Non-Aggressive Adaptive Routing in Traffic

Madhushini Narayana Prasad[a,*], Nedialko Dimitrov[a], Evdokia Nikolova[b]

[a]Department of Operations Research and Industrial Engineering, The University of Texas at Austin, 204 E. Dean Keeton Street, C2200 Austin, Texas 78712.

[b] Department of Electrical and Computer Engineering, The University of Texas at Austin, 2501 Speedway, Austin, Texas 78712

Abstract

Routing a person through a traffic network presents a tension between selecting a fixed route that is easy to navigate and selecting an aggressively adaptive route that minimizes the expected travel time. We propose to create non-aggressive adaptive routes in the middle-ground seeking the best of both these extremes. Specifically, these routes still adapt to changing traffic conditions, however we limit the number of adjustments made in the route. This improves the user experience, by providing a continuum of options between saving travel time and minimizing navigation. We design strategies to model single and multiple route adjustments, and investigate enumerative techniques to solve these models. To alleviate the intractability with handling real-life traffic data, we develop efficient algorithms with easily computable lower and upper bounds. We finally present computational experiments highlighting the benefits of limited adaptability in terms of reducing the expected travel time.

Keywords: Adaptive routing; dynamic programming; directed acyclic graphs; shortest paths.

1 Introduction

Some major cities in the US are facing the problem of rapid population growth. Figure 1(a) shows the fastest growing states based on the census data between 2010 and 2017 [27] and the vast majority of US population growth is concentrated in Texas state, with eight out of fifteen fastest growing cities located in Texas [1]. The capital of Texas, the city of Austin tops this list with 23 percent population growth [3] and

*Corresponding author.
E-mail addresses: madhujournal@gmail.com (M.N.Prasad), ned.dimitrov@gmail.com (N. Dimitrov), nikolova@autsin.utexas.edu (E. Nikolova).

 doi:10.20944/preprints202305.0974.v1

the Forbes [26] also lists Austin as one of the fastest growing American city of 2018. The population in Austin has increased from 650K in 2000 to 925K in 2016 [12] as shown in Figure 1(b) and is expected to increase by at least 30 percent by 2030 [2]. This rapid population growth creates unprecedented problems, major among them being traffic congestion [6]. Already Austin is ranked as the fourth most congested city for the year 2013 by INRIX Inc. [17]. According to their report, due to poor traffic conditions, commuters in Austin spent about 41 hours on average in traffic (three hours more than in 2012) and the the overall travel time is increased by 22 percent. Future predicted population growth will worsen the situation. In order to manage the increasing traffic congestion, it is vital to devise efficient routes to avoid traffic in a metro city like Austin.

There are various strategies and tools currently available to develop 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 using Google maps and Waze are shown in Figure 2(a) and Figure 2(b). An (aggressive) adaptive route 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 to create a middle-ground that seeks the best of both extremes, in this work, we develop 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 several strategies to model and compute the non-aggressive adaptive routes, based on where and how the route adjustments are performed.

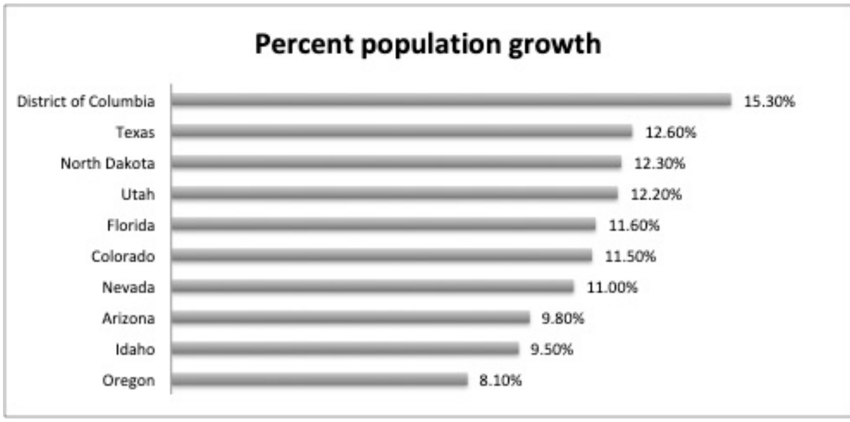
2. We develop exact mathematical methods such as complete enumeration and dynamic programming algorithms for each of the strategies.

3. We derive easily computable bounds to solve the models efficiently for large networks.

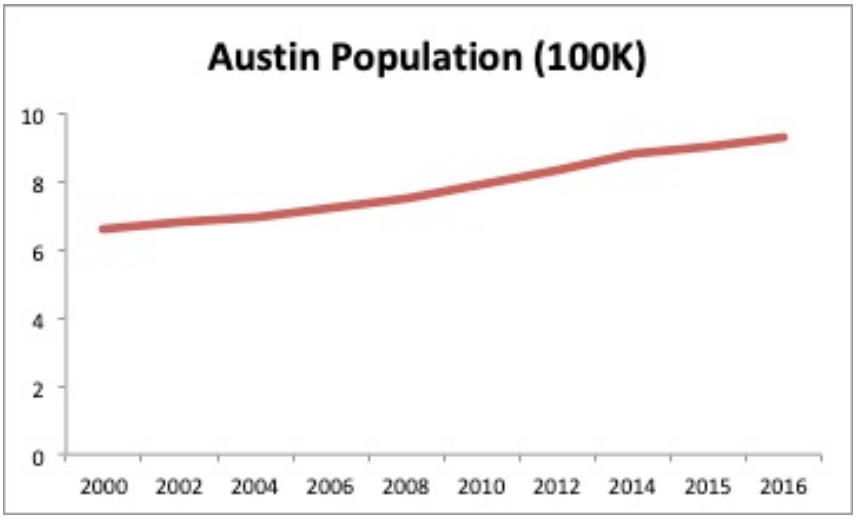
4. We evaluate and analyze the performance of the models using the Austin road network.

The remainder of the paper is organized as follows. Section 2 discusses the related work on adaptive routing. Section 3 describes in detail the proposed modeling strategies and the respective solution methodologies. Section 4 presents a computational evaluation of the proposed models on the Austin road network.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1



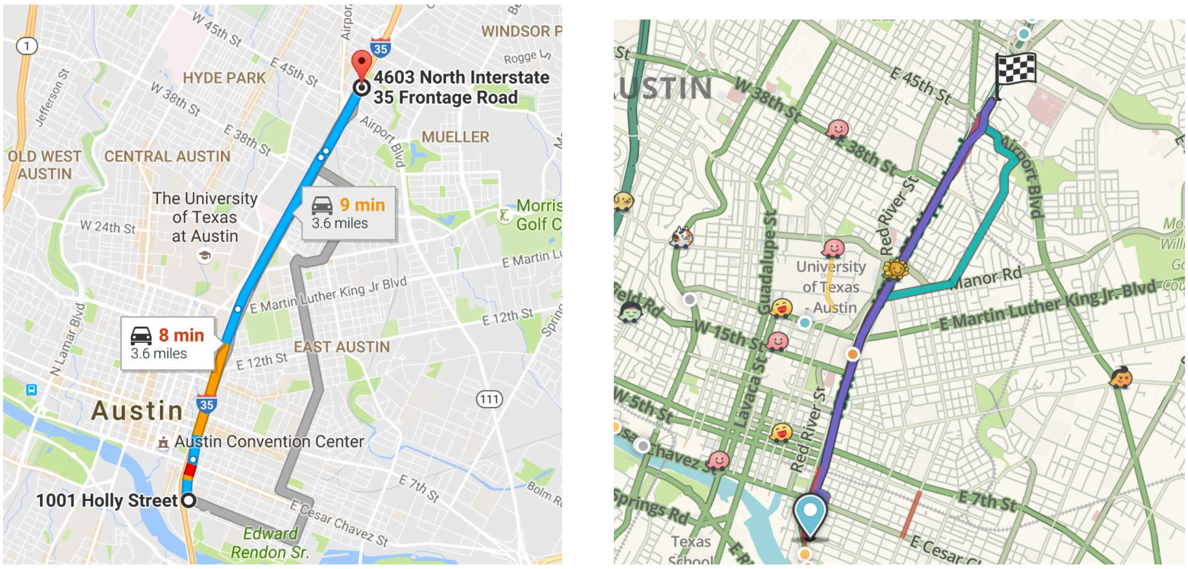
(a) Percent population growth in US



(b) Austin population growth

Figure 1. (a) Top ten US states ranked based on its percent population growth between 2010 to 2017. (b) Population history of Austin city

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1



(a) Google maps: An example

(b) Waze: An example

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

Finally, Section 5 concludes our work by summarizing the key contributions and pointing to future research directions.

2 Related Work

Consider 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 [13] 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 [4] 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 the edge is inactive. In our work, if an edge has traffic congestion, it is still considered active with greater time delay for traversal. However, 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 to all edges for traffic congestion.

Another widely studied variant of adaptive routing is the Canadian Traveller Problem (CTP). CTP was first defined in [22] (see also ([7])). 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 [5]. Polychronopoulos and Tsitsiklis [23] present another variation to CTP where the realization of arc costs is learned progressively as the graph is traversed. They provide dynamic programming algorithms to solve models with both dependent and independent arc costs and they establish 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 [14, 18, 25, 24]. 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 solvable in polynomial time.

Special cases of CTP are studied by Nikolova and Karger [21] to explore exact solutions. They explain the connection of CTP to Markov Decision Processes (MDPs) solvable in polynomial time. They also present polynomially solvable dynamic programming algorithms for 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 total number of edges in the network, in one of our proposed routing strategies, which 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 limited number of adaptations to reduce driving stress. In lieu of this we propose, compare, and contrast several models for defining the decision points, and develop tractable algorithms to compute the optimal routing policy.

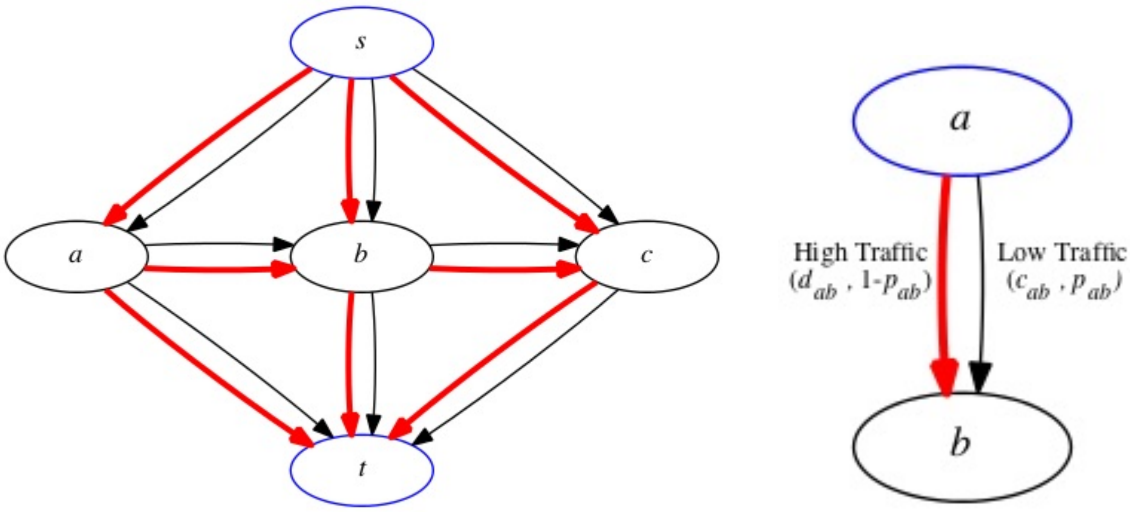
Many other recent extensions to adaptive routing have been proposed, primarily focusing on route planning under uncertainty for different modes of transportation [9, 8, 19], route planning under uncertainty with optimal information location points [11], stability of transportation networks [10], stochastic time dependent networks [15], maximizing the probability of on-time arrival [20], application to online decision problems [16], and competitive analysis of CTP [28].

3 Model Description

Consider a directed acyclic network $G = (N, A)$, with specified source s and destination t nodes as shown in Figure 3(a). 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1



(a) A sample network (b) Traffic Model

Figure 3. Two state traffic model: Red solid line indicates the possibility of high traffic on a 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}.

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 3(b).

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 as *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 $s \to b \to t$ 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 $a \to b \to t$ 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 no-adjustment shortest path.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

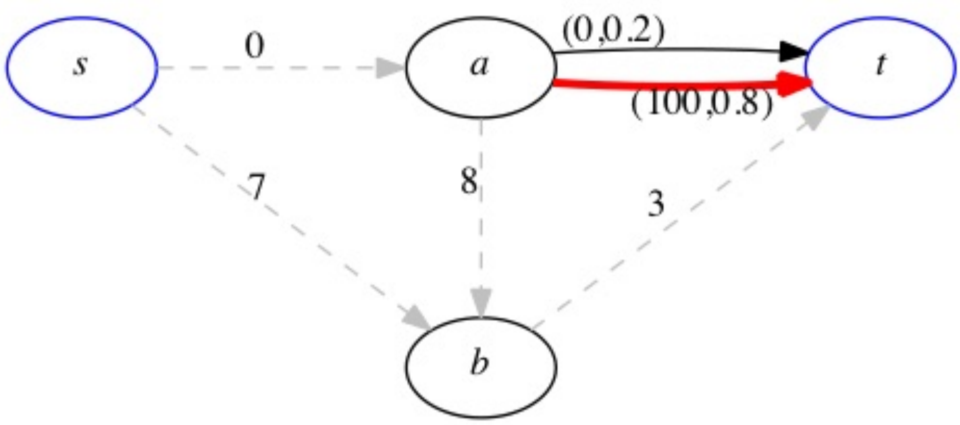


Figure 4. An example to show that an adjustment edge need not be a part of fixed shortest route: Edge weights represent expected travel time. We assume low traffic with probability 1.0 on all the edges except edge (a, t). At edge (a, t) we assume p_{at} as 0.2, c_{at} as 0 and d_{at} as 100 with expected 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, the driver continues on the edge (u, v) and takes the shortest path from v to t. In case of high traffic, the 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

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

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

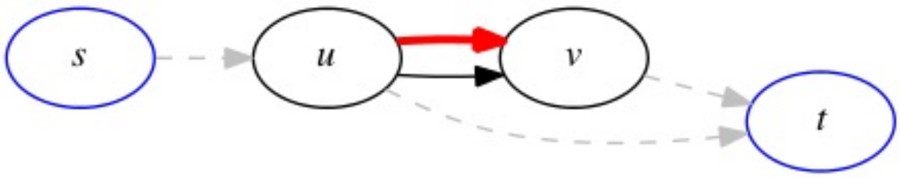


Figure 5. Single Route Adjustment Policy: Solid black line represents an edge and the solid red line represents an adjustment edge. Grey dotted lines represent the shortest paths between the nodes with expected travel time as edge lengths.

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), 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), 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

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

The first summand is the expected travel time from s to u_1. The second summand is the expected travel

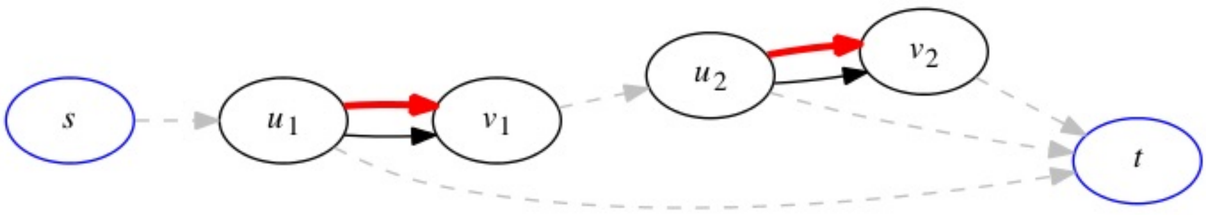


Figure 6. Series Unforced Model with two adjustment edges

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 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

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

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.

The recursive equation yields a dynamic programming algorithm for computing the best set of adjustment edges. The dynamic programming algorithm reduces the computational effort from $O(m^k)$, roughly what is required with complete enumeration, to $O(mk)$ where $m = |A|$. This brings significant computational savings, though as we'll discuss later, 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 has 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

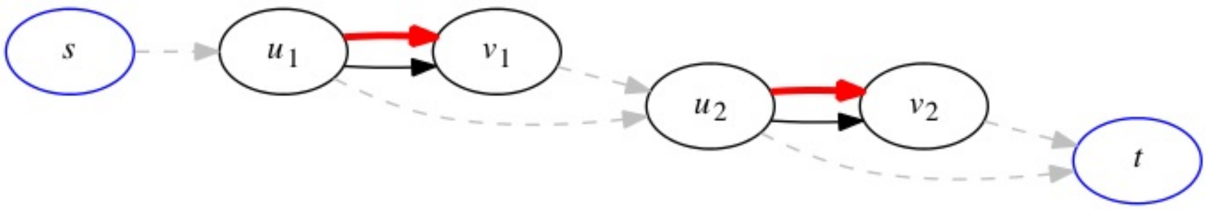


Figure 7. Series Forced Model with two adjustment edges

(u_1, v_1), 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), continues on the highway to reach the destination t. In this model, 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

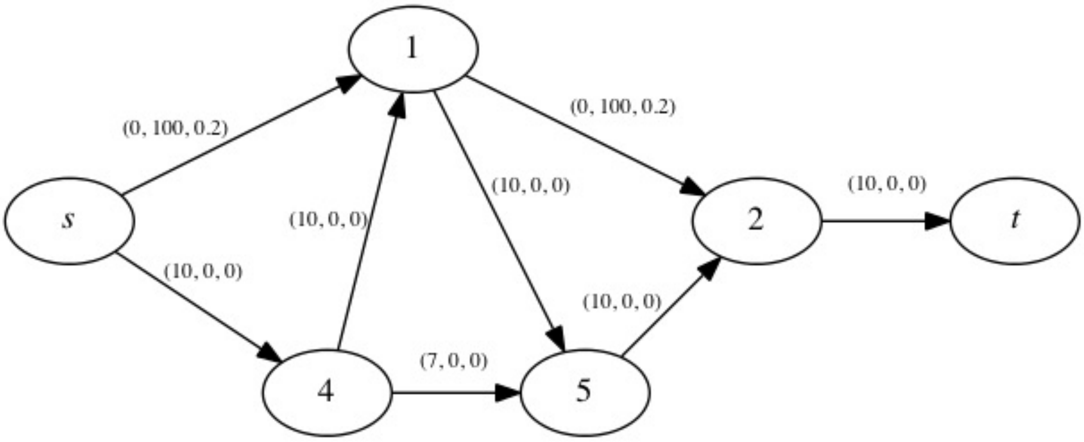
$$\begin{aligned}\mathbb{E}_{\text{sf}}[(u_1, v_1), (u_2, v_2)] = \\ \Big\{\mathbb{E}[s \to u_1] + p_{u_1v_1}[c_{u_1v_1} + \mathbb{E}[v_1 \to u_2]] + (1 - p_{u_1v_1})\mathbb{E}[u_1 \to u_2|d_{u_1v_1}]\Big\} \\ + \Big\{p_{u_2v_2}[c_{u_2v_2} + \mathbb{E}[v_2 \to t]] + (1 - p_{u_2v_2})\mathbb{E}[u_2 \to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

The first summand 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 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 as follows,

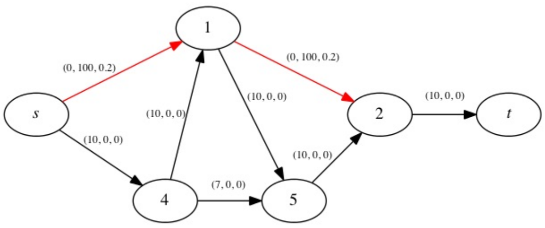
$$\begin{aligned}&Z_1^{\text{sf}}[s \to t] = Z_1[s \to t], \text{and} \\ &Z_k^{\text{sf}}[s \to t] = \min\Big\{Z_{k-1}^{\text{sf}}[s \to t]; \\ &\qquad \min_{(u,v)\in A}\big[Z_{k-1}^{\text{sf}}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}]\big]\Big\},\end{aligned}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

where $Z_k^{\text{sf}}[s \to t]$ 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.

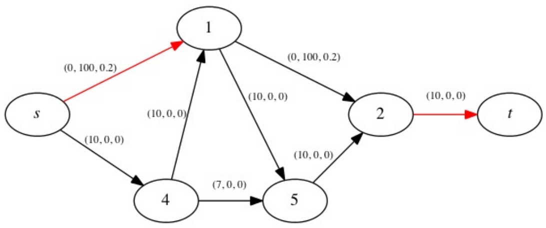
Neither the series unforced nor the forced models are always better in terms of reducing expected travel time. Let us consider the example network in Figure 8(a). The edge weights (c, d, p) represent the travel time under low traffic, the travel time under high traffic and the probability of low traffic respectively. The



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



(b) Solution: Series Unforced Model

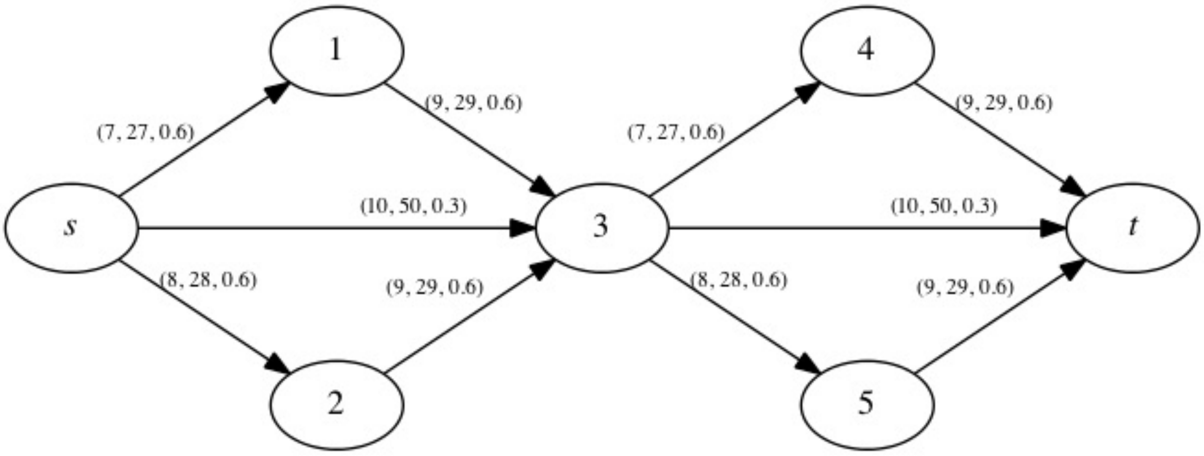


(c) Solution: Series Forced Model

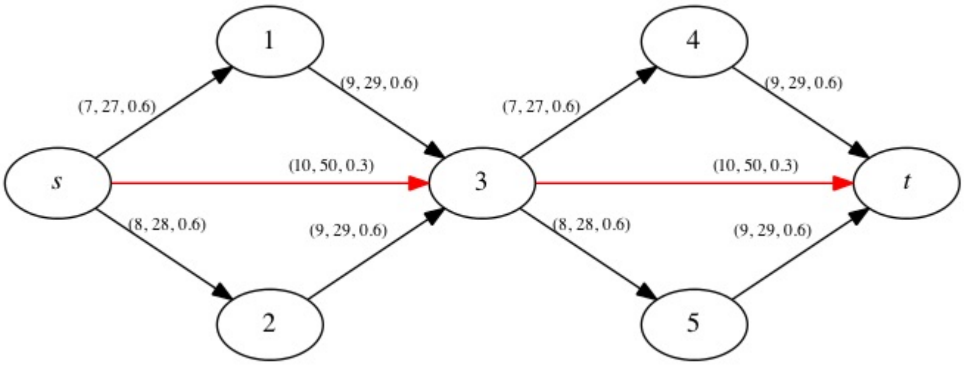
Figure 8. Example 1 – Expected travel time comparison of series models: Red solid lines represent the two best adjustment edges for the respective models.

expected travel time of the series models are computed using (4) and (6), and the resulting best adjustment edges are highlighted in Figure 8(b) and Figure 8(c) respectively. We obtain $Z_2^{\text{suf}}[s \to t]$ as 34.8 and $Z_2^{\text{sf}}[s \to t]$ as 35.6, with series unforced model performing better than the series forced model. Let us now consider another example network as in Figure 9(a). We follow the same routine to obtain $Z_2^{\text{suf}}[s \to t]$ as 55.4 and $Z_2^{\text{sf}}[s \to t]$ as 50.8. In this network, series forced model performs better than the series unforced model. This shows that the performance of the series models are incomparable, and it depends 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, as compared to the series unforced model.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1



(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 of series models: Red solid lines represent the two best adjustment edges for the respective models.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

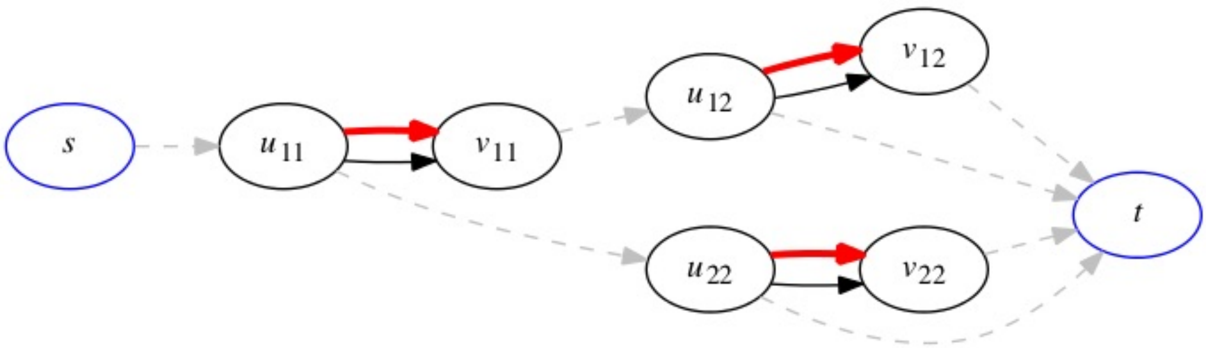


Figure 10. Parallel Model with two adjustment edges

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 the 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}), 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, continues on the route to reach the destination t. Unlike series models, driver observes different adjustment edges based on the traffic state of the previous 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

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

The first summand is the expected travel time from s to u_{11}. The second and third summands together

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

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 dependent on the observation of edge (u_{12}, v_{12}). The third summand includes within it a modified version of (1). The difference being in the first term where we compute the expected travel time from u_{11} to u_{22} given 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,

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

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 the 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 the 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 [21] differs from our algorithm mainly by the following two points: 1) In [21], 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 the adjustment routes to the destination. 2) The algorithm [21] 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 models. The travel times c on the edges are known[1] and we assume the probability of low traffic and delay offsets based on the street type as depicted in Figure 11. The 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 hours 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 eliminate many possibilities and create truly tractable algorithms.

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 4.1. *The minimum expected travel time between two nodes s and t are 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 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 proved for the series models as well. □

[1]Source URL: http://austintexas.gov/department/gis-and-maps/gis-data

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

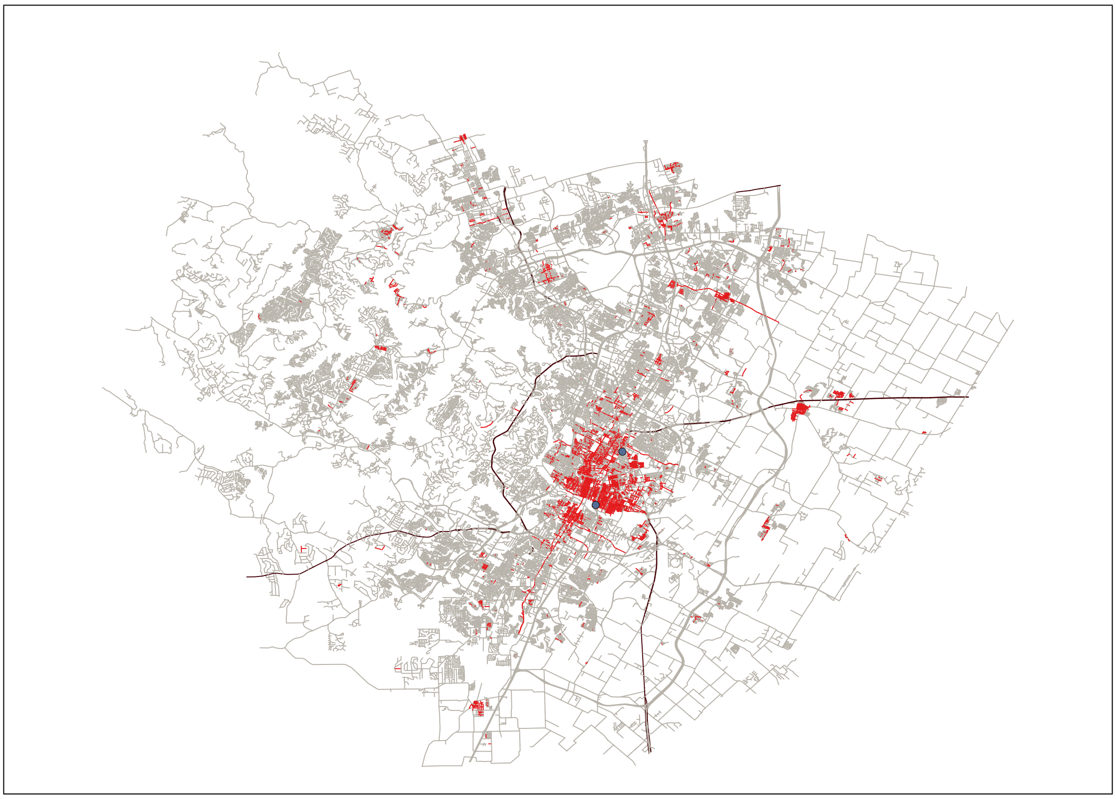


Figure 11. Map of Austin Road Network: Brown colored edges assume $p = 0.4$ and $d = 5 * c$. Red colored edges assume $p = 0.5$ and $d = 4 * c$. Remaining edges assume $p = 0.6$ and $d = 3 * c$. Blue solid dots represent an example source-destination pair.

Let us assume 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 at least ρ. Under k route adjustments, every path occurs with probability at least ρ^k. This leads to the following lemma defining a lower bound on any policy that uses edge (i, j).

Lemma 4.2. *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 4.3. *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 4.2, 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 4.1, π is not an optimal routing policy. Hence the edge (i', j') will not be on any path of the optimal policy π. □

Using the result of Theorem 4.3, one can prune the network eliminating several possibilities. For the example source-destination pair considered, and for $k = 1$ and $\rho = 0.4$, the network is pruned to 17,328 edges.

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. To do this, we employ different lower bounds as discussed in this section.

Lemma 4.4. *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 the 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 4.5. *(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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

the policy π' has length $\mathbb{E}[s \to u]+\mathbb{E}[u \to t|d_{uv}] < \mathbb{E}[s \to u]+(1-p_{uv})\mathbb{E}[u \to t|d_{uv}]+p_{uv}(c_{uv}+Z^{\text{suf}}_{k-1}[v \to t])$, implying π is not optimal. Thus $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + Z^{\text{suf}}_{k-1}[v \to t]$.

Using this result we have,

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

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

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

It is important to note that (14) holds for all route adjustment models since $Z^{\text{suf}}_1[i \to j] = Z^{\text{sf}}_1[i \to j] = Z^{\text{pll}}_1[i \to j] = Z_1[i \to j]$.

The potential savings in travel time from i to j due to two route adjustment policy is given by,

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

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

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

Extending this logic to any k yields,

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

□

Lemma 4.6. *(Series forced model) An optimal route adjustment policy π with edge (u, v) as its last adjustment edge has $Z_k^{\text{sf}}(\pi) \geq LBA_k^{\text{sf}}(u, v)$, where $LBA_k^{\text{sf}}(u, v)$ is given by*

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

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 4.5.

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 $Z_{k-1}^{\text{sf}}[s \to u]$ is followed by $\mathbb{E}[u \to t|d_{uv}]$. Thus the policy π' has length $Z_{k-1}^{\text{sf}}[s \to u] + \mathbb{E}[u \to t|d_{uv}] < Z_{k-1}^{\text{sf}}[s \to u] + (1-p_{uv})E[u \to t|d_{uv}] + p_{uv}(c_{uv} + \mathbb{E}[v \to t])$, 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,

$$\begin{aligned} Z_k^{\text{sf}}(\pi) &= Z_{k-1}^{\text{sf}}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}] \\ &\geq Z_{k-1}^{\text{sf}}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})(c_{uv} + \mathbb{E}[v \to t]) \\ &\geq Z_{k-1}^{\text{sf}}[s \to u] + c_{uv} + \mathbb{E}[v \to t]. \end{aligned} \tag{15}$$

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

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

The penultimate inequality is due to (14). Combining this result with (15) for $k = 3$ yields,

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

By extending the logic to a generic k, we obtain

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

□

Lemma 4.7. *(Parallel model) An optimal route adjustment policy π with edge (u, v) as its first adjustment edge has $Z_k^{\text{pll}}(\pi) \geq LBA_k^{\text{pll}}(u, v)$ and $Z_k^{\text{pll}}(\pi|D) \geq LBA_k^{\text{pll}}(u, v)$, where $LBA_k^{\text{pll}}(u, v)$ is given by*

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

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

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

We show $Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}] \geq c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]$, by contradiction. Assume $Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}] < c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]$. 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 $Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}]$. Thus the policy π' has length $\mathbb{E}[s \to u] + Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}] < \mathbb{E}[s \to u] + (1 - p_{uv})Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}] + p_{uv}(c_{uv} + Z_{k-1}^{\text{pll}}[v \to t])$, implying π is not optimal. Thus $Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}] \geq c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]$.

Using this result we have,

$$\begin{aligned}
Z_k^{\text{pll}}(\pi) &= \mathbb{E}[s \to u] + (1 - p_{uv})Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}] + p_{uv}(c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]) \\
&\geq \mathbb{E}[s \to u] + (1 - p_{uv})(c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]) + p_{uv}(c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]) \\
&\geq \mathbb{E}[s \to u] + c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]. \qquad (16)
\end{aligned}$$

We follow the same procedure as in proof of Lemma 4.5 and 4.6 to obtain a lower bound on $Z^{\text{pll}}_{k-1}[v \to t]$. The potential saving in travel time from i to j due to two route adjustment policy using (8),is given by

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

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 saving in travel time from i to j due to three route adjustment policy is given by,

$$\begin{aligned}
Z_2^{\text{pll}}[i \to j] - Z_3^{\text{pll}}[i \to j] &\leq \max_{(u,v)\in A} \big(p_{uv}(Z_1[v \to j] - Z_2^{\text{pll}}[v \to j]) \\
&\qquad + (1 - p_{uv})(Z_1[u \to j|d_{uv}] - Z_2^{\text{pll}}[u \to j|d_{uv}])\big) \\
&\leq \max_{(u,v)\in A} \big(p_{uv}(\alpha \int[j] + \S[j]) + (1 - p_{uv})\mathbb{E}[u \to j|d_{uv}]\big) \\
&\leq \alpha^2 \int[j] + \S[j].
\end{aligned}$$

The penultimate inequality is due to $Z_2^{\text{pll}}[u \to j|d_{uv}] \geq 0$ and $Z_1[u \to j|d_{uv}] \leq \mathbb{E}[u \to j|d_{uv}]$. Using the above results in (16) for $k = 4$, we get

$$\begin{aligned}
Z_4^{\text{pll}}(\pi) &\geq \mathbb{E}[s \to u] + c_{uv} + Z_3^{\text{pll}}[v \to t] \\
&\geq \mathbb{E}[s \to u] + c_{uv} + Z_2^{\text{pll}}[v \to t] - \alpha^2 \int[t] - \S[t] \\
&\geq \mathbb{E}[s \to u] + c_{uv} + Z_1^{\text{pll}}[v \to t] - \alpha \int[t] - \S[t] - \alpha^2 \int[t] - \S[t] \\
&\geq \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t](1 + \alpha + \alpha^2) - 2\S[t].
\end{aligned}$$

By similar logic we derive for any k,

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

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

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

Theorem 4.8. *An edge (u', v') with $LBA_1(u', v') > \mathbb{E}[s \to t]$ or $LBA_k^{\text{M}}(u', v') > Z_{k-1}^{\text{M}}[s \to t]$, cannot be the first adjustment edge (for M being series unforced or parallel model) or the last adjustment edge (for M being series forced model) in an optimal routing policy.*

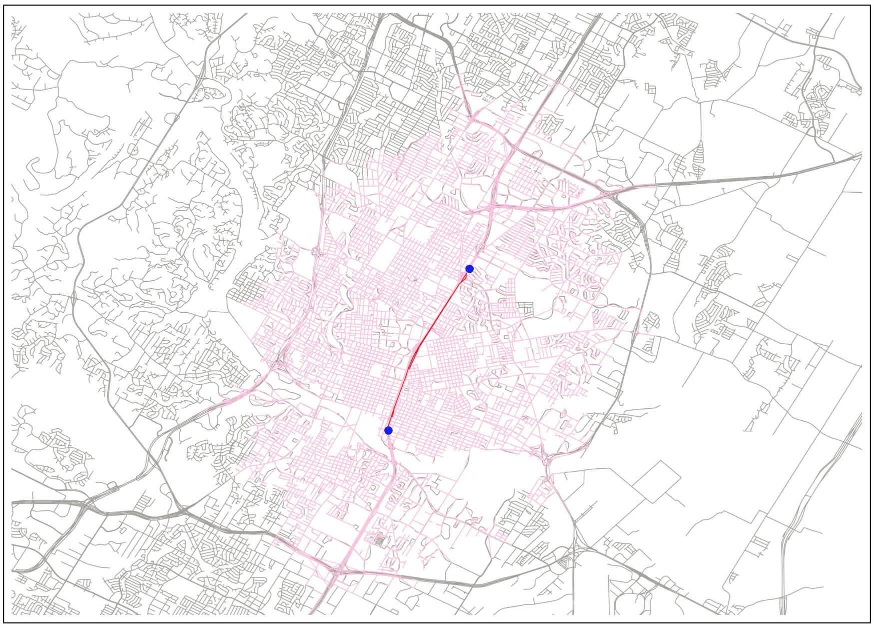


Figure 12. Pruned network and the set of critical adjustment edges for the single route adjustment policy: Shaded portion represents the pruned network and red solid line represents the set of feasible adjustment edges.

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 $LBA_k^{\text{suf}}(u', v') > Z_{k-1}^{\text{suf}}[s \to t]$.

If π is optimal, we have $Z_k^{\text{suf}}(\pi) \geq LBA_k^{\text{suf}}(u', v')$, using the result of Lemma 4.5. Since $LBA_k^{\text{suf}}(u', v') > Z_{k-1}^{\text{suf}}[s \to t]$ (by assumption), we have $Z_k^{\text{suf}}(\pi) > Z_{k-1}^{\text{suf}}[s \to t]$, implying π is not optimal. This completes our proof. Similar logic can be used along with Lemma 4.6 and Lemma 4.7 to prove this claim for series forced and parallel route adjustment models. □

We can now apply the result of Theorem 4.8 to find a set of feasible adjustment edges in the pruned network. For the example source-destination pair and $k = 1$, we obtain a set of 21 feasible adjustment edges, as shown in Figure 12.

This pre-processing step of 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 seconds to prune the network from 108,000 edges to 17,328 edges and to find a set of 21 feasible edges. As a result, the algorithm computes the optimal single route adjustment policy in less than 15 seconds. The solution pertaining to the example considered is presented in Figure 13.

For the same source-destination pair, $k = 2$ and $\rho = 0.16$, Theorem 4.3 prunes the original network to 50,628 edges and Theorem 4.8 yields a set of 2091 feasible adjustment edges. The solutions are presented in

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

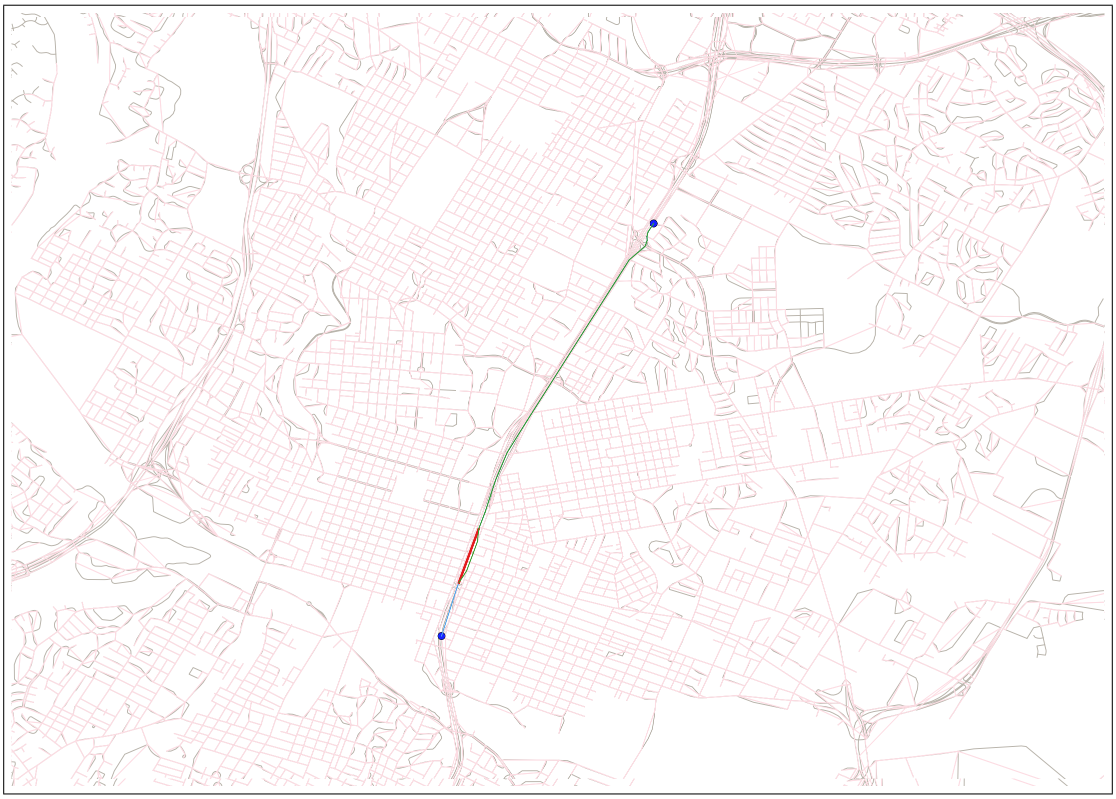


Figure 13. Optimal single route adjustment policy: Red solid line represents the optimal adjustment edge. Blue and green lines represent the non-adjusted and the adjusted shortest route respectively.

Figure 14

4.3 Performance Evaluation

To summarize the performance of our algorithms with two adjustment edges, parallel model performs better than the other two models 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 no-adjustment shortest path. Series forced model yields about 3% saving compared to the single route policy and 9.5% saving compared to the no-adjustment path. Finally, series unforced model provides least saving of about less than 1% and 7% respectively.

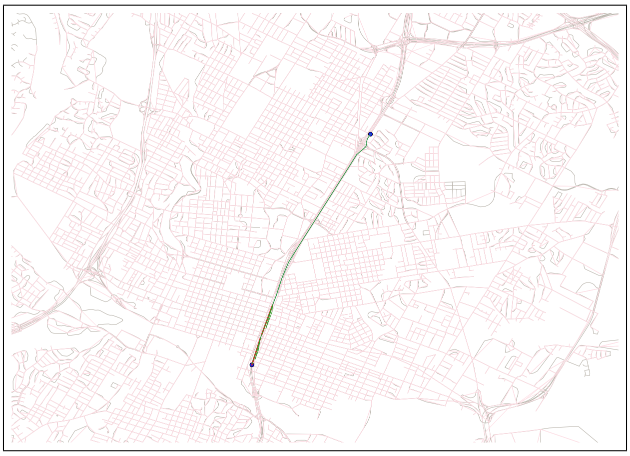
One can achieve more savings with increasing number of route adjustments, however with a large increase in the computational effort. For example, the pruned network size for two adjustment edges is about 50% of the original network size and that of the three edges is almost the same as the original network, increasing the computational effort drastically. Thus a trade-off arises between the number of edges to be observed for traffic and the potential savings in expected travel times. In order to understand this trade-off, we solve the dynamic programming algorithms, for different route adjustment models, on a smaller network consisting of 17,328 edges (given by the pruned network of 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 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 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 7 edges in the network, as opposed to CTP where all the 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 model 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 the expected travel times.

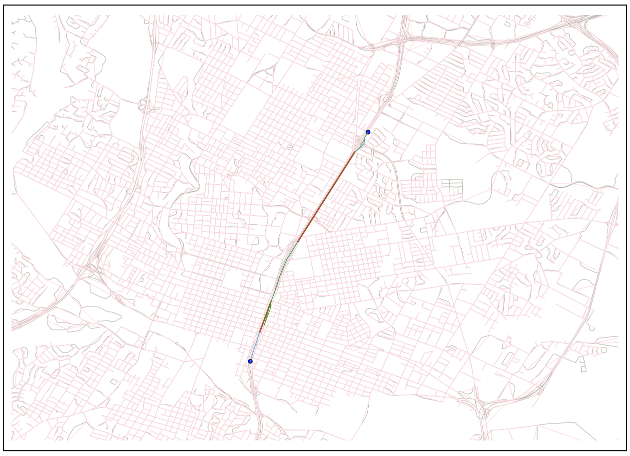
5 Conclusion

In this paper, we deal with the problem of developing non-aggressive adaptive routing models which has limited route adaptability and requires limited decision making. To address this problem, we propose multiple routing strategies which we call *series unforced, series forced and parallel models*, depending on where and

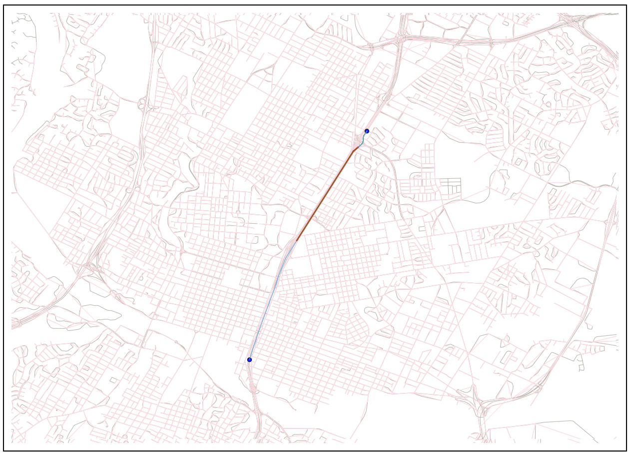
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1



(a) Solution to series unforced model.



(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 the adjusted shortest routes respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2023 doi:10.20944/preprints202305.0974.v1

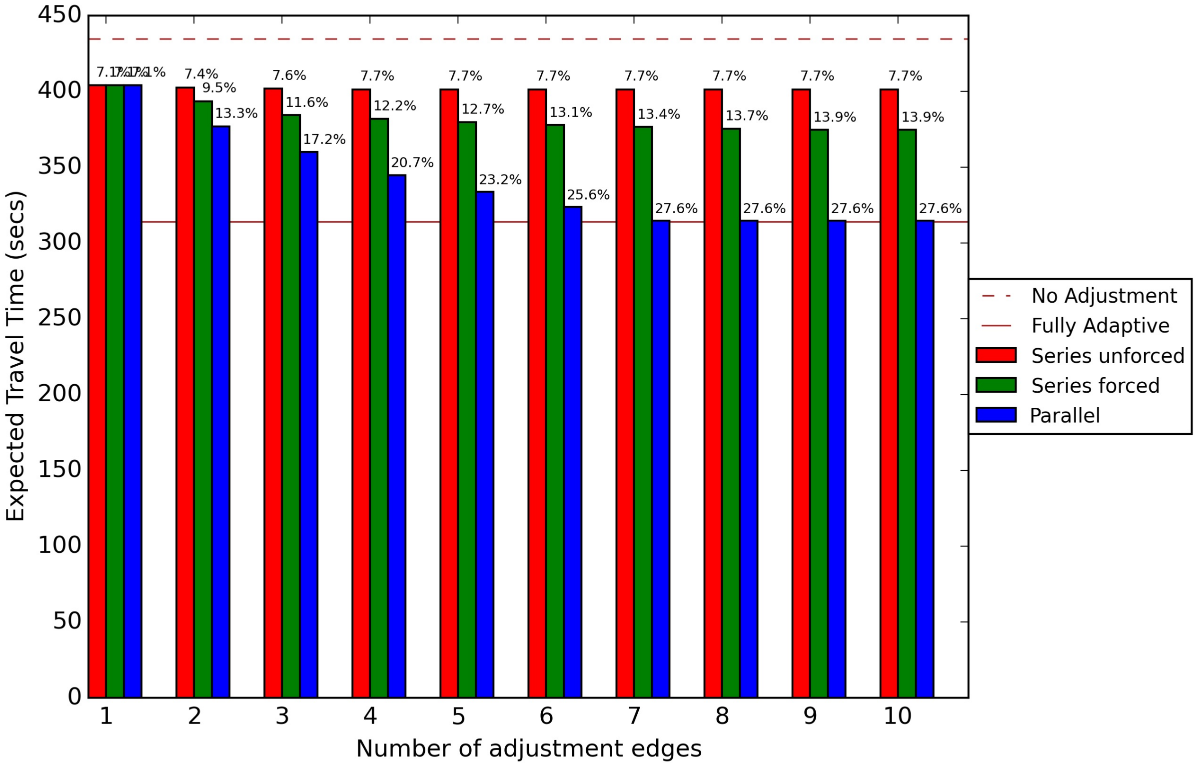


Figure 15. Benefit of Adaptability: This graph summarizes the expected travel time across varying number of adjustment edges. Brown dashed and brown solid lines represent the non-adaptive and completely adaptive expected travel times. Red, green and blue bars represent the summary of series unforced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

 doi:10.20944/preprints202305.0974.v1

how the route adjustments are performed. The main goal of these strategies is to determine the set of k best adjustment edges and the corresponding adjustment and non-adjustment routes, that minimize the expected travel time.

To achieve this goal, we propose exact mathematical models such as complete enumeration and dynamic programming algorithms for each of the aforementioned strategies. While the complete enumeration method is an exponential time algorithm with complexity roughly $O(m^k)$, we propose polynomial time dynamic programming algorithms with complexity $O(mk)$ (where $m = |A|$ and k is the number of adjustment edges). These dynamic programming algorithms seem tractable for small to medium sized networks, however finding solutions for large networks is difficult and rather quite intractable. Thus, we develop easily computable bounds and present several theorems allowing us to reduce the size of network and to find a set of potential adjustment edges. These results lead to tractable algorithms, reducing the computational effort to handle large-sized networks. We numerically assess the performance of our models using single and two route adjustment policies, and present the benefit of adpatability graph for an example source-destination pair in the Austin road network.

In terms of future extensions of this work, there is scope to improve the tractability of the algorithm by developing tighter lower bounds. Another area to explore is to study other multiple route adjustment strategies. For example, one can consider a model where the driver is constrained to switch back and forth between two or more pre-computed routes. Furthermore, instead of assuming the delay data, realistic delay times and realistic probability distribution of traffic can be used from suitable sources. In fact, factoring in the traffic information for different times of a day might yield better results. Another area for extension is to account for the correlation in the traffic states between different edges.

Additional Information

Data Availability

Data used in this paper is accessed from open data sources. Web urls to the data sources are provided in the reference section.

Funding

Not applicable.

Conflict of Interest

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

References

[1] Austin, texas population 2018. http://worldpopulationreview.com/us-cities/austin/, 06 2018. [Online; posted 03-June-2018].

[2] Will Anderson. Population explosion: Austin growth passes major milestone. https://www.bizjournals.com/austin/news/2015/12/11/population-explosion-austin-growth-passes-major.html, 12 2015. [Online; posted 11-Dec-2015].

[3] Will Anderson. Austin's population keeps popping; here's how many people are added each day. https://www.bizjournals.com/austin/news/2018/03/22/austins-population-keeps-popping-heres-how-many.html, 03 2018. [Online; posted 22-Mar-2018].

[4] Giovanni Andreatta and Luciano Romeo. Stochastic shortest paths with recourse. *Networks*, 18(3):193–204, 1988.

[5] Amotz Bar-Noy and Baruch Schieber. The canadian traveller problem. In *SODA*, volume 91, pages 261–270, 1991.

[6] James Barragan. Austin metro area surpasses 2 million residents. https://www.statesman.com/NEWS/20160904/Austin-metro-area-surpasses-2-million-residents, 03 2016. [Online; posted 24-March-2016].

[7] David Meir Blei and Leslie Pack Kaelbling. Shortest paths in a dynamic uncertain domain. In *IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments*, volume 4, page 2, 1999.

[8] Adi Botea, Evdokia Nikolova, and Michele Berlingerio. Multi-modal journey planning in the presence of uncertainty. In *ICAPS*, 2013.

[9] Justin Boyan and Michael Mitzenmacher. Improved results for route planning in stochastic transportation. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 895–902. Society for Industrial and Applied Mathematics, 2001.

[10] Sebastien Boyer, Sebastien Blandin, and Laura Wynter. Stability of transportation networks under adaptive routing policies. *Transportation Research Part B: Methodological*, 81:886–903, 2015.

[11] Stephen D Boyles and S Travis Waller. Optimal information location for adaptive routing. *Networks and Spatial Economics*, 11(2):233–254, 2011.

[12] City of austin population history. https://www.austintexas.gov/sites/default/files/files/Planning/Demographics/population_history_pub.pdf, 2016. [Online; Accessed 29-July-2016].

[13] John S Croucher. A note on the stochastic shortest-route problem. *Naval Research Logistics Quarterly*, 25(4):729–732, 1978.

[14] Song Gao. *Optimal adaptive routing and traffic assignment in stochastic time-dependent networks*. PhD thesis, Massachusetts Institute of Technology, 2005.

[15] Song Gao and Ismail Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.

[16] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[17] Keeping austin weird shouldn't include sitting in traffic for 41 hours. http://inrix.com/blog/2014/03/keeping-austin-weird-shouldnt-include-sitting-in-traffic-for-41-hours/, March 2014. [Online; Accessed 01-Aug-2016].

[18] Elise Miller-Hooks. Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks: An International Journal*, 37(1):35–52, 2001.

[19] Philippe Nain, Don Towsley, Amotz Bar-Noy, Prithwish Basu, Matthew P Johnson, and Feng Yu. Estimating end-to-end delays under changing conditions. In *Proceedings of the 8th ACM MobiCom workshop on Challenged networks*, pages 1–6. ACM, 2013.

[20] Yu Nie and Yueyue Fan. Arriving-on-time problem: discrete algorithm that ensures convergence. *Transportation Research Record*, 1964(1):193–200, 2006.

[21] Evdokia Nikolova and David R Karger. Route planning under uncertainty: The canadian traveller problem. In *AAAI*, pages 969–974, 2008.

[22] Christos H Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150, 1991.

[23] George H Polychronopoulos and John N Tsitsiklis. Stochastic shortest path problems with recourse. *Networks*, 27(2):133–143, 1996.

[24] Daniele Pretolani. A directed hypergraph model for random time dependent shortest paths. *European Journal of Operational Research*, 123(2):315–324, 2000.

[25] J Scott Provan. A polynomial-time algorithm to find shortest paths with recourse. *Networks: An International Journal*, 41(2):115–125, 2003.

[26] Samantha Sharf. America's fastest-growing cities 2018. https://www.forbes.com/sites/samanthasharf/2018/02/28/full-list-americas-fastest-growing-cities-2018/#7ab24e8b7feb, 02 2018. [Online; posted 28-Feb-2018].

[27] United states census bureau. https://www.census.gov/popest/about/geo-topics.html, 05 2015. [Online; posted 21-May-2015].

[28] Yinfeng Xu, Maolin Hu, Bing Su, Binhai Zhu, and Zhijun Zhu. The canadian traveller problem and its competitive analysis. *Journal of combinatorial optimization*, 18(2):195–205, 2009.

A Appendix - Integer Programming Formulations

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 and Z_{uv} is 1 if edge (u, v) is an adjustment edge and 0 otherwise.

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 and Z^l_{uv} is 1 if edge (u, v) is an adjustment edge at l^{th} 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.