Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence

Self-Aware Traffic Route Planning

David Wilkie and **Jur van den Berg** and **Ming Lin** and **Dinesh Manocha**
University of North Carolina at Chapel Hill. E-mail: {wilkie, berg, lin, dm}@cs.unc.edu.

http://gamma.cs.unc.edu/TROUTE

Abstract

One of the most ubiquitous AI applications is vehicle route planning. While state-of-the-art systems take into account current traffic conditions or historic traffic data, current planning approaches ignore the impact of their *own* plans on the future traffic conditions. We present a novel algorithm for *self-aware* route planning that uses the routes it plans for current vehicle traffic to more accurately predict future traffic conditions for subsequent cars. Our planner uses a roadmap with stochastic, time-varying traffic densities that are defined by a combination of historical data and the densities predicted by the planned routes for the cars ahead of the current traffic. We have applied our algorithm to large-scale traffic route planning, and demonstrated that our self-aware route planner can more accurately predict future traffic conditions, which results in a reduction of the travel time for those vehicles that use our algorithm.

Introduction

One of the most ubiquitous applications of AI is vehicle route planning. State-of-the-art route planners consider possible delays due to traffic congestion based on current traffic conditions and/or historical traffic data. Live traffic data can be collected by loop-detectors, cameras, toll port data, and cell phone localization. These systems provide the traffic velocity at certain locations at a fixed frequency (Brakatsoulas et al. 2005), which can then be used for vehicles to plan around congested areas. Live data alone does not enable predicting future traffic conditions. For example, if a route is planned to let a car arrive at a certain road in half an hour, the current conditions may no longer be an accurate estimate for that road 30 minutes later. This problem can be addressed by using a prediction scheme of the future traffic conditions based on historical probabilistic data of traffic conditions at similar times of the day under the similar weather (Horvitz et al. 2005), (Nikolova, Brand, and Karger 2006), (Min, Wynter, and Amemiya 2007).

However, given a large-scale system view of the entire road network and the traffic in the system, such an approach still has a problem: a route planner can affect future traffic conditions by planning for a large portion of the vehicles, thus making prediction based on current and historical data insufficient. Instead, the route planner must also take into account its own previous actions. For example, if a route planner controlled *every* car in the system, historically congested areas would be unduly avoided, causing congestion to appear at the routes that the planning system has provided. This is clearly the worst-case scenario, but the underlying problem is that the historical prediction assumes that cars tend to act the same way as they have historically, which may no longer be the case if a route planner is controlling the trajectories of all vehicles. We propose a novel *self-aware* traffic route planner that uses the routes of vehicles that it has planned so far to more accurately predict future traffic conditions for vehicles whose routes are subsequently planned (see Fig. 1). As a result, our approach overcomes the oscillation issue in case of large-scale adoption of traffic route planners.

Our self-aware approach accounts for the fact that a route planned for a car will cause a little extra traffic density at the roads it will traverse. We use the predicted paths of the route planner itself in addition to historical data to estimate future traffic conditions. Assuming that a large percentage of the cars use the route planning system, the collection of all their planned routes can be used to accurately estimate the future traffic conditions. Every car that queries the route planner can then use this information to plan a route for itself. Its planned route is then used to update the estimate of future traffic conditions for vehicles come in later in the road network. Our experimental results suggest that our self-aware route planner can more accurately predict future traffic conditions, resulting in a reduction of the travel time for those vehicles that use our algorithm.

The rest of the paper is organized as follows. In Section 2, we discuss background work related to our approach. In Section 3, we detail the method used to update the historical probabilistic prediction and the overall planning system. In Section 4, we discuss the implementation and validation of our method.

Prior Work and Background

Our work is perhaps most similar to that of (Nikolova et al. 2006) and (Lim et al. 2009). In fact, our work directly extends these methods to perform 'self-aware' routing. In (Nikolova et al. 2006), the authors propose a method to optimally route cars given uncertain information about travel times within a network. (Lim et al. 2009) provides an optimization to the procedure that allows fewer paths to be explored, while optimizing for a specific arrival deadline, and additional extensions were carried out in (Nikolova 2010) and (Hua and Pei 2010).

Another area of similar work is the study of Dynamic Traffic Assignment (DTA) done primarily in Civil Engineering. This problem involves flows of traffic from known origins to destinations (OD flows). The solution approaches attempt to optimally route all the flows in order to maximize aggregate or individual statistics. A summary of approaches can be found in (Peeta and Ziliaskopoulos 2001). The most relevant of these approaches are the simulation methods, such as (Florian, Mahut, and Tremblay 2008). In these approaches, cars are iteratively routed and simulated. The simulation provides the estimate of the network state that is used for the next iteration of routing. Over a number of iterations, the routes settle into an equilibrium.

Our work is inspired by (Lim et al. 2009), which presents an planning algorithm using graphs with stochastic (time-invariant) edge costs. Their planner assumes a cost function that invalidates the "optimal substructure" property of paths, which prevents using a straightforward A* algorithm, and present an efficient approach to compute optimal paths for such cost functions. In contrast, our algorithm uses a simpler cost function that still makes it possible to use an A* search algorithm, but assumes *time-varying* stochastic edge costs. We use insights from (Chabini and Lan 2010) regarding the first-in-first-out property of traffic that allows us to use A* even if the edge-costs are time-dependent (in general, time-dependent edge costs prohibit the use of A*). Typical traffic related planning approaches assume the edge cost to be given as travel times (potentially time-varying and stochastic) (Lim et al. 2009; Chabini and Lan 2010).

Our approach assumes the input data to be traffic *densities* of the road segments in the network, and we use the *fundamental diagram* of traffic to translate these densities to velocities and travel times. Maintaining densities allows us to update the data with the routes that our system generate to create a self-aware routing system. The observation that the flow (and velocity) of traffic was dependent on the traffic density was made in early traffic studies (Greenshields and others 1935). Since then, the concept has been used as a basis for continuum traffic simulation formulations (Siebel and Mauser 2005) as well as in schemes to estimate the state of traffic given sparse data, such as cell phone localization signals(Work et al. 2010).

Approach

We assume the road network to be given as a directed graph $\mathcal{G} = (V, E)$ consisting of a set of vertices V that model road intersections and edges $E \subset V \times V$ that model road segments between intersections. Associated with each edge e is the capacity C_e, maximum speed $v_e^{\max}$, and length ℓ_e of the corresponding road segment. In addition, a stochastic function $\rho_e(t) \sim \mathcal{N}(\bar{\rho}_e(t), \tilde{\rho}_e(t))$ is maintained for all road segments e that gives a normal distribution with mean

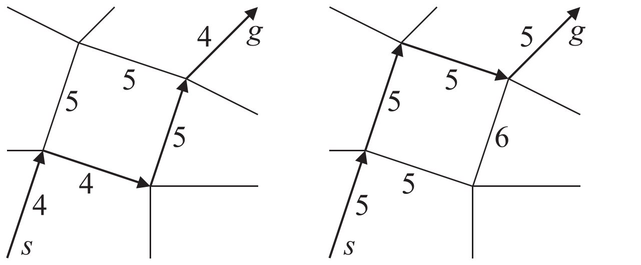

Figure 1: A schematic picture illustrating the idea of our approach. A road network is shown with edge costs. (a) If a route from s to g is requested, the optimal path is computed (shown with thick arrows). If the car follows this route, the densities and hence the edge costs along its path increases (in this case with 1). In (b), the network with the updated edge costs are shown. If a same query (s, g) comes in from a subsequent car, it takes a different route (shown with thick arrows) to avoid the increased traffic densities. Note that this schematic picture does not illustrate the stochastic and time-varying aspects of our approach.

$\bar{\rho}_e(t)$ and variance $\tilde{\rho}_e(t)$ of the traffic density of e at time t. We assume this distribution is independent from the density distributions at other road segments or at other times (similar assumptions were made in (Lim et al. 2009)). Further, we assume that the time-axis is cyclical (e.g. with a daily or weekly period) and discretized into small steps, such that only a finite amount of data is stored with each edge e. The stochasticity of the density function models the uncertainty about future traffic conditions as well as the variability of conditions from day to day.

Our approach can be summarized as follows. We assume that over time, different queries for optimal routes come in from cars that use our self-aware planning system. If a query comes in from a car i at a given time t_0, we plan a route for car i between its start node s and goal node g that optimizes a cost function based on its expected travel time given the current density functions $\rho_e(t)$ for each edge e. Subsequently, assuming this car will actually follow the route it was given, we update the density functions $\rho_e(t)$ along its route such that its presence is accounted for when a route is planned for a subsequent car $i+1$. This cycle continues indefinitely with each query coming in for an optimal route computation. As such, the planner is aware of the routes it has suggested earlier, in order to optimally estimate future traffic conditions.

We will first describe how an optimal route is planned for a car given the stochastic density functions $\rho_e(t)$. Next, we will discuss how this plan is used to update the stochastic density functions such that the presence of the car is accounted for in future plans for other cars. Finally, we will discuss how the problem of "double-counting" cars can be avoided, which would occur when a car being routed also appears in the historical data.

Route Planning

Density and Travel Time Given a query (s, g, t_0) for a car between a start node $s \in V$ and a goal node $g \in V$

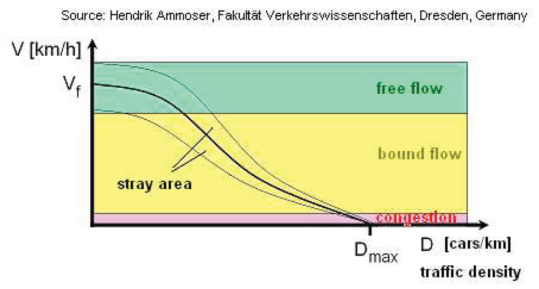

Figure 2: The fundamental diagram relating traffic density to travel speed.

leaving s at time t_0, we want to compute a route that minimizes the travel time to g given the stochastic density functions $\rho_e(t)$. To relate density to travel time, we use the *fundamental diagram*, which is a well-known empirical concept in traffic simulation research (Greenshields and others 1935; Siebel and Mauser 2005; Work et al. 2010) that gives a mapping from the traffic density ρ to the average speed v on a road segment e, given the maximum speed $v_e^{\max}$ and capacity C_e of the road segment e. Let the function described by the fundamental diagram be given by $v = f_e(\rho)$ (see Fig. 2).

Now, if a car arrives at the beginning of a road segment e at time t, we assume the speed with which it can traverse the road segment is given by $f_e(\rho_e(t))$. The travel time $\tau_e(t)$ to traverse e starting at time t is then given by:

$$\tau_e(t) = \frac{\ell_e}{f_e(\rho_e(t))}. \tag{1}$$

Since, the density function $\rho_e(t)$ is stochastic, the travel time is stochastic too. We approximate it with a normal distribution as $\mathcal{N}(\bar{\tau}_e(t), \tilde{\tau}_e(t))$, with mean $\bar{\tau}_e(t)$ and variance $\tilde{\tau}_e(t)$ given by the first-order Taylor expansion of $\tau_e(t)$:

$$\bar{\tau}_e(t) = \frac{\ell_e}{f_e(\bar{\rho}_e(t))} \tag{2}$$

$$\tilde{\tau}_e(t) = \big(\frac{d\tau_e(t)}{d\rho}[\bar{\rho}_e(t)]\big)^2 \tilde{\rho}_e(t). \tag{3}$$

For a path $\pi = \{e_1, \ldots, e_n\}$ consisting of a series of road segments when travel is commenced at time t_0, the total travel time $\tau_\pi(t_0)$ is given recursively by:

$$\tau_{\{e_1\}}(t_0) = \tau_{e_1}(t_0) \tag{4}$$

$$\tau_{\{e_1,\ldots,e_k\}}(t_0) = \tau_{\{e_1,\ldots,e_{k-1}\}}(t_0) + \tau_{e_k}(t_0 + \tau_{\{e_1,\ldots,e_{k-1}\}}(t_0)) \tag{5}$$

Its mean $\bar{\tau}_\pi(t_0)$ and variance $\tilde{\tau}_\pi(t_0)$ are hence given by:

$$\bar{\tau}_{\{e_1\}}(t_0) = \bar{\tau}_{e_1}(t_0) \tag{6}$$

$$\tilde{\tau}_{\{e_1\}}(t_0) = \tilde{\tau}_{e_1}(t_0) \tag{7}$$

$$\bar{\tau}_{\{e_1,\ldots,e_k\}}(t_0) = \bar{\tau}_{\{e_1,\ldots,e_{k-1}\}}(t_0) + \bar{\tau}_{e_k}(t_0 + \bar{\tau}_{\{e_1,\ldots,e_{k-1}\}}(t_0)) \tag{8}$$

$$\tilde{\tau}_{\{e_1,\ldots,e_k\}}(t_0) = \tilde{\tau}_{\{e_1,\ldots,e_{k-1}\}}(t_0) + \tilde{\tau}_{e_k}(t_0 + \bar{\tau}_{\{e_1,\ldots,e_{k-1}\}}(t_0)). \tag{9}$$

Cost Function Our objective is to find a route π that minimizes the expectation $\mathbb{E}[c(\tau_\pi(t_0))]$, given a cost function $c(\tau)$ on the travel time τ. We consider two cases here:

- *Linear cost:* The cost increases linearly with the travel time: $c(\tau) = \tau$. Let $pdf_A(t)$ denote the probability density function of normal distribution A. Then, the expected cost is given by

$$\mathbb{E}[c(\tau_\pi(t_0))] = \int_{-\infty}^{\infty} pdf_{\tau_\pi(t_0)}(t) \cdot t\, dt = \bar{\tau}_\pi(t_0),$$

  which is the mean of the travel time of route π when travel is commenced at time t_0.
- *Exponential cost:* The cost increases exponentially with the travel time to more heavily penalize late arrivals: $c(\tau) = \exp(2w\tau)$ for some weight parameter w. The expected cost in this case is given by

$$\mathbb{E}[c(\tau_\pi(t_0))] = \int_{-\infty}^{\infty} pdf_{\tau_\pi(t_0)}(t) \exp(2wt)\, dt = \exp(\bar{\tau}_\pi(t_0) + w\tilde{\tau}_\pi(t_0)).$$

  The result of minimizing for $\mathbb{E}[c(\tau)]$ in equivalent to minimizing for $\log \mathbb{E}[c(\tau)]$. Following this, the cost then becomes $\bar{\tau}_\pi(t_0) + w\tilde{\tau}_\pi(t_0)$, and is hence a linear combination of the mean and the variance of the travel time (Lim et al. 2009).

Our approach works for either of these cost functions. In our implementation we used the exponential cost function, for it attempts to minimize both the mean and the variance of the travel time.

Planning Algorithm To find a path in the graph $\mathcal{G}$ between start node s and goal node g, we are confronted with a shortest path problem in a graph with time-varying and stochastic edge costs. In general, such problems are hard (Lim et al. 2009; Chabini and Lan 2010), but in our case we can exploit properties of the cost function that allow us to a standard A* algorithm, which we will slightly adapt to handle the stochastic travel times.

Firstly, both of the cost functions as defined above are *additive* given the way the mean and variance of the travel time are computed (see Equations (8) and (9)). That is,

$$\mathbb{E}[c(\tau_{\{e_1,\ldots,e_k\}}(t_0))] = \mathbb{E}[c(\tau_{\{e_1,\ldots,e_{k-1}\}}(t_0))] + x, \tag{10}$$

where x is a linear combination of the second terms of Equations (8) and (9). Second, traffic observes the so-called *first-in-first-out* property (Chabini and Lan 2010). This means that arriving earlier at a node u in the graph will never produce a costlier route than a route that arrives later at u. Note that this is not the case for graphs with general time-dependent edge costs.

These two properties allow us to use the standard A* algorithm, which is adapted to handle the stochastic travel times along a route. The algorithm is given in Fig. 3. Instead of maintaining a single cost value of each node u in the graph as in standard A*, we maintain both the mean $\bar{\tau}_u$ and variance $\tilde{\tau}_u$ of the travel time of the current-best route from s to u. Initially, these are infinity for all nodes u, except the start node

```
TRAFFICA*$(s, g, t_0)$
1: $\forall u \in V : \bar{\tau}_u \leftarrow \infty, \tilde{\tau}_u \leftarrow \infty; \bar{\tau}_s \leftarrow 0; \tilde{\tau}_s \leftarrow 0$
2: $OPEN \leftarrow \{s\}$
3: while $OPEN \neq \emptyset$ do
4:     $u \leftarrow \arg\max_{u \in OPEN}\{\bar{\tau}_u + \bar{h}(u) + w(\tilde{\tau}_u + \tilde{h}(u))\}$
5:     $OPEN \leftarrow OPEN \setminus \{u\}$
6:     if $u = g$ then
7:         return
8:     for each edge $e = (u, v)$ in $\mathcal{G}$ do
9:         $\mu \leftarrow \bar{\tau}_e(t_0 + \bar{\tau}_u)$
10:        $\sigma \leftarrow \tilde{\tau}_e(t_0 + \bar{\tau}_u)$
11:        if $\bar{\tau}_u + \mu + w(\tilde{\tau}_u + \sigma) < \bar{\tau}_v + w\tilde{\tau}_v$ then
12:            $\bar{\tau}_v = \bar{\tau}_u + \mu$
13:            $\tilde{\tau}_v = \tilde{\tau}_u + \sigma$
14:            pred$(v) \leftarrow u$
15:            $OPEN \leftarrow OPEN \cup \{v\}$
```

Figure 3: The modified A* algorithm to compute an optimal route with respect to the exponential cost metric between start node s and goal node g when traffic is commenced at time t_0. When planning has finished, the optimal route is inferred by following the backpointers from the goal g.

s, for which they are zero. The heuristic value $\bar{h}(u)$ provides a lower-bound estimate of the mean travel time to the goal g from a given node u, for which we use the Euclidean distance between u and g divided by the largest maximum speed in the road network. The heursitic value $\tilde{h}(u)$ provides a lower-bound estimate of the variance of the travel time between u and g, for which we use $\tilde{h}(u) = 0$. The functions $\bar{\tau}_e(t)$ and $\tilde{\tau}_e(t)$ which we refer to in lines 9 and 10 are given by Equations (2) and (3).

Maintaining Density Functions

Once a route has been planned for a car, we wish to take its presence into account when subsequent routes are planned for other cars. Based on the route that has been suggested, we can assume the car will follow it and add to the traffic densities at the road segments along its route at the times it is expected to traverse these road segments. At the same time, not all cars on the road use our self-aware planning system, and the system is not aware of the future plans of the cars that do use our system but have not entered the road network (yet). So, previously planned routes alone do not provide an accurate estimate of future traffic data.

Blending Historical and System Data In order to predict future traffic conditions, we let the density functions $\rho_e(t)$ used in the above algorithm be a combination of historical traffic density data $\rho_e^{\text{hist}}(t)$, and density data $\rho_e^{\text{syst}}(t)$ generated by route plans provided by our self-aware system. However, care needs to be taken that the historical data is partly *phased out* when actual data of planned routes is included in the densities, as to avoid cars being double counted. We proceed as follows. Let $\alpha \in [0, 1]$ be the proportion of cars that use our system to compute their routes. Further, let there be a function $\beta(\Delta t) \in [0, 1]$ that provides the proportion of cars that will be on the road at Δt time into the future which are already on the road currently. We assume $\beta(\Delta t)$ is time-independent, and can be inferred from historical traffic data on average travel times.

The traffic density $\rho_e(t)$ as used in our algorithm for a car starting travel at time t_0 is then computed as follows:

$$\rho_e(t) = (1 - \alpha\beta(t - t_0))\rho_e^{\text{hist}}(t) + \rho_e^{\text{syst}}(t). \tag{11}$$

This can be explained by the fact that a fraction $\alpha\beta(t - t_0)$ of all cars that will be on the road at time t have already been accounted for at time t_0 in the densities generated by our system.

Updating Traffic Densities When a route has been planned for a car by our algorithm, we wish to take its presence into account when subsequent routes are planned for other cars. To this end, we update the density data $\rho_e^{\text{syst}}(t)$ that only counts cars for which routes have been planned using our self-aware system. We update these traffic densities as follows.

The algorithm above will give us a route $\pi = (e_1, \ldots, e_n)$ and distributions $\tau_u \sim \mathcal{N}(\bar{\tau}_u, \tilde{\tau}_u)$ of the travel times from the start node s to each node u along path π. Let edge $e = (u, v)$ be part of π. The car for which a path is planned will be on $e = (u, v)$ at time t with probability:

$$q_{(u,v)}(t) = \int_{-\infty}^{t-t_0} pdf_{\tau_u}(t')\,dt' \cdot \int_{t-t_0}^{\infty} pdf_{\tau_v}(t')\,dt', \tag{12}$$

where t_0 is the time at which the car commences its route π.

The above equation computes the probability that the car both arrives at e before time t and leaves e after time t. The density on e at time t is defined by the number of cars on e at time t divided by the length ℓ_e of e. Hence, the distribution of the density $\rho_e^{\text{syst}}(t)$ at time t is updated as follows:

$$\bar{\rho}_e^{\text{syst}}(t) \leftarrow \bar{\rho}_e^{\text{syst}}(t) + q_e(t)/\ell_e \tag{13}$$

$$\tilde{\rho}_e^{\text{syst}}(t) \leftarrow \tilde{\rho}_e^{\text{syst}}(t) + q_e(t)(1 - q_e(t))/\ell_e^2, \tag{14}$$

which follows from treating the distribution $\rho_e(t)$ for each edge e and for each time t as an independent Poisson binomial distribution consisting of a number of cars each with a different probability of contributing to the density. When the number of cars gets large, the Poisson binomial distribution is well approximated by a normal distribution $\mathcal{N}(\bar{\rho}_e(t), \tilde{\rho}_e(t))$. This justifies the assumption in the planning algorithm of Fig. 3 that $\rho_e(t)$ is a normal distribution.

As the time axis is discrete, we only need to update a finite set of density distributions along the route planned for the car. We use the same discretization to compute the integrals in Equation (12). We use the updated mean and variances for the densities to route subsequent cars. This cycle of routing cars and updating densities continues indefinitely.

Empirical Results

In this section, we present our empirical study of the performance of our approach. Our hypothesis is that our system plans routes that have, on average, lower travel times than routes planned using the shortest path metric or stochastic-historical prediction method, increasingly so when the proportion of users of our system increases. This reinforces

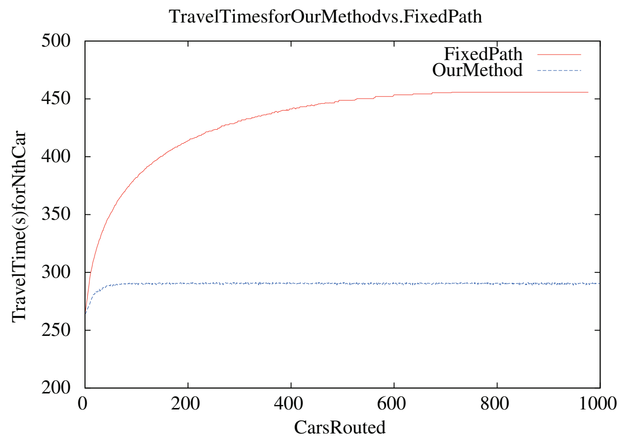

Figure 4: We highlight the performance of our algorithm compared with routing cars along a single path. The flow of cars quickly leads to congestion and long travel times for the single path. Our approach distributes the cars and settles to a constant travel time.

the essential claim of our paper: by taking into account the routes planned by the system itself, a planning system can find routes with substantially shorter travel time.

We have validated our approach by calculating plans for a fixed population of cars and queries using varying route planning methods. The validation was done in four parts. First, we compare our method with using a single path in a network. Next, we compare the performance of our self-aware algorithm to using shortest path A* searching. Next, we compare our algorithm to using stochastic-historical prediction. Finally, we investigate the performance of our algorithm as the percentage of total number of cars that are controlled varies.

Traffic Simulation

To simulate the travel times of the cars in these experiments, we use the same derivation as above. We calculate the estimated travel times using the fundamental diagram and the time-varying density data. This density data is then updated for every car that travels the network. For these experiments, we have added a *cutoff-capacity* to our road network edges. This is the maximum density value the edge will be assigned, regardless of how many cars are routed on it. This ensures that the planners we compare against do not plan routes with infinite time duration.

Avoiding Congestion on a Single Path

We designed this benchmark to showcase the basic premise of our approach. The road network is a rectangular grid of 5 $\times$ 5 intersections, connected by a road segments with equal maximum speeds, 22.35 m/s, capacities, 0.09 cars per meter, and cut-off capacities, 0.085 cars per meter. Each road segment is approximately 1000m. We assume the road network is initially empty (i.e. there is no traffic). Then, we begin routing a set S of cars, each defined by a tuple (s, g, t) of a starting vertex s, an ending vertex g, and a starting time

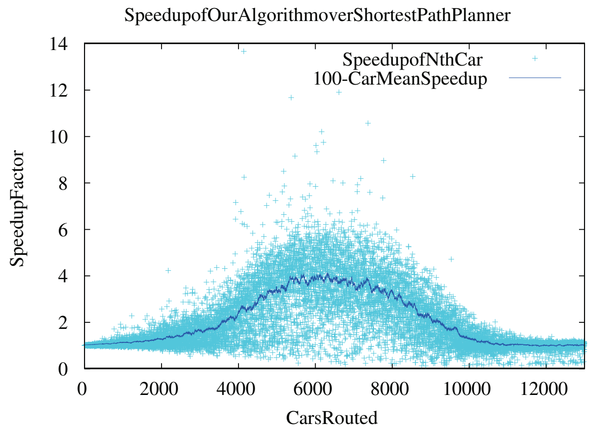

Figure 5: This figure shows the speedup factor that our method achieves over a shortest path planner for a series of cars. Each car has a random start and goal position.

t, enter the road network and traverse the route given by a route planner. We assign the starting vertex for each car to be the bottom left corner of the grid and the goal to be the top right corner. Each car is given a starting time $t = 2 * i$ seconds, where i is the car index in S.

We ran this experiment for both planners, the route planner that simply returns the shortest path between the start and goal vertex, independent of traffic conditions, and our self-aware system.

The result of this can be seen in Figure 4. Obviously, since all cars have the same start and goal vertices, they are all assigned the same route by the shortest-path planner, quickly causing growing congestion on this route. The incoming flow is enough to cause significant congestion, but not to cause a complete traffic jam. Our method distributes the cars along multiple paths to the goal based on densities predicted by earlier planned routes. By doing so, it can handle the flow of vehicles at a relatively constant travel time.

Comparison with Shortest Path Planner

In the second benchmark, we compare the behavior of our algorithm to the shortest path planner while routing a set of cars S with random initial and goal intersections. This scenario takes place on a grid road network with 15 $\times$ 15 intersections and a road length of 100m. The road network is initially empty. As above, we perform the experiment for both planners: first, we route each car in S using a shortest path planner and calculate the resulting travel times. Second, we do the same assuming all cars are routed using our self-aware route planner. To best illustrate the effect of network load in this scenario, each car has a starting time of zero. The parameters used for the experiment were capacity = 0.09 cars per meter, maximum velocity = 22.35 m/s, and cut-off capacity = 0.085. For storing the density information for each edge, a time discretization of 15s was used.

Figure 5 shows the result of this experiment. The result is given in terms of the speedup of our method over the shortest path planner, i.e. the ratio of the travel time planned by

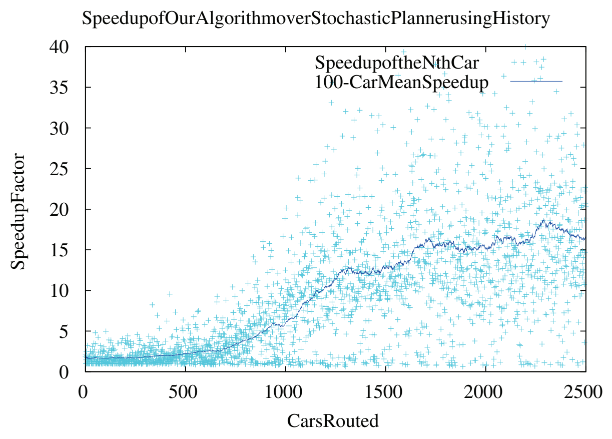

Figure 6: This figure shows the speedup factor (up to 20 for the 100-car mean) that our method achieves over a stochastic planner using only historical data for a series of cars, indexed from 0 to 2500. Each car has a random start and goal position.

the shortest path planner over the travel time planned by our method. The speedup for every car routed is displayed as the cyan scatter plot, and the average speedup factor for each cohort of 100 cars is displayed as the blue line. As can be seen, initially the speedup factor is negligible, since the road network has low traffic densities that do not significantly slow down traffic. Hence, in these cases the shortest path is indeed also the time-optimal path. However, as more an more cars have entered the road network, the average speedup in travel time by using our self-aware planner rather than a shortest path planner increases, peaking at a factor of approximately 4 after 6,000 cars have been planned. Eventually, the average speedup decreases again, since the road network has become so congested that alternative routes do not provide any benefit in terms of travel time.

Comparison with Stochastic Planner Using Historical Data

In this experiment, we compare the behavior of our algorithm to a stochastic planner using historical data (SPUHD) when routing a population of cars S. This experiment takes place on the same grid road network as above, with 15×15 intersections. We generate *historic* traffic data by defining a set of cars S with random start and goal intersections, routing each car in S using a shortest path planner, and calculating the resulting densities. The cars are created in 40 batches of 200, with each batch having a starting time of 5 seconds later than the preceding batch. This creates a maximum average network density, using the shortest path routing, of 0.054 cars per meter, and areas of full congestion, i.e. 0.085 cars per meter. These time-varying, edge specific stochastic densities are the *history* for the scenario. We assume that S represents the typical traffic flow. We compare our method and the SPUHD by planning for all of S, given the calculated stochastic *history*. In our self-aware routing algorithm, the planned routes are used to update the traffic densities for plans of future cars, but our planner is unaware of the *history*. In the SPUHD, each car in S is planned for assuming the *history* is a valid prediction, and these predicted densities are not updated based on the SPUHD's planned routes.

The results are shown in Figure 6. We can see that only using the stochastic *history* is not an effective strategy when all cars are being navigated by the planner. This is an extreme example, but it illustrates a basic motivating problem with using stochastic prediction. If the planner were routing one car in S, then the *history* would be almost perfect; as the planner is routing all of S, the predictions based on the *history* are not valid. As the SPUHD *believes* a certain congestion pattern will occur, due to the *history*, it routes cars around that congestion pattern. However, as the SPUHD is controlling all the cars, the predicted congestion pattern does not occur, but instead the planner creates congestion in other areas. The SPUHD assigns cars sub-optimal routes due to its belief that the typical, historical traffic flows will remain constant. Our method, on the other hand, distributes car routes and achieves a 100-car mean speedup of up to a factor of 20.

The high speedup factors here illustrate how unsuitable pure historical prediction is when the entire set of cars is being routed. In attempting to avoid predicted densities, the stochastic planner using historical data irrationally prefers domains of the road network, which then causes congestion and traffic jams in those areas. Even at low network densities, approximately 0.014 cars per meter, the SPUHD causes traffic jams, with some roads being saturated to their cutoff capacity, 0.085 cars per meter.

Effect of Adoption Rate

In our final benchmark, we analyze the effect of the adoption rate of our system in scenarios in which part of the cars use our self-aware system, and the other part of the cars use a shortest path planner. We let a set S of cars with random start and goal intersections enter the (initially empty) road network at a rate such that over time heavy congestion is likely to be created on the road network. We use the same road network as above. A percentage α of the cars (randomly sampled from S) use our self-aware traffic route planner to plan their routes, whereas $1-\alpha$ of the cars use a shortest path planner. For the sake of the simplicity, our self-aware route planner ignores the portion of cars it does not control; in reality this data can be estimated from historical data (see Section **Blending Historical and System Data**). We repeat this experiment for various values of α. The proportion of cars, α, that are routed by our method are chosen using a consistent random number seed: this implies that a car routed for a low α will also be routed for a high α, preserving features of the graphs for each value of α. The cars enter the road network at a rate of 50 per second.

Figure 7 shows the results, a graph depicting the 100-car mean speedup for various adoption values from 50% to 100%. We see a peek speedup of over 10 when 100% of the cars are controlled: our system avoids the creation of heavy congestion and large scale traffic jams. However, the maxi-

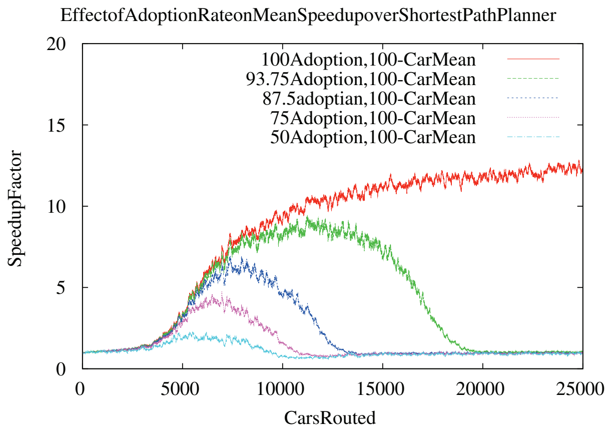

Figure 7: The 100-car mean speedup of our method over the simple planner for varying adoption percentages.

mum speedup and the integral of the speedup curve decrease rapidly with α, showing a strong sensitivity to uncontrolled cars creating traffic jams. At the 50%, a maximum speedup of 2 is observed, and for smaller α values, a similarly small speedup is observed. These results clearly show that the benefit of our system increases with the adoption rate.

Conclusion and Future Work

State of the art approaches can handle stochastic planning and can make use of traffic predictions, but they ignore a useful source of information, i.e. the previously planned routes. As routing systems become more pervasive, the routes they plan will begin to significantly influence the future state of traffic. Prior plans then become relevant to future plans. Our approach addresses this issue. We provide a method to update stochastic traffic predictions with previously planned routes. Given stochastic predictions of future traffic states, we plan for cars within this space. For each path planned, we update the stochastic predictions based on the routes planned for each car. The density of each edge is updated according to the estimated arrival and departure times for the car. The velocity of each edge is then updated according to the fundamental diagram, an empirical relationship between density and velocity. In our simulations, the improved routing algorithm results in better utilization of the road network, reduces congestion and the travel time for each car.

There are many avenues for future work. We would like to perform more analysis and validate the performance of our algorithm on actual traffic data. It would be useful to relax some of our assumptions in terms of normal distributions along each edge of the road network. Finally, it may be useful to develop a decentralized version of our self-aware traffic planning algorithm.

Acknowledgments

This work was supported in part by ARO Contract W911NF-10-1-0506, NSF awards 0917040, 0904990 and 1000579, and RDECOM Contract WR91CRB-08-C-0137.

References

Brakatsoulas, S.; Pfoser, D.; Salas, R.; and Wenk, C. 2005. On map-matching vehicle tracking data. In *Proceedings of the 31st international conference on Very large data bases*, 853–864. VLDB Endowment.

Chabini, I., and Lan, S. 2010. Adaptations of the A* Algorithm for the Computation of Fastest Paths in Deterministic Discrete-Time Dynamic Networks. *IEEE Transactions on Intelligent Transportation Systems* 3(1):60–74.

Florian, M.; Mahut, M.; and Tremblay, N. 2008. Application of a simulation-based dynamic traffic assignment model. *European Journal of Operational Research* 189(3):1381–1392.

Greenshields, B., et al. 1935. A study of traffic capacity. In *Highway Research Board Proceedings*, volume 14, 448–477.

Horvitz, E.; Apacible, J.; Sarin, R.; and Liao, L. 2005. Prediction, Expectation, and Surprise: Methods, Designs, and Study of a Deployed Traffic Forecasting Service. *Conf. on Uncertainty in Artificial Intelligence*.

Hua, M., and Pei, J. 2010. Probabilistic path queries in road networks: traffic uncertainty aware path selection. In *Proceedings of the 13th International Conference on Extending Database Technology*, 347–358. ACM.

Lim, S.; Balakrishnan, H.; Gifford, D.; Madden, S.; and Rus, D. 2009. Stochastic Motion Planning and Applications to Traffic. *Algorithmic Foundation of Robotics VIII* 483–500.

Min, W.; Wynter, L.; and Amemiya, Y. 2007. Road traffic prediction with spatio-temporal correlations. In *Proceedings of the Sixth Triennial Symposium on Transportation Analysis, Phuket Island, Thailand (June 2007)*.

Nikolova, E.; Kelner, J.; Brand, M.; and Mitzenmacher, M. 2006. Stochastic shortest paths via quasi-convex maximization. *Algorithms–ESA 2006* 552–563.

Nikolova, E.; Brand, M.; and Karger, D. 2006. Optimal route planning under uncertainty. In *Proceedings of International Conference on Automated Planning and Scheduling*.

Nikolova, E. 2010. High-Performance Heuristics for Optimization in Stochastic Traffic Engineering Problems. *Large-Scale Scientific Computing* 352–360.

Peeta, S., and Ziliaskopoulos, A. 2001. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics* 1(3):233–265.

Siebel, F., and Mauser, W. 2005. On the fundamental diagram of traffic flow. *Arxiv preprint cond-mat/0503290*.

Work, D.; Blandin, S.; Tossavainen, O.; Piccoli, B.; and Bayen, A. 2010. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress*.