Online Route Replanning for Scalable System-Optimal Route Planning

Robert J. Fitzgerald
University of Colorado Denver
Denver, Colorado, USA
robert.fitzgerald@ucdenver.edu

Farnoush Banaei-Kashani
University of Colorado-Kashani
Denver, Colorado, USA
farnoush.banaei-kashani@ucdenver.edu

ABSTRACT
Route planning in transportation networks is typically performed as a single optimization at trip departure. In this paper, we consider the impact of within-trip replanning on the performance of the overall network in a fully-algorithmic route selection scenario. An experimental study of three real road networks using synthetic demand demonstrates in over 200 trials the effects of replanning with respect to the replanning rate and the adoption rate of replanning. Overall network travel times are reduced by up to 48.49% from a baseline where all drivers are assigned a single route, demonstrating the profound effect of dynamic within-trip replanning. These observations are part of our work exploring a system-optimal route planning strategy that is robust to network size and conditions.

CCS CONCEPTS
• Applied computing → Transportation; • Networks → Traffic engineering algorithms; • Information systems → Online analytical processing.

KEYWORDS
route planning, online optimization, system-optimal, transportation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGSPATIAL ‘21, November 2–5, 2021, Beijing, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8664-7/21/11...
https://doi.org/10.1145/3474717.3484262

1 INTRODUCTION
In our previous work [8], we explored transportation route planning with a system-optimal objective using a batch formulation. This technique, while effective in small settings, fails to scale. We hypothesize that this is caused by route plans becoming stale under unstable network conditions, coupled with the fact that average routes are longer in larger networks. In order to address this, one can allow for within-trip route replanning, which allows the route planning algorithm to periodically update the optimality of the chosen route plan for each driver.

2 RELATED WORK
In order to serve route plans in an online context, basic graph search techniques (such as Dijkstra’s, Bellman-Ford, and A*) have been extended with improvements which address the trade-off between query time and pre-compute space/time. The simplest technique to minimize query time is to create a lookup table of all possible route queries. This can be achieved by running the Floyd-Warshall algorithm, which produces a massive memory allocation. Parallel optimizations, such as PHAST [5], exist to address this constraint. Smaller sets of data can also be stored effectively on the vertices with linear performance which nears lookup table queries, such as in Hub Labeling (HL) [4].

Some query time optimizations require a small sweep of the graph, but provide optimizations which do not have large trade-offs in terms of memory footprint and pre-computation time, and are reasonably competitive in query times to PHAST and HL. For example, Contraction Hierarchies (CH) [9] produces a hierarchy of hyper-graphs, limiting the search space as the search ascends the hierarchy. Customizable Route Planning (CRP) [6] is more robust to changes to network flows, due to the separation of pre-computation into a bootstrapping step and an update step. For more details on these techniques, we refer the reader to a comprehensive route planning survey [2].

Extending a route plan with a within-trip replanning is simply a re-computation of one of these search techniques using dynamic network speed data.

3 REPLANNING
Let $G$ represent a road network, as a directed, fully-connected, finite graph with vertices $V(G)$ and edges $E(G) \subseteq V(G) \times V(G)$. Each vertex $v \in V(G)$ represents an intersection on a road network map. Each edge $e = (u, v)$ $\forall e \in E(G)$ represents a road segment traversing the road network between two vertices $u, v \in V(G)$. Each edge has a positively-valued and monotonically increasing edge cost/flow function $C(x)$.
Let $t$ be a time in the range $[t_0, T]$, where $t_0$ and $T$ mark the start and end of the simulation, respectively. The time $t$ is a point along a sequence of times with a uniform step size set by the batch window parameter, denoted $b$. Let $R_t$ be a finite batch of requests at time $t$, where each request $r \in R_t$ captures the state of some driver at time $t$ along a route between an origin $O(r)$ and destination $D(r)$, where $O(r), D(r) \in V(G)$. The number of agents in batch $R_t$ is a factor of the overall adoption rate $a$ of the replanning algorithm. This driver agent may be recently departing or somewhere along their trip, and may have already received a route plan at some previous time. Each request $r$ is assigned a path $p_r$, which is a vector of the form $<v_0, v_1, ..., v_n, v_n>$, where $v_0 = O(r), v_n = D(r), (v_k, v_{k+1}) \in E(G) \forall 0 \leq k < n, v_n > 0$.

The solution to this problem finds a path $p_r$ using Dijkstra’s Algorithm [7], with the current road network conditions denoted $G_t$, and with edge costs $C$ evaluated using the Bureau of Public Roads cost/flow function [1] as shown in Equation 1, where $q$ is the road segment capacity per hour and $f$ is the free-flow speed per hour for this edge.

$$C(x) = \frac{fb}{3600} \left( 1 + 0.15 \left( \frac{x}{qb/3600} \right)^4 \right)$$  \hspace{1cm} (1)

The complete set of definitions is listed in Table 1.

### Table 1: Definitions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>a directed, connected, finite graph</td>
</tr>
<tr>
<td>$v \in V(G)$</td>
<td>a vertex in the vertex set of $G$</td>
</tr>
<tr>
<td>$e \in E(G)$</td>
<td>an edge in the edge set of $G$</td>
</tr>
<tr>
<td>$C(x)$</td>
<td>edge cost/flow function</td>
</tr>
<tr>
<td>$b$</td>
<td>batch window (seconds)</td>
</tr>
<tr>
<td>$t$</td>
<td>current time (seconds)</td>
</tr>
<tr>
<td>$a$</td>
<td>algorithm adoption rate (%)</td>
</tr>
<tr>
<td>$r \in R_t$</td>
<td>a request in the set of agent routing requests</td>
</tr>
<tr>
<td>$O(r)$</td>
<td>origin vertex of request $r$</td>
</tr>
<tr>
<td>$D(r)$</td>
<td>destination vertex of request $r$</td>
</tr>
<tr>
<td>$r_t$</td>
<td>time of request $r$</td>
</tr>
<tr>
<td>$p_r$</td>
<td>a path between vertex $v_0$ and $v_n$ for request $r$ of the form $&lt;v_0, v_1, ..., v_{n-1}, v_n&gt;$</td>
</tr>
<tr>
<td>$f$</td>
<td>free-flow edge speed</td>
</tr>
<tr>
<td>$q$</td>
<td>edge capacity</td>
</tr>
</tbody>
</table>

### 4 EXPERIMENTS

#### 4.1 Experimental Methodology

A set of 228 randomized trials were run using the open-source traffic simulator MATSim [10] as an evaluation platform. MATSim is a highly-extensible, open-source mesoscopic traffic simulation framework with an active and supportive community. Mesoscopic traffic simulation represents the traffic flows in a queue simulation, which is fine-grained enough to capture route plans for individual agents and the wave propagation of congestion effects. In this section, the supply and demand modeling details will be discussed, followed by a walk-through of the experimental procedure.

Three road networks of different sizes were collected from OpenStreetMap [11], a crowd-sourced data repository of map data. The resulting road networks are shown in Figure 1 and their attributes summarized in Table 2. Each road network was cleaned and modified for MATSim using the OSMnx Python library [3] as well as a built-in MATSim `OsmNetworkReader` tool for parsing OSM networks.

The networks were downloaded in two phases. First, all ‘drive’ network links within the city boundary are downloaded. Second, the city boundary is extended by 1 kilometer and only primary, secondary, and trunk links are downloaded. The two networks are then combined, dismissing redundant overlapping links. This process, as described in section 7.2.1 of the 2016 MATSim book [10], prevents creating road networks with unwanted boundary flow behaviors. The final network is encoded in the Web Mercator coordinate system (EPSG:3857), which approximates a meter unit distance and is acceptable for modeling in MATSim. MATSim was used as-is, without activating any external modules, such as left turn delays and traffic signals.

Uniformly random demand was chosen to help test the robustness of the algorithm to arbitrary loads. Activity locations were randomly sampled from the road network links, and work departure times were also uniformly sampled in the range [8:30am, 9:30am] with a 8 hour work activity, to approximate a demand curve with peaks during morning and evening commutes. Population sizes were randomly sampled for each scenario in a range which creates low to high congestion effects.

In each random experiment, two simulations are run, with and without replanning. For both simulations, a random road network is selected. A synthetic population of a random size is generated which is shared by both simulations. The size is chosen to fit within a range which we observe to produce reasonable congestion effects under baseline conditions, which we quantify as between a 1-3x increase in travel times from those experienced in a network with free flow speeds (no congestion). In the replanning simulation, a random adoption rate $a \in [0, 1]$ and batch window $b \in [5, 600]$ are selected.

To measure the performance of a configuration, the travel times with replanning were compared on an agent-by-agent basis between the result without replanning. The average travel time difference $TT^*$ is calculated, as shown in Equation 2, where $\alpha$ is a driver agent sampled from the entire population $A$, $w(\alpha)$ is the observation without replanning, and $r(\alpha)$ is the result with replanning. When reviewing results showing $TT^*$, lower scores show greater improvement to travel time.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Golden</th>
<th>Lafayette</th>
<th>Boulder</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertices $</td>
<td>V(G)</td>
<td>$</td>
<td>904</td>
</tr>
<tr>
<td>edges $</td>
<td>E(G)</td>
<td>$</td>
<td>2114</td>
</tr>
<tr>
<td>sum of link lengths, miles</td>
<td>219.84</td>
<td>286.77</td>
<td>710.24</td>
</tr>
<tr>
<td>average link length, feet</td>
<td>568</td>
<td>433</td>
<td>475</td>
</tr>
<tr>
<td>average link speed, mph</td>
<td>19.08</td>
<td>18.14</td>
<td>20.29</td>
</tr>
</tbody>
</table>

### Table 2: Road network statistics
Online Route Replanning for Scalable System-Optimal Route Planning

SIGSPATIAL ’21, November 2–5, 2021, Beijing, China

Figure 1: Road networks for (a) Golden, (b) Lafayette, and (c) Boulder, all cities in Colorado, USA

\[ TT^* = \frac{\sum_{\alpha \in A} r(\alpha) - w(\alpha)}{|A|} \]  

(2)

All algorithms were evaluated using a Dell Optiplex 790 desktop computer, with quad-core i7 processors, 16GB of RAM, an 500GB SSD hard drive, and 8M L3 cache, running Ubuntu 16.04.3. MATSim 12.0 was run in standard queue simulation mode using the QSim network engine, with all algorithmic extensions to MATSim implemented in Scala by the authors.

4.2 Experimental Results

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Mean $TT^*$</th>
<th>Min $TT^*$</th>
<th>Trials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golden</td>
<td>9.78%</td>
<td>-5.63%</td>
<td>84</td>
</tr>
<tr>
<td>Lafayette</td>
<td>-10.03%</td>
<td>-24.89%</td>
<td>78</td>
</tr>
<tr>
<td>Boulder</td>
<td>-29.84%</td>
<td>-48.49%</td>
<td>66</td>
</tr>
</tbody>
</table>

Table 3: Travel Time Difference $TT^*$ by Scenario

Across the set of experimental trials, a trend emerges that shows travel time improvement with increased adoption $a$ and smaller batch windows $b$ for trials set in Lafayette and Boulder. However, the opposite is observed with the smaller road network of Golden. We hypothesize that there are boundary conditions where route plans in constrained networks may produce trips which oscillate under replanning. In order to address this, a coordinated route planning objective is required, such as the system-optimal route planning approach described in our past research [8]. Explorations of this are ongoing and the subject of future work.

In Figure 2, the $TT^*$ values for all experiments are shown for each road network. The results are summarized in Table 3. Golden is the smallest network, and does not show $TT^*$ improvement on average. Alternatively, the worst-performing trials for both Lafayette and
Boulder do not result in as dramatically increased travel times. The Lafayette network, which is slightly larger than Golden, shows an average $TT^*$ of -10.03%, and in Boulder, this is extended to 29.84%.

In the following figures we observe these effects as a function of the adoption rate $a$ and batch window $b$ parameters.

The adoption rate $a$, as shown in Figure 3, shows that a majority adoption of route replanning creates the most pronounced effect. There is also evidence that event low amounts of adoption, such as 10-20%, may lead to improved travel times in larger road networks. This is useful, as it fits the scenario where an organized business fleet or mobility-as-a-service (MaaS) provider may be suited to uniformly adopt algorithmic route planning strategies.

In Figure 4, the batch window $b$ is shown. For all scenarios, it is clear that batch windows greater than roughly 2 minutes result in a $TT^*$ of 0%. We hypothesize that, as the rate of updates slow, the assigned routes are persisted longer, leading to decaying optimality in the route plans. For all scenarios, $b < 30$ shows the greatest result, implying a real-time route replanning algorithm should aim to make decisions in an under-minute time frames. Replanning at this frequency has not yet been explored by mapping applications in the real world and may not be suitable for human-driven vehicles, but it could become a promising area of research for the planning of autonomous vehicle routing.

5 CONCLUSIONS AND FUTURE WORK

A study of the effects of replanning in a traffic simulation have been presented. The effect of replanning can be a net positive for network congestion, improving the average travel time, but it can create more problems under certain conditions, such as a small congested network with limited route alternatives. The results do show a moderate-sized road network has the capability to reduce travel times up to 48.49% compared to a traditional single route assignment. Research in applying replanning in a system-optimal route planning setting is ongoing.

ACKNOWLEDGMENT

The work presented in this paper was conducted with support from University of Colorado Denver and the Mountain-Plains Consortium, a University Transportation Center funded by the U.S. Department of Transportation. The contents of this paper reflect the views of the authors, who are responsible for the facts and accuracy of the information presented.

REFERENCES