected for observation and found to be congested, the driver may revert to a recourse route. Unlike the past literature, our work describes a sequence of models in which the driver may observe between one and all edges for traffic congestion.

Another widely studied variant of adaptive routing is the Canadian Traveller Problem (CTP). CTP was first defined in [9] (see also [10]). The goal is to find an optimal routing policy that guarantees a good route under uncertain road conditions, minimizing the expected cost of travel. In this problem, the arc costs are deterministic but unknown and, once a road is considered blocked, it remains blocked forever. In general, CTP is known to be #P-hard and there has been no significant progress on approximation algorithms. Several variants to this problem, such as $k-$CTP, $k-$vital edges problem, and deterministic and stochastic recoverable CTP, are defined in [11]. Polychronopoulos and Tsitsiklis [12] presented another variation to CTP where the realization of arc costs is learned progressively as the graph is traversed. They provided dynamic programming algorithms to solve models with both dependent and independent arc costs and they established that the running time of these algorithms is exponential in number of arcs. There are few other algorithms that address the adaptive routing problem with time-dependent and stochastic costs [13–16]. In our work, we assume independent arc costs and limit the number of re-routing decisions, as opposed to CTP and its variants. We also present tractable dynamic programming algorithms which are solvable in polynomial time.

Special cases of CTP were studied by Nikolova and Karger [17] to explore exact solutions. They explained the connection of CTP to Markov Decision Processes (MDPs) solvable in polynomial time. They also presented polynomially solvable dynamic programming algorithms for a standard version of CTP on directed acyclic graphs (DAGs). It is important to note that our problem is a generic version of CTP. CTP can be derived by equating the number of re-routing decisions to the total number of edges in the network, which, in one of our proposed routing strategies, we call the *parallel model*.

The research presented here complements these works by examining non-aggressive adaptive routing, identifying an optimal yet small number of decision points on a route. The focus of our work is to derive the benefits of adaptive routing but with a limited number of adaptations to reduce driving stress. To this end, we propose, compare, and contrast several models for defining the decision points, and develop tractable algorithms to compute the optimal routing policy.

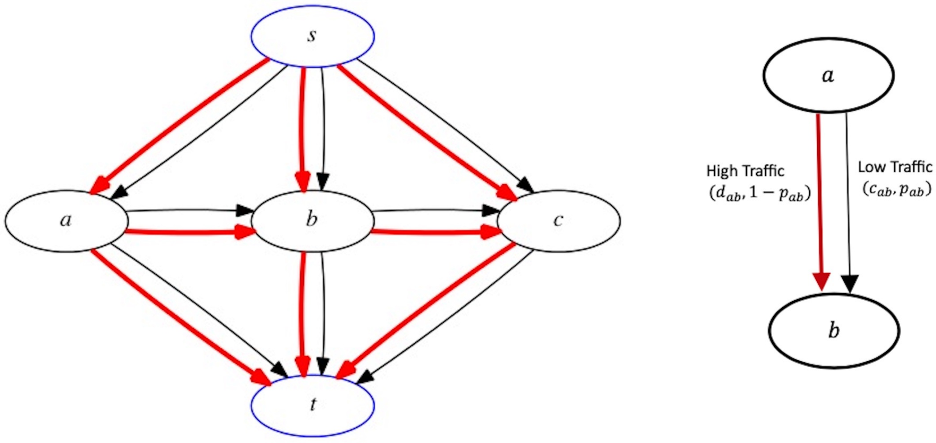
Many other recent extensions to adaptive routing have been proposed, primarily focusing on route planning under uncertainty for different modes of transportation [18–20], route planning under uncertainty with optimal information location points [21], stability of transportation networks [22], stochastic time-dependent networks [23], maximizing the probability of on-time arrival [24], application to online decision problems [25], and competitive analysis of CTP [26].

Another emerging field in adaptive routing models is the development of connected automated vehicles (CAV) technology [27,28], vehicle-to-vehicle (V2V) commnuication [29], and automonous vehicle tracking [30,31], with a variation to consider user preferences [32]. CAVs can provide real-time information about the individual position and velocity of vehicles, which can be used to develop more accurate and efficient routing algorithms. However, in this work, we focus on macroscopic routing models with high-level traffic congestion and delay information. We believe that, by understanding macroscopic traffic patterns, we can develop better routing algorithms for both CAVs and non-CAVs in the future.

3. Model Description

Consider a directed acyclic network $G = (N, A)$, with specified source s and destination t nodes, as shown in Figure 3a. We believe that directed acyclic networks (DAGs) are a more general representation of real-world traffic networks and best suited to generate practical recommendations to users. On a city road network G, N represents the set of road intersections and A represents the set of roads or edges connecting those intersections. We consider potential traffic congestion on the edges given by the set A.

We examine a simple model of traffic congestion where each edge is in either a high traffic state or low traffic state, independently of other edges. The traffic probability distribution is assumed to be known ahead of time. Every edge $e = (a, b)$ is defined by three inputs: $e = (c, d, p)$, where c_{ab} represents the travel time under low traffic, d_{ab} represents the travel time due to high traffic, and p_{ab} represents the probability of low traffic on the edge. This is visually depicted in Figure 3b.



(**a**) A sample network (**b**) Traffic Model

Figure 3. Two-state traffic model. Blue nodes represent the source and destination nodes. Red solid line indicates the possibility of high traffic on an edge, for example, edge (a, b) with probability $1 - p_{ab}$ and travel time d_{ab}. Black solid line indicates the possibility of low traffic with probability p_{ab} and travel time c_{ab}.

Given these inputs, we determine the edges to be observed for traffic congestion and the corresponding adjustment routes should high traffic states be observed on those edges. We call an edge selected for observation and for possible route adjustment an *adjustment edge*. When the driver reaches the source node of an adjustment edge and observes low traffic, they proceed through the edge. If the driver observes high traffic, then they take an *adjustment route*. To simplify the exposition, we start with a single route adjustment and then provide several extensions to multiple route adjustments. A detailed discussion on these route adjustment strategies is presented in the following subsections.

We begin with a simple example presented in Figure 4. This example shows that the optimal adjustment edge need not be a part of the fixed non-adaptive shortest route. The shortest path from s to t can be computed as $s \rightarrow b \rightarrow t$ with expected travel time 10. If edge (a, t) is observed, there is 20% chance of low traffic with zero travel time. However, there is 80% chance of high traffic at edge (a, t), and, if the driver adjusts the route to $a \rightarrow b \rightarrow t$, then the travel time is 11. With the single observation of edge (a, t), the expected travel time is $11 \cdot 0.8 + 0 \cdot 0.2 = 8.8$, which is lower than the expected travel time without any adjustments (which is equal to 10). An interesting aspect of this example is that the edge (a, t) is not on the fixed (no-adjustment) shortest path.

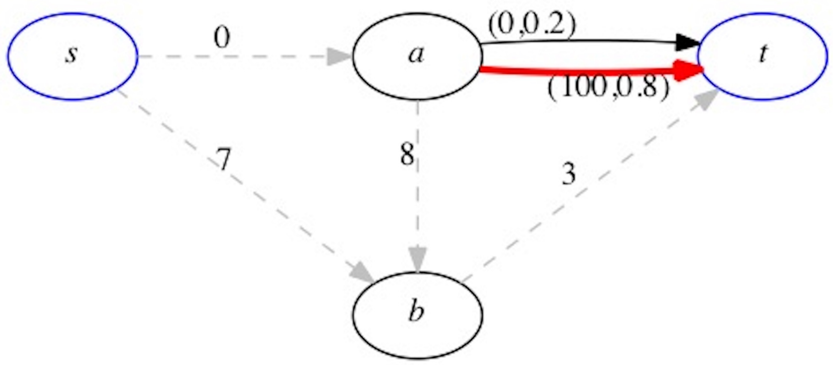


Figure 4. An example to show that an adjustment edge need not be a part of the fixed shortest route. Edge weights represent expected travel time with probability of low traffic assumed to be 1.0 on all the edges in dotted line, except edge (a, t) in solid line. At edge (a, t), we assume p_{at} as 0.2, c_{at} as 0, and d_{at} as 100, with expected travel time 80.

3.1. Single Route Adjustment Policy

A pictorial representation of a single route adjustment policy is shown in Figure 5, where the route from source s to destination t has a single adjustment edge, (u, v). In this policy, the driver takes the shortest path from s to u, and observes edge (u, v) for traffic. In case of low traffic, driver continues on the edge (u, v) and takes the shortest path from v to t. In case of high traffic, driver takes an adjustment route from u to t. The overall expected travel time for any adjustment edge (u, v) is computed using

$$\mathbb{E}_1[(u,v)] = \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}], \tag{1}$$

where $\mathbb{E}[i \to j]$ represents the expected travel time of a no-adjustment shortest path from node i to j, $\mathbb{E}[i \to j|d_{ik}]$ represents the expected travel time of a no-adjustment shortest path given edge (i, k) is congested, and $\mathbb{E}_1[(u, v)]$ represents the expected travel time of a single route adjustment policy using the adjustment edge (u, v). One could determine the adjustment edge that yields minimal expected travel time, $\arg\min_{(u,v)} \mathbb{E}_1[(u, v)]$, using complete enumeration, given by

$$\begin{aligned} Z_1[s \to t] =& \min\{\mathbb{E}[s \to t]; \\ & \min_{(u,v)\in A} \mathbb{E}_1[(u,v)]\}, \end{aligned} \tag{2}$$

where $Z_1[s \to t]$ represents the overall minimum expected travel time from s to t due to the single-route adjustment policy. An equivalent integer programming formulation is presented in Appendix A.1.

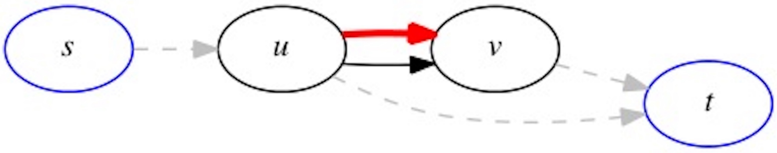


Figure 5. Single-route adjustment policy. Solid black line represents an edge and solid red line represents an adjustment edge. Grey dotted lines represent the shortest paths between nodes, with expected travel time as edge lengths.

3.2. Multiple Route Adjustment Policy

There are several potential models for multiple route adjustments. We present and explore three different strategies that we call the *series unforced*, *series forced*, and *parallel* models. We develop dynamic programming-based algorithms to solve these route adjustment models, and, finally, compare their performances.

3.2.1. Series Unforced Model

Let us start with two adjustment edges as shown in Figure 6, which follow what we call a *series unforced* model. In this model, once the driver makes a route adjustment, he loses the potential to observe the other edges for traffic. Say, for instance, the source s and destination t nodes are connected by a highway. The driver enters the highway from source s, and, upon arriving at u_1, observes edge (u_1, v_1) for traffic. In case of high traffic, driver adjusts the route to reach the destination t and never gets to make any other route adjustments. In case of low traffic, driver traverses the edge (u_1, v_1), and continues on the highway until u_2 where they observe edge (u_2, v_2) for traffic. In case of high traffic at (u_2, v_2), driver adjusts the route to destination t. In case of low traffic, driver traverses the edge (u_2, v_2), and continues on the highway to reach the destination t.

Let $\mathbb{E}_{\text{suf}}[(u_1, v_1), (u_2, v_2)]$ denote the expected travel time with respect to the adjustment edges (u_1, v_1) and (u_2, v_2). One could find a pair of edges that yield a minimum expected travel time through complete enumeration, $\arg\min_{(u_1,v_1)(u_2,v_2)} \mathbb{E}_{\text{suf}}[(u_1, v_1), (u_2, v_2)]$, using

$$
\begin{aligned}
&\mathbb{E}_{\text{suf}}[(u_1,v_1),(u_2,v_2)] = \\
&\qquad \mathbb{E}[s \to u_1] + (1-p_{u_1v_1})\mathbb{E}[u_1 \to t|d_{u_1v_1}] \\
&\qquad\qquad + p_{u_1v_1}\Big\{ c_{u_1v_1} + \mathbb{E}[v_1 \to u_2] + p_{u_2v_2}[c_{u_2v_2} + \mathbb{E}[v_2 \to t]] \\
&\qquad\qquad\qquad + (1-p_{u_2v_2})\mathbb{E}[u_2 \to t|d_{u_2v_2}]\Big\}. \qquad (3)
\end{aligned}
$$

The first summand is the expected travel time from s to u_1. The second summand is the expected travel time from u_1 to t if high traffic is observed at (u_1, v_1). The third summand is the travel time from u_1 to t if low traffic is observed at (u_1, v_1). This third summand includes within it a version of (1), computing the travel time from v_1 to t that is dependent on the observation of traffic at edge (u_2, v_2). Similarly, one could express the computation of the minimum expected travel time with k adjustment edges recursively with the equation for $k-1$ adjustment edges as follows. Let $Z_k^{\text{suf}}[s \to t]$ denote the overall minimum expected travel time when k adjustment edges are observed for traffic. We can then write

$$
\begin{aligned}
Z_1^{\text{suf}}[s \to t] &= Z_1[s \to t], \text{and} \\
Z_k^{\text{suf}}[s \to t] &= \min\Big\{ Z_{k-1}^{\text{suf}}[s \to t]; \\
&\quad \min_{(u,v)\in A} \big[\mathbb{E}[s \to u] + (1-p_{uv})\mathbb{E}[u \to t|d_{uv}] \\
&\qquad + p_{uv}(c_{uv} + Z_{k-1}^{\text{suf}}[v \to t])\big]\Big\}. \qquad (4)
\end{aligned}
$$

The basecase $Z_1^{\text{suf}}[s \to t]$ represents the minimum expected travel time for a single adjustment edge. The recursive equation to compute $Z_k^{\text{suf}}[s \to t]$ includes a $Z_{k-1}^{\text{suf}}[s \to t]$ in case it is unnecessary to observe k edges. The second term involves picking the first edge (u, v) for observation, and the remaining length of the paths to destination is based on the probabilities of that observation.

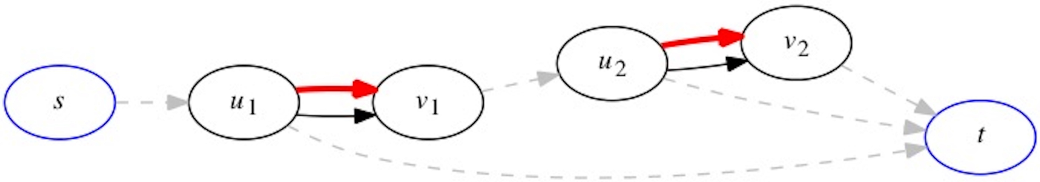


Figure 6. Series unforced model with two adjustment edges

The recursive equation (4) yields a dynamic programming algorithm for computing the best set of k adjustment edges. The dynamic programming algorithm reduces the computational effort from $O(m^k)$, roughly what is required with complete enumeration using (3), to $O(mk)$, where $m = |A|$. This brings significant computational savings, though, as is discussed later, this is still insufficient for practical applications.

3.2.2. Series Forced Model

An alternate model with two adjustment edges, which we call *series forced*, is depicted in Figure 7. In this model, the driver is forced to observe all the adjustment edges; hence, they have the potential to adjust the route at every such adjustment edge. Consider the same instance where the driver enters the highway from source s and observes an edge (u_1, v_1) for traffic. In case of high traffic, driver adjusts the route but returns to the next source node u_2 to observe adjustment edge (u_2, v_2). In case of low traffic, driver traverses the edge (u_1, v_1), and continues on the highway until u_2, where they observe edge (u_2, v_2) for traffic. In case of high traffic at (u_2, v_2), driver adjusts the route to destination t. In case of low traffic, driver traverses the edge (u_2, v_2), and continues on the highway to reach the destination t. In this model, the driver always observes all the adjustment edges, irrespective of the traffic states of previous adjustment edges.

Let $\mathbb{E}_{\text{sf}}[(u_1, v_1),(u_2, v_2)]$ denote the expected travel time if adjustment edges (u_1, v_1) and (u_2, v_2) are selected. One could find a pair of adjustment edges that yield a minimum expected travel time, which is given by $\arg\min_{(u_1,v_1)(u_2,v_2)} \mathbb{E}_{\text{sf}}[(u_1, v_1),(u_2, v_2)]$ through complete enumeration using

$$\begin{aligned}\mathbb{E}_{\text{sf}}[(u_1, v_1),(u_2, v_2)] =&\\ &\Big\{\mathbb{E}[s \to u_1] + p_{u_1v_1}[c_{u_1v_1} + \mathbb{E}[v_1 \to u_2]] + (1 - p_{u_1v_1})\mathbb{E}[u_1 \to u_2|d_{u_1v_1}]\Big\}\\ &\qquad + \Big\{p_{u_2v_2}[c_{u_2v_2} + \mathbb{E}[v_2 \to t]] + (1 - p_{u_2v_2})\mathbb{E}[u_2 \to t|d_{u_2v_2}]\Big\}. \end{aligned} \tag{5}$$

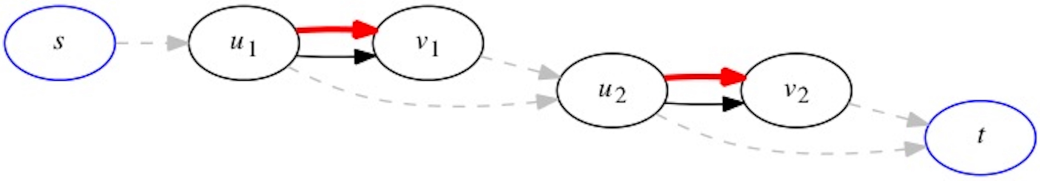


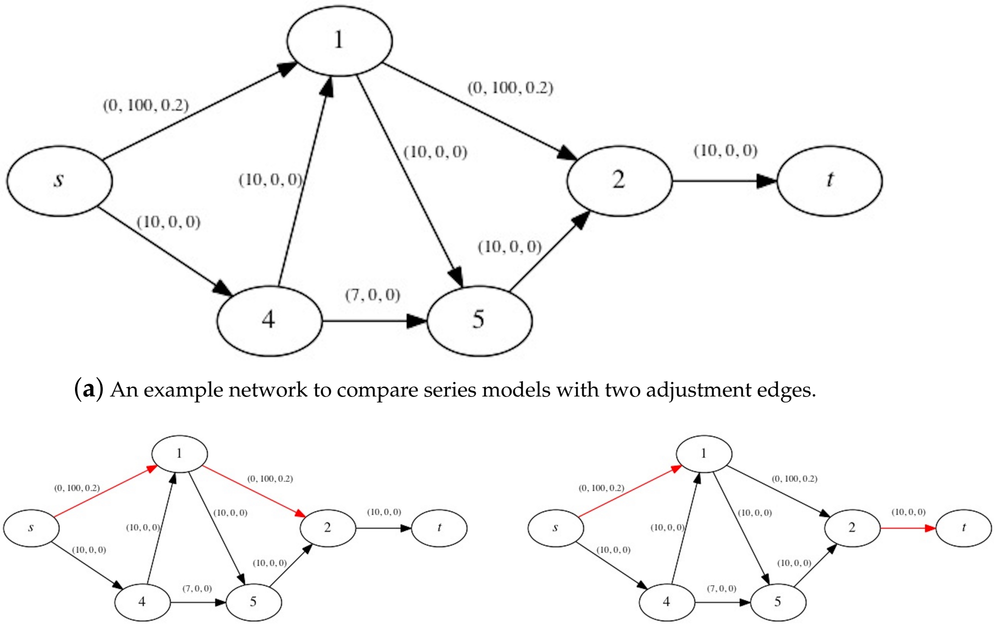
Figure 7. Series forced model with two adjustment edges.

The first summand in (5) is the expected travel time from s to u_2. This first summand includes within it a version of (1), computing the travel time from s to u_2 as dependent on the observation of edge (u_1, v_1). The second summand is the expected travel time from u_2 to t with traffic state observed at (u_2, v_2). Thus, (5) can be expressed as a recursive equation for k adjustment edges as follows:

$$\begin{aligned} &Z^{\text{sf}}_1[s \to t] = Z_1[s \to t], \text{and}\\ &Z^{\text{sf}}_k[s \to t] = \min\Big\{Z^{\text{sf}}_{k-1}[s \to t];\\ &\qquad \min_{(u,v)\in A}\big[Z^{\text{sf}}_{k-1}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}]\big]\Big\}, \end{aligned} \tag{6}$$

where $Z^{\text{sf}}_k[s \to t]$ denotes the overall minimum expected travel time obtained using the series forced model when k adjustment edges are observed for traffic. Though inefficient, an integer programming formulation for this model is presented in Appendix A.2.

An interesting fact is that there is no evidence to show the superiority of one series model over the other, in terms of reducing expected travel time. To illustrate this, let us consider the example network in Figure 8a. The edge weights (c, d, p) represent the travel time under low traffic, the travel time under high traffic, and the probability of low traffic, respectively. The expected travel time of the series models are computed using (4) and (6), and the resulting best adjustment edges are highlighted in in red in Figure 8b and Figure 8c, respectively. We obtain $Z^{\text{suf}}_2[s \to t]$ as 34.8 and $Z^{\text{sf}}_2[s \to t]$ as 35.6, with the series unforced model performing better than the series forced model. Let us now consider another example network as in Figure 9a. We follow the same routine to obtain $Z^{\text{suf}}_2[s \to t]$ as 55.4 and $Z^{\text{sf}}_2[s \to t]$ as 50.8. In this network, the series forced model performs better than the series unforced model. This shows that the performances of the series models are incomparable, and they depend on the network instance considered. Generally, one may think that the series forced model should perform better, because it has the ability to execute several observations in sequence as opposed to just one. However, as these examples demonstrate, it may be too expensive to execute the secondary observations, when unnecessary, as compared to the series unforced model.

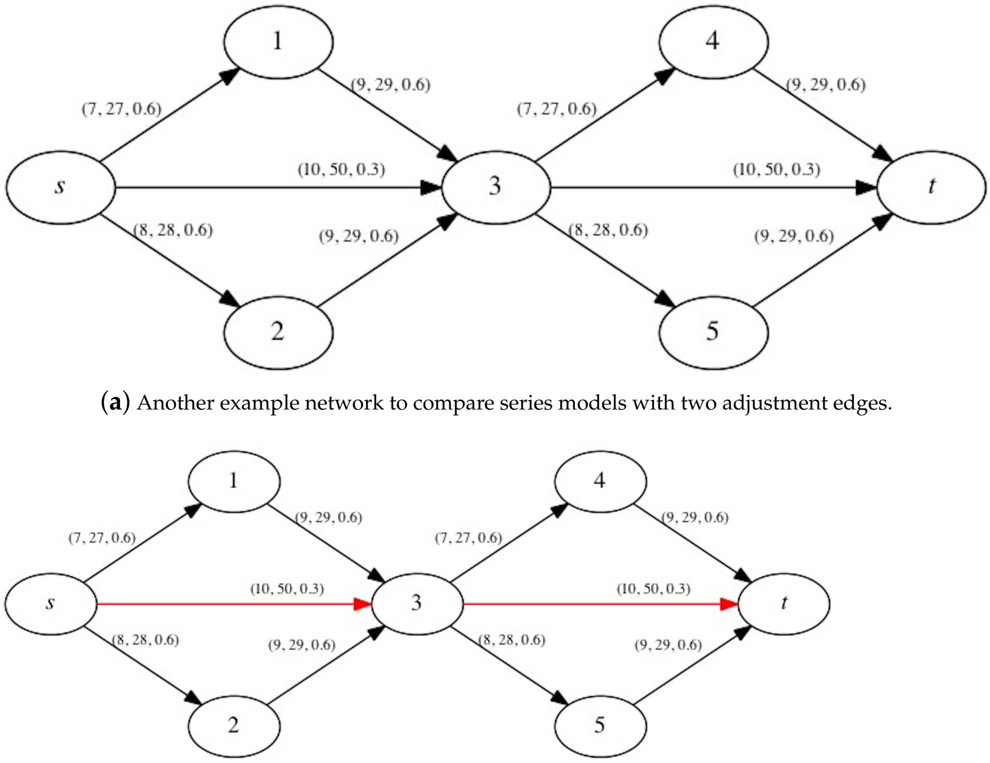


(**a**) An example network to compare series models with two adjustment edges.

(**b**) Solution: Series Unforced Model.

(**c**) Solution: Series Forced Model.

Figure 8. Example 1—Expected travel time comparison to show series unforced model is better. Red solid lines represent the two best adjustment edges for the respective models. Edge weights $e = (c, d, p)$ depict $c_{[e]}$ as travel time under low traffic, $d_{[e]}$ as travel time due to high traffic, and $p_{[e]}$ as the probability of low traffic on the edge e.



(**a**) Another example network to compare series models with two adjustment edges.

(**b**) Solution: Series Unforced and Forced Models.

Figure 9. Example 2—Expected travel time comparison to show series forced model is better. Red solid lines represent the two best adjustment edges for the respective models. Edge weights $e = (c, d, p)$ depict $c_{[e]}$ as travel time under low traffic, $d_{[e]}$ as travel time due to high traffic, and $p_{[e]}$ as the probability of low traffic on the edge e.

3.2.3. Parallel Model

Another model with two adjustment edges, which we call *parallel*, is depicted in Figure 10. In this model, the driver has the potential to observe edges and make route adjustments, in both original and adjustment routes. Consider the same instance where the driver enters the highway from source s and observes an edge (u_{11}, v_{11}) for traffic. In case of low traffic, driver traverses the edge (u_{11}, v_{11}), and continues on the highway until u_{12}, where they observe edge (u_{12}, v_{12}) for traffic. In case of high traffic at (u_{11}, v_{11}), driver adjusts the route to reach node u_{22} and observes an edge (u_{22}, v_{22}) for traffic in the adjusted route. In case of high traffic at the second adjustment edge ((u_{12}, v_{12}) or (u_{22}, v_{22})), driver adjusts the route to destination t. In case of low traffic, driver traverses the edge, and continues on the route to reach the destination t. Unlike series models, here driver observes different adjustment edges based on the traffic state of previous adjustment edges.

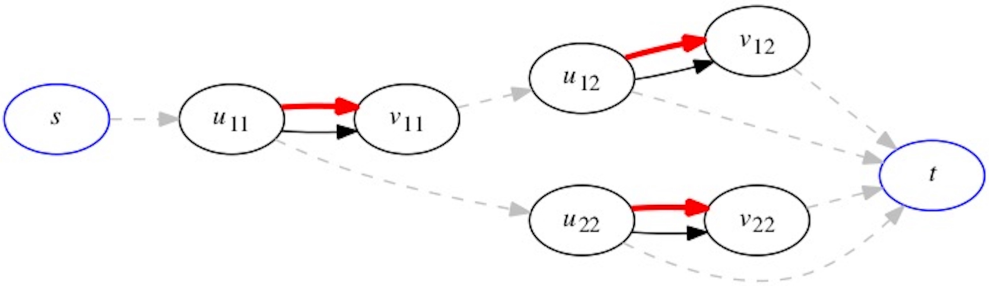


Figure 10. Parallel model with two adjustment edges.

Let $\mathbb{E}_{\text{pll}}[(u_{11}, v_{11}), (u_{12}, v_{12}), (u_{22}, v_{22})]$ denote the expected travel time if adjustment edges (u_{11}, v_{11}), (u_{12}, v_{12}), and (u_{22}, v_{22}) are selected. One could find a set of adjustment edges that yield a minimum expected travel time, $\arg\min_{(u_{11},v_{11}),(u_{12},v_{12}),(u_{22},v_{22})} \mathbb{E}_{\text{pll}}[(u_{11}, v_{11}),$ (u_{12}, v_{12}), $(u_{22}, v_{22})]$, through complete enumeration using

$$\begin{aligned}
\mathbb{E}_{\text{pll}}[(u_{11}, v_{11}), (u_{12}, v_{12}), (u_{22}, v_{22})] &= \mathbb{E}[s \to u_{11}] \\
&\quad + p_{u_{11}v_{11}}[c_{u_{11}v_{11}} + \Big\{ \mathbb{E}[v_{11} \to u_{12}] + p_{u_{12}v_{12}}[c_{u_{12}v_{12}} + \mathbb{E}[v_{12} \to t]] \\
&\qquad + (1 - p_{u_{12}v_{12}})\mathbb{E}[u_{12} \to t | d_{u_{12}v_{12}}] \Big\}] \\
&\quad + (1 - p_{u_{11}v_{11}}) \Big\{ \mathbb{E}[u_{11} \to u_{22} | d_{u_{11}v_{11}}] + p_{u_{22}v_{22}}[c_{u_{22}v_{22}} + \mathbb{E}[v_{22} \to t]] \\
&\qquad + (1 - p_{u_{22}v_{22}})\mathbb{E}[u_{22} \to t | d_{u_{22}v_{22}}] \Big\}
\end{aligned} \tag{7}$$

The first summand is the expected travel time from s to u_{11}. The second and third summands together represent the weighted sum of expected travel times from u_{11} to t, with weights representing the traffic state at (u_{11}, v_{11}). The second summand includes within it a version of (1), computing the travel time from v_{11} to t as dependent on the observation of edge (u_{12}, v_{12}). The third summand includes within it a modified version of (1), the difference being the first term, where we compute the expected travel time from u_{11} to u_{22} given that high traffic is observed at (u_{11}, v_{11}).

Let $Z_k^{\text{pll}}[s \to t]$ denote the overall minimum expected travel time from s to t with k adjustment edges. It is to be noted that the driver's policy may include more than k adjustment edges, but only k edges will be observed in total as they travel from s to t. We now express (7) as recursive equations as follows:

$$
\begin{aligned}
Z_1^{\text{pll}}[s \to t] &= Z_1[s \to t],\\
Z_k^{\text{pll}}[s \to t] &= \min\Big\{ Z_{k-1}^{\text{pll}}[s \to t];\\
&\quad \min_{(u,v)\in A}\big[\mathbb{E}[s \to u] + p_{uv}(c_{uv} + Z_{k-1}^{\text{pll}}[v \to t])\\
&\qquad + (1-p_{uv})Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}]\big]\Big\}, \text{and}\\
Z_1^{\text{pll}}[g \to i|D] &= \min\Big\{\\
&\quad \min_{(g,v)\in A-D}\big[p_{gv}(c_{gv} + \mathbb{E}[v \to i])\\
&\qquad + (1-p_{gv})E[g \to i|D \cup \{d_{gv}\}]\big],\\
&\quad \min_{(u\neq g,v)\in A}\big[\mathbb{E}[g \to u|D] + p_{uv}(c_{uv} + E[v \to i])\\
&\qquad + (1-p_{uv})\mathbb{E}[u \to i|\{d_{uv}\}]\big]\Big\},\\
Z_k^{\text{pll}}[g \to i|D] &= \min\Big\{\\
&\quad \min_{(g,v)\in A-D}\big[p_{gv}(c_{gv} + Z_{k-1}^{\text{pll}}[v \to i])\\
&\qquad + (1-p_{gv})Z_{k-1}^{\text{pll}}[g \to i|D \cup \{d_{gv}\}]\big],\\
&\quad \min_{(u\neq g,v)\in A}\big[\mathbb{E}[g \to u|D] + p_{uv}(c_{uv} + Z_{k-1}^{\text{pll}}[v \to i])\\
&\qquad + (1-p_{uv})Z_{k-1}^{\text{pll}}[u \to i|\{d_{uv}\}]\big]\Big\},
\end{aligned}
\tag{8}
$$

where $Z_k^{\text{pll}}[g \to i|D]$ denotes the minimum expected travel time from any g to i, given that high traffic is observed at all edges in set $D = \{d_{g,j_1}, d_{g,j_2}, \ldots, d_{g,j_{k-1}}\}$.

It is easy to see that the series models are special cases of parallel model, i.e., a solution to a series model can be expressed as a solution to the corresponding parallel model. Hence, the parallel model always outperforms the series models in terms of reducing travel time, but at the expense of more computational effort. It also follows that a parallel model reduces to a Canadian Traveller Problem (CTP) on directed acyclic graphs (DAGs) when all edges in the network are observed for traffic, i.e., k equals $|A|$. Thus, the proposed dynamic programming algorithm can be used to solve CTP on DAGs. The dynamic programming algorithm proposed in [17] differs from our algorithm mainly by the following two points:

(1) In [17], an optimal outgoing edge is computed upon arrival at a node as the graph is traversed. This is different from our dynamic programming approach, where we pre-compute both the original and adjustment routes to the destination;

(2) The algorithm [17] iterates over all the edges in the network, whereas our algorithm is made to stop when observing more adjustment edges no longer reduces the expected travel time.

4. Large Scale Tractable Algorithms

We use the Austin road network (Figure 11) to evaluate the performance of our proposed dynamic programming algorithms. The travel times c on the edges are known (Source URL: http://austintexas.gov/department/gis-and-maps/gis-data, accessed on 2 November 2018), and we assume the probability of low traffic and delay offsets based on the street type, as depicted in Figure 11. The Austin road network consists of about 100,000 edges and it is impractical to find the best adjustment edges, even in a single route adjustment policy, through complete enumeration. For example, it takes about 6 h to find a single adjustment edge for the example source–destination pair shown in Figure 11. Inspired by the traditional branch and bound techniques, we develop easily computable lower and upper bounds to first prune the network and then eliminate many possibilities of adjustment edges, in order to reduce the overall problem size and create truly tractable algorithms.

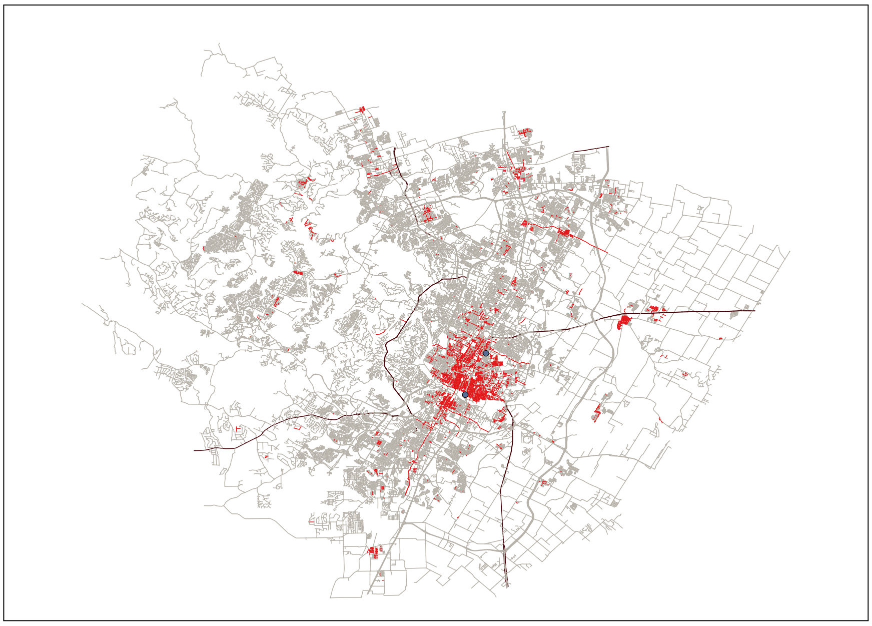


Figure 11. Map of the Austin Road Network. Brown-colored edges assume p = 0.4 and $d = 5 \times c$. Red-colored edges assume p = 0.5 and $d = 4 \times c$. Remaining edges assume p = 0.6 and $d = 3 \times c$. Solid blue dots represent an example source–destination pair.

4.1. Network Pruning

We first focus on developing some easily computable upper and lower bounds to prune the network size. This improves the run time of the shortest path procedures and, consequently, the tractability of the proposed dynamic programming algorithms.

Let $Z_k^{\mathrm{M}}[s \to t]$ represent the minimum expected travel time with k adjustment edges and any route adjustment model M.

Lemma 1. *The minimum expected travel time between two nodes s and t is non-decreasing with k adjustment edges, i.e.,* $\mathbb{E}[s \to t] \geq Z_1[s \to t] \geq Z_2^{\mathrm{M}}[s \to t] \geq \cdots \geq Z_{k-1}^{\mathrm{M}}[s \to t] \geq Z_k^{\mathrm{M}}[s \to t]$.

Proof. The recursive Equation (8) shows that $Z_k^{\mathrm{pll}}[s \to t] \leq Z_{k-1}^{\mathrm{pll}}[s \to t]$, for any $k \geq 2$. Recursively, we can write, $Z_1[s \to t] \geq Z_2^{\mathrm{pll}}[s \to t] \geq \cdots \geq Z_{k-1}^{\mathrm{pll}}[s \to t] \geq Z_k^{\mathrm{pll}}[s \to t]$. To show $\mathbb{E}[s \to t] \geq Z_1[s \to t]$, consider an edge (u, v) on the shortest path from s to t. Then, the expected travel time of the shortest path, $\mathbb{E}[s \to t]$, can be written as

$$\mathbb{E}[s \to t] = \mathbb{E}[s \to u] + p_{uv}c_{uv} + (1 - p_{uv})d_{uv} + \mathbb{E}[v \to t]. \tag{9}$$

A term-by-term comparison of (9) with (1) shows that $\mathbb{E}[s \to t]$ is an upper bound to $\mathbb{E}_1[(u, v)]$ because going through a high traffic edge (u, v) is one potential routing for $\mathbb{E}[u \to t \mid d_{uv}]$. Thus, $\mathbb{E}[s \to t]$ is an upper bound to $Z_1[s \to t]$. Using similar logic, the lemma can be proven for the series models as well. □

Let us assume that there exists an optimal policy π that includes edge (i, j) on one of the paths generated and let $Z_k(\pi)$ be the corresponding expected travel time. A lower bound on the travel time of any path going through edge (i, j) can be given by

$$LBP(i, j) = c_{s \to i} + c_{ij} + c_{j \to t}. \tag{10}$$

where $c_{i \to j}$ represents the shortest path from i to j with edge lengths c, i.e., assuming low traffic on all the edges.

Let $\rho = \min_{(u,v) \in A} \{p_{uv}, 1 - p_{uv}\}$. In other words, the probabilities of low traffic are bounded away from (0, 1) by at least ρ. Under a single route adjustment, every path occurs in policy π with probability of at least ρ. Under k route adjustments, every path occurs with probability of at least ρ^k. This leads to the following lemma defining a lower bound on any policy that uses edge (i, j).

Lemma 2. *Every k-route adjustment policy π that includes edge (i, j) on some path has $Z_k(\pi) \geq \rho^k LBP(i, j) + (1 - \rho^k)c_{s \to t}$.*

Proof. Any path with edge (i, j) occurs with probability at least ρ^k and has length at least $LBP(i, j)$. All other paths in the policy π have length at least $c_{s \to t}$. □

Now, we are ready to present our theorem on network pruning.

Theorem 1. *An edge (i', j') with $\rho^k LBP(i', j') + (1 - \rho^k)c_{s \to t} > \mathbb{E}[s \to t]$, for any $k \geq 1$, will not be on any path in the optimal routing policy.*

Proof. Let π denote the optimal routing policy. By Lemma 2, we have $Z_k(\pi) \geq \rho^k LBP(i', j') + (1 - \rho^k)c_{s \to t}$. If $Z_k(\pi) > \mathbb{E}[s \to t]$, by Lemma 1, π is not an optimal routing policy. Hence, the edge (i', j') will not be on any path in the optimal policy π. □

Using the result of Theorem 1, one can prune the network by eliminating several possibilities. The network pruning procedure is also summarized in Algorithm 1. For the example source-destination pair considered, and for $k = 1$ and $\rho = 0.4$, the network is pruned to 17,328 edges.

Algorithm 1 Network Pruning

Require: The following are input parameters:

1. *Network* which denotes city network with latitutde and longitude of nodes, and edges connecting those nodes;
2. Edge weights $e = (c, d, p)$ include free flow times $c_{[e]}$, low traffic probability $p_{[e]}$ and travel time due to high traffic $d_{[e]}$;
3. Latitude and longitude of source s and destination t, and maximum number of adjustment edges k.

Ensure: *prunedNetwork*

```
function networkPruning($Network, s, t, k$):
Create an empty network, $prunedNetwork$
Compute the expected travel time due to shortest path from $s$ to $t$ as $\mathbb{E}[s \to t]$
for every edge $(i, j) \in Network$ do
    Compute lower bound $LB(i, j) = \rho^k LBP(i, j) + (1 - \rho^k)c_{s \to t}$, using (10) and Lemma 2
    if $LB(i, j) \leq \mathbb{E}[s \to t]$ then
        Append edge $(i, j)$ to $prunedNetwork$
    end if
end for
return $prunedNetwork$
```

4.2. Critical Adjustment Edges

In addition to pruning the network size, it is also possible to obtain a set of critical adjustment edges that contain the optimal solution (as summarized in Algorithm 2). For this, we employed different lower bounds, as discussed in this section.

Lemma 3. *An optimal single route adjustment policy π with adjustment edge (u, v) has $Z_1(\pi) \geq LBA_1(u, v)$, where $LBA_1(u, v)$ is given by*

$$LBA_1(u,v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t].$$

Proof. By definition, we have $d_{uv} > c_{uv}$. Edge (u,v) is an optimal adjustment edge, so using (1) and (2) yields

$$\begin{aligned} Z_1(\pi) &= \min_{(u',v')\in A} \mathbb{E}_1[(u',v')] \\ &= \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1-p_{uv})\mathbb{E}[u \to t|d_{uv}]. \end{aligned}$$

We proceed to complete the proof by contradiction.

Assume $\mathbb{E}[u \to t \mid d_{uv}] < c_{uv} + \mathbb{E}[v \to t]$. Consider a policy π' that uses no adjustment edge and the route $\mathbb{E}[s \to u]$ is followed by $\mathbb{E}[u \to t|d_{uv}]$. Thus, the policy π' has length $\mathbb{E}[s \to u] + \mathbb{E}[u \to t|d_{uv}] < \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1-p_{uv})\mathbb{E}[u \to t|d_{uv}]$, implying π is not optimal. This yields $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + \mathbb{E}[v \to t]$.

Using this result, we have

$$\begin{aligned} Z_1(\pi) &= \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1-p_{uv})\mathbb{E}[u \to t|d_{uv}] \\ &\geq \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1-p_{uv})(c_{uv} + \mathbb{E}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + (c_{uv} + \mathbb{E}[v \to t]). \end{aligned}$$

□

We define the following variables to simplify our notations in the remainder of the section. One can easily pre-compute these quantities and use it in subsequent computations.

$$\begin{aligned} \int[j] &= \max_{(a,b)\in A} (1-p_{ab})\big[d_{ab} + \mathbb{E}[b \to j] - \mathbb{E}[a \to j|d_{ab}]\big], \\ \S[j] &= \max_{(a,b)\in A} (1-p_{ab})\big[\mathbb{E}[a \to j|d_{ab}]\big], \\ \alpha &= \max_{(a,b)\in A} p_{ab} \qquad \text{and} \qquad \beta = \max_{j\in N} \int[j]. \end{aligned} \tag{11}$$

Lemma 4 (Series unforced model)**.** *An optimal route adjustment policy π with edge (u,v) as its first adjustment edge has $Z_k^{\text{suf}}(\pi) \geq LBA_k^{\text{suf}}(u,v)$, where $LBA_k^{\text{suf}}(u,v)$, is given by*

$$LBA_k^{\text{suf}}(u,v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t] \sum_{k'=0}^{k-2} \alpha^{k'}. \tag{12}$$

Proof. Because π is an optimal policy and edge (u,v) is the first adjustment edge, using (4), we have

$$Z_k^{\text{suf}}(\pi) = \mathbb{E}[s \to u] + (1-p_{uv})\mathbb{E}[u \to t|d_{uv}] + p_{uv}(c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]).$$

We show $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]$, by contradiction. Assume $\mathbb{E}[u \to t|d_{uv}] < c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]$. Consider a policy π' that uses no adjustment edges and the route $\mathbb{E}[s \to u]$ is followed by $\mathbb{E}[u \to t|d_{uv}]$. Thus, the policy π' has length $\mathbb{E}[s \to u] + \mathbb{E}[u \to t|d_{uv}] < \mathbb{E}[s \to u] + (1-p_{uv})\mathbb{E}[u \to t|d_{uv}] + p_{uv}(c_{uv} + Z_{k-1}^{\text{suf}}[v \to t])$, implying π is not optimal. Thus, $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]$.

Using this result, we have

$$\begin{aligned} Z_k^{\text{suf}}(\pi) &= \mathbb{E}[s \to u] + (1-p_{uv})\mathbb{E}[u \to t|d_{uv}] + p_{uv}(c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + (1-p_{uv})(c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]) + p_{uv}(c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]. \end{aligned} \tag{13}$$

This is a valid yet intractable lower bound to $Z_k^{\text{suf}}(\pi)$. To alleviate this issue, we derive a lower bound for $Z_{k-1}^{\text{suf}}[v \to t]$. The potential savings in travel time from i to j due to the single-route adjustment policy (using (1) and (4)) are given by

$$\begin{aligned}
\mathbb{E}[i \to j] - Z_1[i \to j] &\le \mathbb{E}[i \to j] - \min_{(u,v)\in A}\big(\mathbb{E}[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\
&\qquad\qquad + (1-p_{uv})\mathbb{E}[u \to j|d_{uv}]\big) \\
&\le \max_{(u,v)\in A}\big(\mathbb{E}[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\
&\qquad\qquad + (1-p_{uv})(d_{uv} + \mathbb{E}[v \to j]) \\
&\qquad - (\mathbb{E}[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\
&\qquad\qquad + (1-p_{uv})\mathbb{E}[u \to j|d_{uv}])\big) \\
&\le \max_{(u,v)\in A}(1-p_{uv})\big[d_{uv} + \mathbb{E}[v \to j] - \mathbb{E}[u \to j|d_{uv}]\big] \\
&= \int[j].
\end{aligned} \tag{14}$$

It is important to note that (14) holds for all route adjustment models since $Z_1^{\text{suf}}[i \to j] = Z_1^{\text{sf}}[i \to j] = Z_1^{\text{pll}}[i \to j] = Z_1[i \to j]$.

The potential savings in travel time from i to j due to the two-route adjustment policy are given by

$$\begin{aligned}
Z_1[i \to j] - Z_2^{\text{suf}}[i \to j] &\le \max_{(u,v)\in A}\big(\mathbb{E}[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\
&\qquad + (1-p_{uv})\mathbb{E}[u \to j|d_{uv}] \\
&\qquad - (\mathbb{E}[i \to u] + p_{uv}(c_{uv} + Z_1[v \to j]) \\
&\qquad\qquad + (1-p_{uv})\mathbb{E}[u \to j|d_{uv}])\big) \\
&\le \max_{(u,v)\in A} p_{uv}(\mathbb{E}[v \to j] - Z_1[v \to j]) \\
&\le \max_{(u,v)\in A} p_{uv}\int[j] = \alpha.\int[j].
\end{aligned}$$

The penultimate inequality is due to (14). Combining this result with (13) for $k = 3$, we obtain

$$\begin{aligned}
Z_3^{\text{suf}}(\pi) &\ge \mathbb{E}[s \to u] + c_{uv} + Z_2^{\text{suf}}[v \to t] \\
&\ge \mathbb{E}[s \to u] + c_{uv} + Z_1[v \to t] - \alpha\int[t] \\
&\ge \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t] - \alpha\int[t].
\end{aligned}$$

Extending this logic to any k yields

$$Z_k^{\text{suf}}(\pi) \ge \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t]) - \int[t]\sum_{k'=0}^{k-2}\alpha^{k'}.$$

□

Lemma 5 (Series forced model)**.** *An optimal route adjustment policy π with edge (u,v) as its last adjustment edge has $Z_k^{\text{sf}}(\pi) \ge LBA_k^{\text{sf}}(u,v)$, where $LBA_k^{\text{sf}}(u,v)$ is given by*

$$LBA_k^{\text{sf}}(u,v) = \mathbb{E}[s \to u] - \int[u] - (k-2)\beta + c_{uv} + \mathbb{E}[v \to t]. \tag{15}$$

Proof. Because π is an optimal policy and edge (u,v) is the last adjustment edge, using (6), we have

$$Z_k^{\text{sf}}(\pi) = Z_{k-1}^{\text{sf}}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}E[v \to t]) + (1-p_{uv})\mathbb{E}[u \to t|d_{uv}].$$

Given edge (u, v) is the optimal adjustment edge, we show $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + \mathbb{E}[v \to t]$ following the same procedure as in the proof of Lemma 4.

Assume $\mathbb{E}[u \to t|d_{uv}] < c_{uv} + \mathbb{E}[v \to t]$. Consider a policy π' that uses only the first $k-1$ adjustment edges of π in the same sequence and does not use the last adjustment edge. In other words, the route $Z^{\text{sf}}_{k-1}[s \to u]$ is followed by $\mathbb{E}[u \to t|d_{uv}]$. Thus, the policy π' has length $Z^{\text{sf}}_{k-1}[s \to u] + \mathbb{E}[u \to t|d_{uv}] < Z^{\text{sf}}_{k-1}[s \to u] + (1 - p_{uv})E[u \to t|d_{uv}] + p_{uv}(c_{uv} + \mathbb{E}[v \to t])$, implying π is not optimal. Thus, $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + \mathbb{E}[v \to t]$, given (u, v) is the last adjustment edge.

Using this result, we have

$$\begin{aligned} Z^{\text{sf}}_k(\pi) &= Z^{\text{sf}}_{k-1}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}] \\ &\geq Z^{\text{sf}}_{k-1}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})(c_{uv} + \mathbb{E}[v \to t]) \\ &\geq Z^{\text{sf}}_{k-1}[s \to u] + c_{uv} + \mathbb{E}[v \to t]. \end{aligned} \tag{16}$$

We now proceed to obtain a tractable lower bound on $Z^{\text{sf}}_{k-1}[s \to u]$. The potential savings in travel time from i to j due to the two-route adjustment policy using (6) are given by

$$\begin{aligned} Z_1[i \to j] - Z^{\text{sf}}_2[i \to j] &\leq \max_{(u,v) \in A} \big(Z_1[i \to j] - (Z_1[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\ &\qquad\qquad + (1 - p_{uv})\mathbb{E}[u \to j|d_{uv}]) \big) \\ &\leq \max_{(u,v) \in A} (\mathbb{E}[i \to u] - Z_1[i \to u]) \\ &\leq \max_{(u,v) \in A} \int[j] = \beta. \end{aligned}$$

The penultimate inequality is due to (14). Combining this result with (16) for $k = 3$ yields

$$\begin{aligned} Z^{\text{sf}}_3(\pi) &\geq Z^{\text{sf}}_2[s \to u] + c_{uv} + \mathbb{E}[v \to t] \\ &\geq Z_1[s \to u] - \beta + c_{uv} + \mathbb{E}[v \to t] \\ &\geq \mathbb{E}[s \to u] - \int[u] - \beta + c_{uv} + \mathbb{E}[v \to t]. \end{aligned}$$

By extending the logic to a generic k, we obtain

$$Z^{\text{sf}}_k(\pi) \geq \mathbb{E}[s \to u] - \int[u] - (k-2)\beta + c_{uv} + \mathbb{E}[v \to t].$$

□

Lemma 6 (Parallel model)**.** *An optimal route adjustment policy π with edge (u, v) as its first adjustment edge has $Z^{\text{pll}}_k(\pi) \geq LBA^{\text{pll}}_k(u, v)$ and $Z^{\text{pll}}_k(\pi|D) \geq LBA^{\text{pll}}_k(u, v)$, where $LBA^{\text{pll}}_k(u, v)$ is given by*

$$LBA^{\text{pll}}_k(u, v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t] \sum_{k'=0}^{k-2} \alpha^{k'} - (k-2)\S[t]. \tag{17}$$

Proof. Because π is an optimal policy and edge (u, v) is the first adjustment edge, using (8), we have

$$Z^{\text{pll}}_k(\pi) = \mathbb{E}[s \to u] + (1 - p_{uv})Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] + p_{uv}(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]).$$

We show $Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] \geq c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]$ by contradiction. Assume $Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] < c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]$. Consider a policy π' that uses the $k-1$ adjustment edges of the first adjusted route of π (in the same sequence) and does not use any other adjustment edges. In other words, route $\mathbb{E}[s \to u]$ is followed by $Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}]$. Thus, the policy π' has length $\mathbb{E}[s \to u] + Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] < \mathbb{E}[s \to u] + (1 - p_{uv})Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] + p_{uv}(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t])$, implying π is not optimal. Thus, $Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] \geq c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]$.

Using this result, we have

$$\begin{aligned} Z^{\text{pll}}_{k}(\pi) &= \mathbb{E}[s \to u] + (1 - p_{uv})Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] + p_{uv}(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + (1 - p_{uv})(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]) + p_{uv}(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]. \end{aligned} \tag{18}$$

We follow the same procedure as in the proofs of Lemmas 4 and 5 to obtain a lower bound on $Z^{\text{pll}}_{k-1}[v \to t]$. The potential savings in travel time from i to j due to the two-route adjustment policy using (8) are given by

$$\begin{aligned} Z_1[i \to j] - Z^{\text{pll}}_2[i \to j] &\leq \max_{(u,v)\in A} \big(Z_1[i \to j] - (\mathbb{E}[i \to u] + p_{uv}(c_{uv} + Z_1[v \to j]) \\ &\qquad\qquad + (1 - p_{uv})Z_1[u \to j|d_{uv}])\big) \\ &\leq \max_{(u,v)\in A} \big(p_{uv}(\mathbb{E}[v \to j] - Z_1[v \to j]) \\ &\qquad + (1 - p_{uv})(\mathbb{E}[u \to j|d_{uv}] - Z_1[u \to j|d_{uv}])\big) \\ &\leq \max_{(u,v)\in A} \big(p_{uv}\int[j]\big) + \max_{(u,v)\in A} \big((1 - p_{uv})\mathbb{E}[u \to j|d_{uv}]\big) \\ &= \alpha\int[j] + \S[j]. \end{aligned}$$

The penultimate inequality is due to the fact that $Z_1[u \to j|d_{uv}] \geq 0$ and due to (14).

Similarly, the potential savings in travel time from i to j due to the three-route adjustment policy are given by

$$\begin{aligned} Z^{\text{pll}}_2[i \to j] - Z^{\text{pll}}_3[i \to j] &\leq \max_{(u,v)\in A} \big(p_{uv}(Z_1[v \to j] - Z^{\text{pll}}_2[v \to j]) \\ &\qquad + (1 - p_{uv})(Z_1[u \to j|d_{uv}] - Z^{\text{pll}}_2[u \to j|d_{uv}])\big) \\ &\leq \max_{(u,v)\in A} \big(p_{uv}(\alpha\int[j] + \S[j]) + (1 - p_{uv})\mathbb{E}[u \to j|d_{uv}]\big) \\ &\leq \alpha^2\int[j] + \S[j]. \end{aligned}$$

The penultimate inequality is due to $Z^{\text{pll}}_2[u \to j|d_{uv}] \geq 0$ and $Z_1[u \to j|d_{uv}] \leq \mathbb{E}[u \to j|d_{uv}]$. Using the above results in (18) for $k = 4$, we obtain

$$\begin{aligned} Z^{\text{pll}}_4(\pi) &\geq \mathbb{E}[s \to u] + c_{uv} + Z^{\text{pll}}_3[v \to t] \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z^{\text{pll}}_2[v \to t] - \alpha^2\int[t] - \S[t] \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z^{\text{pll}}_1[v \to t] - \alpha\int[t] - \S[t] - \alpha^2\int[t] - \S[t] \\ &\geq \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t](1 + \alpha + \alpha^2) - 2\S[t]. \end{aligned}$$

By similar logic, we derive, for any k,

$$Z^{\text{pll}}_k(\pi) \geq \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t]\sum_{k'=0}^{k-2}\alpha^{k'} - (k-2)\S[t].$$

Since $Z^{\text{pll}}_k(\pi|D) \geq Z^{\text{pll}}_k(\pi)$ by definition, $LBA^{\text{pll}}_k(u,v)$ is a valid lower bound to $Z^{\text{pll}}_k(\pi|D)$. □

Now, we present our theorem to obtain a set of feasible adjustment edges.

Theorem 2. *An edge (u', v') with $LBA_1(u', v') > \mathbb{E}[s \to t]$ or $LBA_k^{\mathrm{M}}(u', v') > Z_{k-1}^{\mathrm{M}}[s \to t]$ cannot be the first adjustment edge (for M being series unforced or parallel models) or the last adjustment edge (for M being series forced model) in an optimal routing policy.*

Proof. Let π be a routing policy using series unforced model and edge (u', v') as the first adjustment edge.

We show that π is not optimal if $LBA_k^{\mathrm{suf}}(u', v') > Z_{k-1}^{\mathrm{suf}}[s \to t]$.

If π is optimal, we have $Z_k^{\mathrm{suf}}(\pi) \geq LBA_k^{\mathrm{suf}}(u', v')$, using the result of Lemma 4. Since $LBA_k^{\mathrm{suf}}(u', v') > Z_{k-1}^{\mathrm{suf}}[s \to t]$ (by assumption), we have $Z_k^{\mathrm{suf}}(\pi) > Z_{k-1}^{\mathrm{suf}}[s \to t]$, implying π is not optimal. This completes our proof. Similar logic can be used along with Lemmas 5 and 6 to prove this claim for the series forced and parallel route adjustment models. □

Algorithm 2 Critial Adjustment Edges

```
Require: The following are input parameters:
    1. A road network $Network$ with edge weights $e = (c, d, p)$, depicting free flow times $c_{[e]}$, low traffic probability $p_{[e]}$ and travel time due to high traffic $d_{[e]}$;
    2. Source $s$, destination $t$ and number of adjustment edges $k$;
    3. Routing model type M, where $M \in \{Series\ Unforced, Series\ forced, Parallel\}$; and
    4. Optimal expected travel time $Z_{k-1}^{\mathrm{M}}[s \to t]$ for $k-1$ route adjustment policy M. Here, recursive term $Z_{k-1}^{\mathrm{M}}[s \to t] = recursiveFunction(Network, s, t, k-1, \mathrm{M})$, from Algorithm 3 with respect to routing model M and $k-1$ adjustment edges.
Ensure: Set $criticalEdges$
function $feasibleEdges(Network, s, t, k, \mathrm{M}, Z_{k-1}^{\mathrm{M}}[s \to t])$:
Create an empty set $criticalEdges$
if $k = 1$ (Single route adjustment policy) then
    for every edge $(u, v) \in Network$ do
        Compute lower bound $LBA_1(u, v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t]$, as in Lemma 3
        if $LBA_1(u, v) \leq \mathbb{E}[s \to t]$ then
            Append edge $(u, v)$ to set $criticalEdges$
        end if
    end for
end if
if $k > 1$, Multiple route adjustment policy then
    for every edge $(u, v) \in Network$ do
        Compute lower bound $LBA_k^{\mathrm{M}}(u, v)$:
                (a) If model M is Series Unforced Model, $LBA_k^{\mathrm{suf}}(u, v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t] \sum_{k'=0}^{k-2} \alpha^{k'}$, as in Lemma 4 using (11) and (12)
                (b) If model M is Series Forced Model, $LBA_k^{\mathrm{sf}}(u, v) = \mathbb{E}[s \to u] - \int[u] - (k-2)\beta + c_{uv} + \mathbb{E}[v \to t]$, as in Lemma 5 using (11) and (15)
                (c) If model M is Parallel Model, $LBA_k^{\mathrm{pll}}(u, v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t] \sum_{k'=0}^{k-2} \alpha^{k'} - (k-2)\S[t]$, as in Lemma 6 using (11) and (17)
        if $LBA_k^{\mathrm{M}}(u, v) \leq Z_{k-1}^{\mathrm{M}}[s \to t]$ then
            Append edge $(u, v)$ to set $criticalEdges$
        end if
    end for
end if
return $criticalEdges$
```

These two pre-processing algorithms for pruning the network size and eliminating the possibilities of adjustment edges reduce the computation time from several hours to seconds. Specifically, it takes about 10 s to prune the network from 108,000 edges to 17,328 edges and to find a set of 21 feasible edges for a single route adjustment policy, which is depicted in Figure 12. As a result of these pruning processes, the algorithm is able to compute an optimal single-route adjustment policy in less than 15 s. The solution pertaining to the example considered is presented in Figure 13.

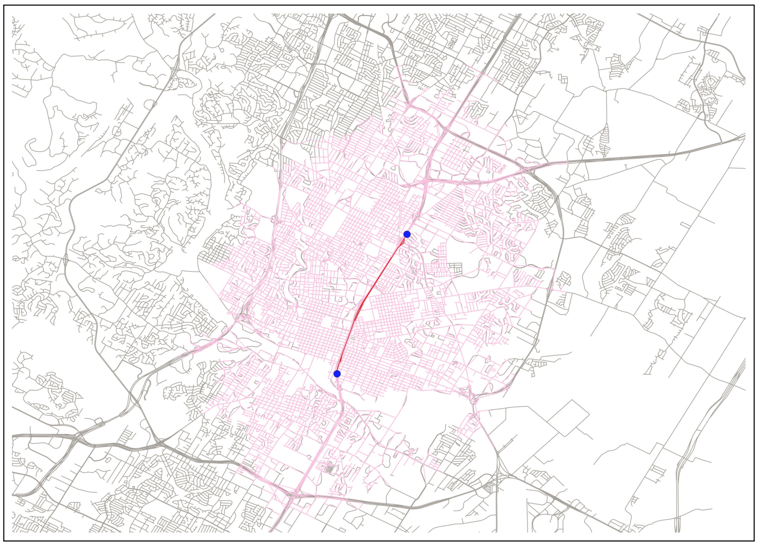


Figure 12. Pruned network and the set of critical adjustment edges for the single-route adjustment policy. Shaded portion in pink represents the pruned network and red solid line represents the set of feasible adjustment edges.

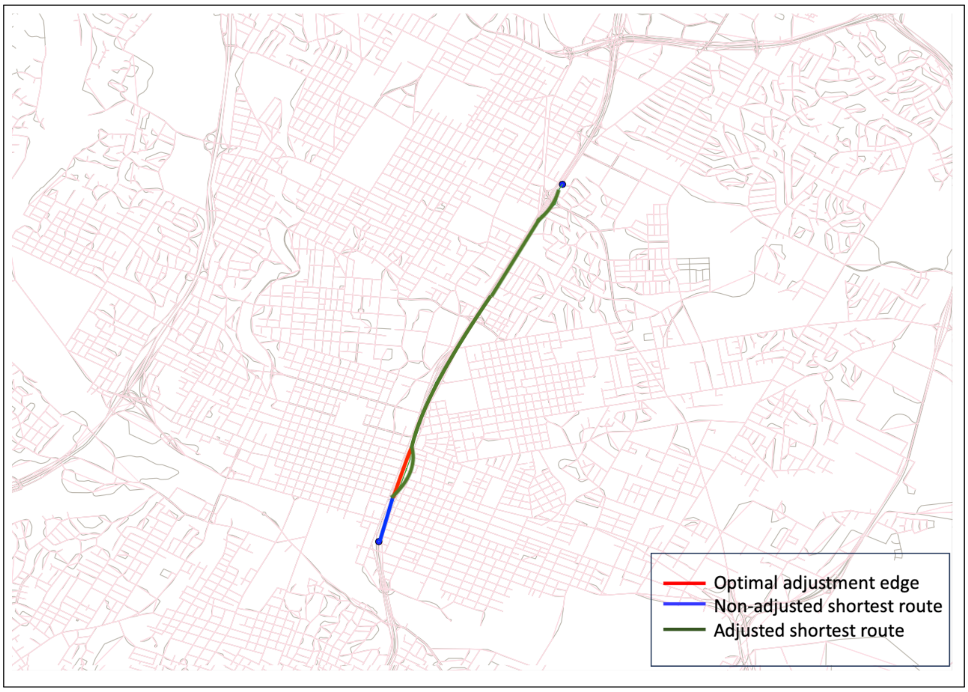
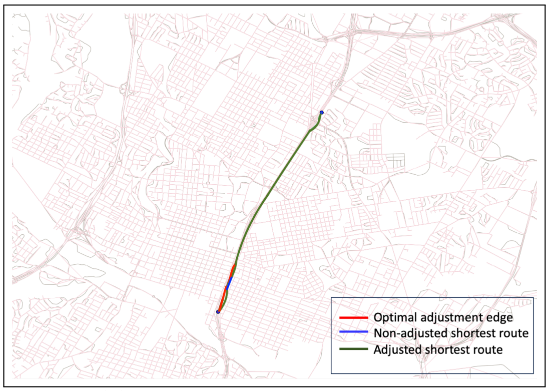


Figure 13. Optimal single route adjustment policy. Red solid line represents the optimal adjustment edge. Blue and green lines represent the non-adjusted and adjusted shortest route, respectively.

For the same source–destination pair, $k = 2$ and $\rho = 0.16$, Theorem 1 prunes the original network to 50,628 edges and Theorem 2 yields a set of 2091 feasible adjustment edges. The routing algorithm(s) is (are) summarized in Algorithm 3 and the solutions are presented in Figure 14. In the final steps of Algorithm 3, we use the popular Dijkstra's shortest path algorithm to compute the shortest paths for original and adjusted routes.

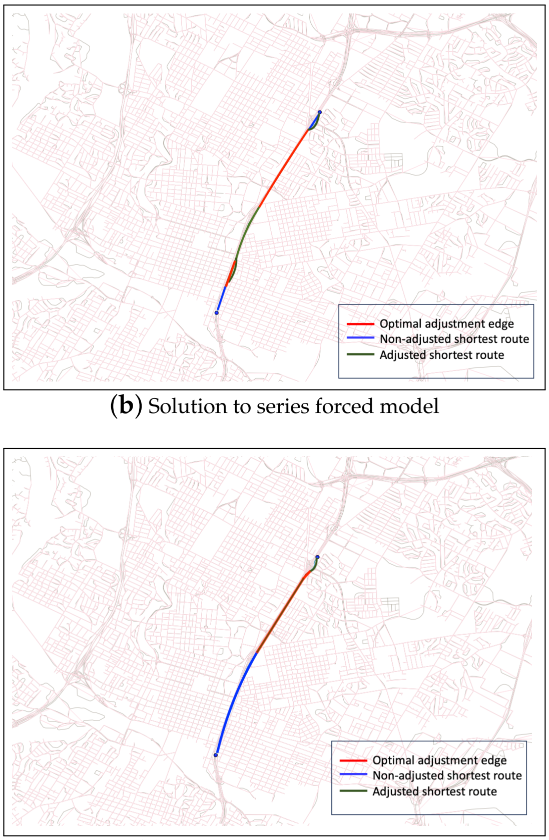
Algorithm 3 Non-Aggressive Adaptive Routing—Recursive Algorithm

```
Require: The following are input parameters:
    1. A road network $Network$ with edge weights $e = (c, d, p)$, depicting free flow times $c_{[e]}$, low traffic probability $p_{[e]}$ and travel time due to high traffic $d_{[e]}$;
    2. Source $s$, destination $t$, and maximum number of adjustment edges $k$; and
    3. Routing model type M, where $M \in \{Series\ Unforced, Series\ forced, Parallel\}$
Ensure: Optimal routing policy $\pi$ that minimizes expected travel time $Z_k^{\mathrm{M}}[s \rightarrow t]$
function $recursiveFunction(Network, s, t, k, \mathrm{M})$:
Create an empty set $optimalAdjEdges$
for every $k' = [1, 2, \cdots, k]$ do
    Compute $prunedNetwork = networkPruning(Network, s, t, k')$,using Algorithm 1
    Compute set $criticalEdges = feasibleEdges(prunedNetwork, s, t, k', \mathrm{M}, Z_{k'-1}^{\mathrm{M}}[s \rightarrow t])$, using Algorithm 2
    Find an optimal edge $(u, v)$ that minimizes the expected travel time $Z_{k'}^{\mathrm{M}}[s \rightarrow t]$:
        (a) If model M is Series Unforced Model, use (4) to compute $Z_{k'}^{\mathrm{suf}}[s \rightarrow t]$
        (b) If model M is Series Forced Model, use (6) to compute $Z_{k'}^{\mathrm{sf}}[s \rightarrow t]$
        (c) If model M is Parallel Model, use (8) to compute $Z_{k'}^{\mathrm{pll}}[s \rightarrow t]$
    if $Z_{k'}^{\mathrm{M}}[s \rightarrow t] \leq Z_{k'-1}^{\mathrm{M}}[s \rightarrow t]$ then
        Append the optimal adjustment edge $(u, v)$ to $optimalAdjEdges$
    else Terminate
    end if
end for
Derive $OriginalRoute$ that combines the shortest paths from source $s$ to start node of every adjustment edge $(u_i, v_i)$ and finally to the destination $t$. Here, edge $(u_i, v_i) = optimalAdjEdges[i]$ is $i^{th}$ adjustment edge with weight $c_{u_i v_i}, \forall i \in [1, 2, \cdots, |optimalAdjEdges|]$. Remaining edges assume expected times as their edge weights.
Derive $AdjustedRoute[i]$ for every $i^{th}$ adjustment edge, that combines the shortest paths from source $s$ to destination $t$ via the adjustment edge following the design presented in Section 3. Here, $i^{th}$ adjustment edge $(u_i, v_i) = optimalAdjEdges[i]$ with weight $d_{u_i v_i}, \forall i \in [1, 2, \cdots, |optimalAdjEdges|]$. Remaining edges assume expected times as their edge weights.
return Optimal policy $\pi = \left[OriginalRoute, optimalAdjEdges, (AdjustedRoute[i], \forall i \in [1, 2, \cdots, |optimalAdjEdges|]\right]$
```



(**a**) Solution to series unforced model

Figure 14. *Cont.*



(**b**) Solution to series forced model

(**c**) Solution to parallel model

Figure 14. Optimal two-route adjustment policy. Red solid line represents the optimal adjustment edges. Blue and green lines represent the non-adjusted and adjusted shortest routes, respectively.

4.3. Performance Evaluation

We conducted several computational experiments by implementing our algorithms in Python and solving it using MacOS 2.6 GHz Intel i5 processor with 8 GB RAM. We summarize the performance of our algorithms with respect to two adjustment edges, using the same example network as in Figure 12, the parallel model performs the best in terms of reducing expected travel time. We save about 7% of travel time when compared to the single-route adjustment policy and about 13% compared to the fixed (non-adaptive) shortest path. Followed by parallel model, the series forced model performs better yielding about 3% savings compared to single-route policy and 9.5% savings compared to fixed path. Finally, the series unforced model provides the least savings, of less than 1% and 7%, respectively.

One can achieve more savings by increasing the number of route adjustments, albeit with a large increase in the computational effort. For the above example, the pruned network size for two adjustment edges is about 50% of the original network size and that of the three edges is almost the same as the original network, increasing the computational effort drastically. Thus, a trade-off arises between the number of edges to be observed for traffic and potential savings in expected travel times. In order to understand this trade-off, we solved our dynamic programming algorithms for different route adjustment models, on a smaller network consisting of 17,328 edges (given by the pruned network of the single-route adjustment model). The graph summarizing the benefit of adaptability is presented in Figure 15.

It can be inferred from the graph that there is not much improvement in the expected travel time beyond two adjustment edges using the series unforced model. However, the series forced model yields about 2% reduction in expected travel time with three adjustment edges, after which the reduction deteriorates and tends to saturate. Similarly, the parallel model results in 3–5% reduction in the travel time up to seven adjustment edges, after which the reduction saturates. Thus, we can conclude that observing more than seven edges in this network instance, as opposed to CTP where all edges are decision points, does not contribute significantly to the reduction in travel time. We emphasize the fact that this summary is specific to the problem instance considered and the performance graph is likely to vary for different instances. Thus, choosing the right adjustment strategy and the right number of adjustments is a decision to be made by the user, based on the trade-off between the computational effort required and the anticipated reduction in expected travel times.

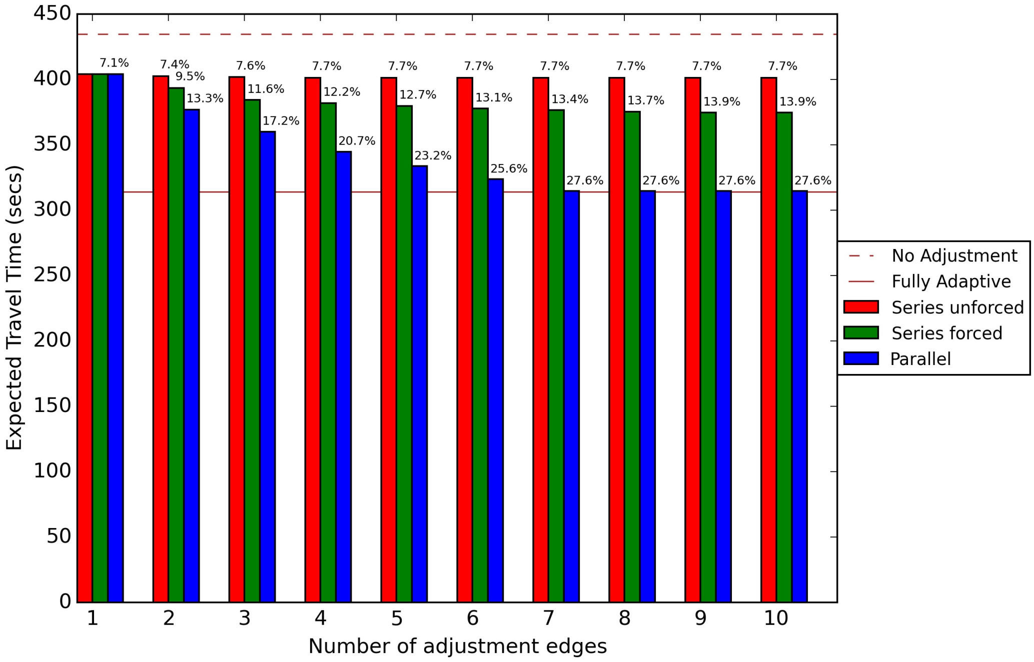


Figure 15. Benefit of adaptability. This graph summarizes the expected travel time across varying number of adjustment edges. Brown dashed and brown solid lines represent the non-adaptive and completely adaptive expected travel times. Red, green, and blue bars represent the summaries of our proposed series unforced, series forced, and parallel models, respectively.

5. Conclusions

In this paper, we present a new way to route traffic in large networks. Our approach is called non-aggressive adaptive routing, and it allows the driver to make limited adjustments to routes while still minimizing the expected travel times. We propose three different routing strategies: series unforced, series forced, and parallel models. In every strategy, the driver observes the optimal adjustment edges for traffic and makes route adjustments in case of high traffic on the observing edge(s). However, where and how these adjustments are performed is the key differentiating factor to choose the right strategy.

We develop exact mathematical models to generate an optimal routing policy from each proposed strategy. In spite of our proposed dynamic programming algorithms being polynomially solvable, the algorithms may get intractable for medium to large sized networks. To alleviate this problem, we design some efficient bounds and present several theorems, to reduce the network size and compute a reduced feasible set of adjustment edges, leading to tractable algorithms.

Finally, we numerically assess the performance of our tractable algorithms using single- and two-route adjustment policies, and present the benefit of adaptability graph for an example source-destination pair in the Austin road network.

In terms of future extensions of this work, there is still scope to improve the tractability of our algorithms by developing even tighter lower bounds. There is also scope to studying other multiple route adjustment strategies; for example, one can consider a model where the driver is constrained to switching back and forth between two or more pre-computed routes. One can extend this model to account for correlation in traffic states between different edges. Another area of exploration will be using real-time traffic data from suitable sources, like unmanned aerial vehicles. Detecting tassels with imagery can provide instantaneous traffic information, in fact, factoring in the traffic information for different times of day might yield better results. Another possible future extension is to consider microscopic traffic factors like an individual vehicle's position and velocity to further optimize the routing policies. We strongly believe that these extensions require significant additional research and therefore we defer their investigation to future work.

Author Contributions: Conceptualization, M.N.P., N.D. and E.N.; Methodology, M.N.P., N.D. and E.N.; Validation, M.N.P.; Formal analysis, M.N.P. and N.D.; Resources, M.N.P.; Data curation, M.N.P.; Writing—original draft, M.N.P.; Writing—review & editing, M.N.P.; Supervision, N.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data used in this paper were accessed from open data sources. Web URLs to the data sources are provided in the reference section.

Acknowledgments: The authors are thankful to the Editor and two anonymous referees for their constructive comments and suggestions that led to substantial improvements of the paper.

Conflicts of Interest: On behalf of all authors, the corresponding author states that there is no conflict of interest.

Appendix A. Integer Programming Formulations

Appendix A.1. IP Formulation–Single Route Adjustment Policy

An IP formulation for a single route adjustment policy is given by

$$\begin{aligned}\underset{(u,v)\in A}{\text{Min}} \sum_{(u,v)\in A} & (E[s\to u] + p_{uv}(c_{uv} + E[v\to t]) + (1-p_{uv})E[u\to t|d_{uv}]) * Z_{uv} \\ \text{s.t.} \sum_{(u,v)\in A} & Z_{uv} = 1\end{aligned}$$

where Z_{uv} is a binary variable that is equal to 1 if edge (u,v) is an adjustment edge and 0 otherwise.

Appendix A.2. IP Formulation–Series Forced Adjustment Policy

An IP formulation to a series forced route adjustment policy with k-route adjustments is given by

$$\begin{aligned}\underset{(u,v)\in A}{\text{Min}} \sum_{(u,v)\in A} & E[s\to u]Z^1_{uv} + \sum_{l=1}^{k}\sum_{(u,v)\in A}\sum_{m\in N} p_{uv}(c_{uv} + E[v\to m]\sum_{n\in N} Z^{l+1}_{mn})Z^l_{uv} \\ & + \sum_{l=1}^{k}\sum_{(u,v)\in A}\sum_{m\in N}(1-p_{uv})E[u\to m|d_{uv}]\sum_{n\in N} Z^{l+1}_{mn}Z^l_{uv} \\ \text{s.t.} \sum_{(u,v)\in A} & Z^l_{uv} = 1 \qquad \forall l = 1,\ldots,k\end{aligned}$$

where Z^l_{uv} is a binary variable that is 1 if edge (u,v) is an adjustment edge at the lth route adjustment and 0 otherwise.

The objective function is quadratic and can be converted to a linear form using any standard conversion technique. IP formulations for other models can also be devised in the same manner.

References

1. Number of Motor Vehicles Registered in the United States from 1990 to 2021. 2022. Available online: https://www.statista.com/statistics/183505/number-of-vehicles-in-the-united-states-since-1990/ (accessed on 27 February 2023).
2. Vehicle Ownership in U.S. Cities Data and Map. 2014. Available online: https://www.governing.com/archive/car-ownership-numbers-of-vehicles-by-city-map.html (accessed on 9 December 2014).
3. City of Austin Population History. 2016. Available online: https://www.austintexas.gov/sites/default/files/files/Planning/Demographics/population_history_pub.pdf (accessed on 29 July 2016).
4. Anderson, W. Population Explosion: Austin Growth Passes Major Milestone. 2015. Available online: https://www.bizjournals.com/austin/news/2015/12/11/population-explosion-austin-growth-passes-major.html (accessed on 11 December 2015).
5. Keeping Austin Weird Shouldn't Include Sitting in Traffic for 41 Hours. 2014. Available online: http://inrix.com/blog/2014/03/keeping-austin-weird-shouldnt-include-sitting-in-traffic-for-41-hours/ (accessed on 1 August 2016).
6. New Report Finds Austin-Area Commuters Wasted More Time in Traffic Than Other Similar Sized Cities in 2020. 2021. Available online: https://communityimpact.com/austin/central-austin/transportation/2021/06/29/new-report-finds-austin-area-commuters-wasted-more-time-in-traffic-than-other-similar-sized-cities-in-2020/ (accessed on 29 June 2021).
7. Croucher, J.S. A note on the stochastic shortest-route problem. *Nav. Res. Logist. Q.* **1978**, *25*, 729–732. [CrossRef]
8. Andreatta, G.; Romeo, L. Stochastic shortest paths with recourse. *Networks* **1988**, *18*, 193–204. [CrossRef]
9. Papadimitriou, C.H.; Yannakakis, M. Shortest paths without a map. *Theor. Comput. Sci.* **1991**, *84*, 127–150. [CrossRef]
10. Blei, D.M.; Kaelbling, L.P. Shortest paths in a dynamic uncertain domain. In Proceedings of the IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments, Stockholm, Sweden, 31 July–6 August 1999; Volume 4, p. 2.
11. Bar-Noy, A.; Schieber, B. The Canadian Traveller Problem. In Proceedings of the SODA, San Francisco, CA, USA, 28–30 January 1991; Volume 91, pp. 261–270.
12. Polychronopoulos, G.H.; Tsitsiklis, J.N. Stochastic shortest path problems with recourse. *Networks* **1996**, *27*, 133–143. [CrossRef]
13. Gao, S. Optimal Adaptive Routing and Traffic Assignment in Stochastic Time-Dependent Networks. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2005.
14. Miller-Hooks, E. Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Netw. Int. J.* **2001**, *37*, 35–52. [CrossRef]
15. Provan, J.S. A polynomial-time algorithm to find shortest paths with recourse. *Netw. Int. J.* **2003**, *41*, 115–125. [CrossRef]
16. Pretolani, D. A directed hypergraph model for random time dependent shortest paths. *Eur. J. Oper. Res.* **2000**, *123*, 315–324. [CrossRef]
17. Nikolova, E.; Karger, D.R. Route Planning under Uncertainty: The Canadian Traveller Problem. In Proceedings of the AAAI, Chicago, IL, USA, 13–17 July 2008; pp. 969–974.
18. Boyan, J.; Mitzenmacher, M. Improved results for route planning in stochastic transportation. In Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, Washington, DC, USA, 7–9 January 2001; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2001; pp. 895–902.
19. Botea, A.; Nikolova, E.; Berlingerio, M. Multi-Modal Journey Planning in the Presence of Uncertainty. In Proceedings of the ICAPS, Rome, Italy, 10–14 June 2013.
20. Nain, P.; Towsley, D.; Bar-Noy, A.; Basu, P.; Johnson, M.P.; Yu, F. Estimating end-to-end delays under changing conditions. In Proceedings of the 8th ACM MobiCom workshop on Challenged networks, Miami, FL, USA, 30 September 2013; ACM: New York, NY, USA; pp. 1–6.
21. Boyles, S.D.; Waller, S.T. Optimal information location for adaptive routing. *Netw. Spat. Econ.* **2011**, *11*, 233–254. [CrossRef]
22. Boyer, S.; Blandin, S.; Wynter, L. Stability of transportation networks under adaptive routing policies. *Transp. Res. Part B Methodol.* **2015**, *81*, 886–903. [CrossRef]
23. Gao, S.; Chabini, I. Optimal routing policy problems in stochastic time-dependent networks. *Transp. Res. Part B Methodol.* **2006**, *40*, 93–122. [CrossRef]
24. Nie, Y.; Fan, Y. Arriving-on-time problem: Discrete algorithm that ensures convergence. *Transp. Res. Rec.* **2006**, *1964*, 193–200. [CrossRef]
25. Kalai, A.; Vempala, S. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.* **2005**, *71*, 291–307. [CrossRef]
26. Xu, Y.; Hu, M.; Su, B.; Zhu, B.; Zhu, Z. The Canadian traveller problem and its competitive analysis. *J. Comb. Optim.* **2009**, *18*, 195–205. [CrossRef]
27. Liu, W.; Hua, M.; Deng, Z.; Huang, Y.; Hu, C.; Song, S.; Gao, L.; Liu, C.; Xiong, L.; Xia, X. A systematic survey of control techniques and applications: From autonomous vehicles to connected and automated vehicles. *arXiv* **2023**, arXiv:2303.05665.
28. Xia, X.; Meng, Z.; Han, X.; Li, H.; Tsukiji, T.; Xu, R.; Zheng, Z.; Ma, J. An automated driving systems data acquisition and analytics platform. *Transp. Res. Part C Emerg. Technol.* **2023**, *151*, 104120. [CrossRef]
29. Gould, B.T.; Brown, P.N. Information design for Vehicle-to-Vehicle communication. *Transp. Res. Part C Emerg. Technol.* **2023**, *150*, 104084. [CrossRef]
30. Chen, G.; Zhao, X.; Gao, Z.; Hua, M. Dynamic drifting control for general path tracking of autonomous vehicles. *IEEE Trans. Intell. Veh.* **2023**, *8*, 2527–2537. [CrossRef]

31. Valiente, R.; Toghi, B.; Pedarsani, R.; Fallah, Y.P. Robustness and adaptability of reinforcement learning-based cooperative autonomous driving in mixed-autonomy traffic. *IEEE Open J. Intell. Transp. Syst.* **2022**, *3*, 397–410. [CrossRef]
32. Zeng, Y.; Chen, X.; Cao, X.; Qin, S.; Cavazza, M.; Xiang, Y. Optimal route search with the coverage of users' preferences. In Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.