Participatory Route Planning

David Wilkie Cenk Baykal Ming C. Lin

Department of Computer Science
University of North Carolina at Chapel Hill
{wilkie, baykal, lin}@cs.unc.edu
http://gamma.cs.unc.edu/PRP

ABSTRACT

We present an approach to "participatory route planning," a novel concept that takes advantage of mobile devices, such as cellular phones or embedded systems in cars, to form an interactive, participatory network of vehicles that plan their travel routes based on the current traffic conditions *and* existing routes planned by the network of participants, thereby making more informed travel decision for each participating user. The premise of this approach is that a route, or plan, for a vehicle is also a prediction of where the car will travel. If routes are created for a sizable percentage of the total vehicle population, an estimate for the overall traffic pattern is attainable. Taking planned routes into account as predictions allows the entire traffic route planning system to better distribute vehicles and minimize traffic congestion. We present an approach that is suitable for realistic, city-scale scenarios, a prototype system to demonstrate feasibility, and experiments using a state-of-the-art microscopic traffic simulator.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; I.6.5 [**Simulation And Modeling**]: Model Development; I.2.9 [**Robotics**]: Autonomous vehicles

Keywords

Participatory Sensing, Adaptive Routing

1. INTRODUCTION

Traffic congestion management is a global challenge that has increasingly important economic, societal, and environmental impacts. It is unlikely that traditional physically-centered mitigation strategies by themselves will be successful or sustainable. Numerous strategies have been proposed to construct Intelligent Transportation Systems (ITS) by incorporating sensing, information, and communication technologies in transportation infrastructure and vehicles. Through networks of sensors, recent cutting-edge efforts can provide real-time traffic monitoring for subsets of the road network, but they have not yet offered system-level relief to the traffic congestion problem.

SIGSPATIAL'14, November 04 - 07 2014, Dallas/Fort Worth, TX, USA

http://dx.doi.org/10.1145/2666310.2666406

When planning a route, state-of-the-art planners use traffic predictions derived largely from recent 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 fixed frequencies [3], which can then be used by vehicles to plan around congested areas. However, live data alone does not enable predicting future traffic conditions. For example, if a route is planned in which a car arrives at a certain road in 30 minutes, the current conditions may not be an accurate estimate for the conditions of that road 30 minutes later.

This problem can be addressed by using a prediction scheme for the future traffic conditions based on the current and historical probabilistic data of traffic conditions at similar times of the day under similar weather conditions [10], [19], [18]. However, these predictions are not valid if a large portion of the vehicles take the historical knowledge into account. For example, if there exists a centralized route planner controlling *every* car in the system, historically congested areas would be unduly avoided, causing congestion to appear in new areas: the historically predicted pattern would not arise.

Extending and exploiting the idea of participatory sensing [4], we propose the novel concept of *participatory route planning*, which uses the routes of vehicles in the current networked system to coordinate with each other by sharing their planned routes with a central router via mobile communication. Our adaptive approach accounts for the fact that when a route is planned for a car, that car will then cause a small increase to the traffic density on the roads it later traverses. We can thus use the predicted paths from the route planner itself as an information source, *in addition to* historical data and/or current traffic condition. This allows our participatory route planning system to plan routes more effectively for a large percentage of the cars as the impact of their routing choices on future traffic conditions is taken into account. Every car that queries the route planner can then use this information to plan a route for itself and implicitly coordinates the route with the previously routed vehicles. Its planned route is then used to update the estimate of future traffic conditions for other vehicles participating in this system.

We present large-scale experiments using a state-of-the-art microscopic traffic simulator to demonstrate the functionality of our system in scenarios ranging from simple convoys of a couple hundred cars to city-scale rush hour scenarios with thousands of cars. Our experimental results suggest that such participatory route planning can better coordinate and distribute vehicles, resulting in an overall

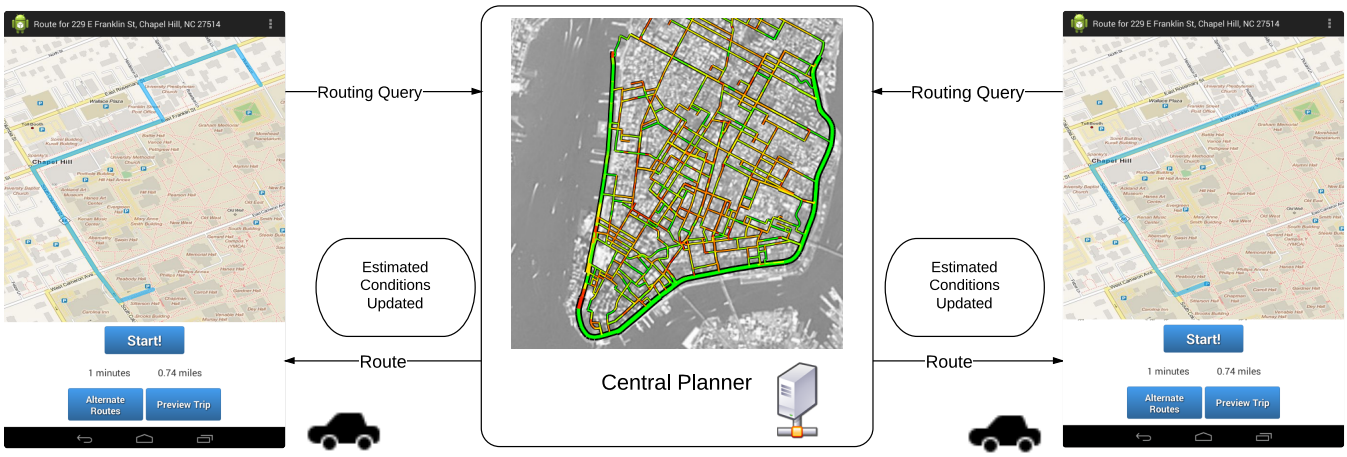

Figure 1: **A System Architecture of Participatory Route Planning:** The mobile clients send route queries to the central planner, which takes the updated traffic and routing information from participants and live traffic sensor information to plan a new route for each participating client in the network.

reduction in travel time. Further, we provide a prototype system demonstrating how a system such as ours could be feasibly implemented for city-scale demand.

Main Result. The key contributions of this paper include:

- The concept of *participatory route planning*, in which users query a route planner via mobile communication with their current location and goal, and the route planning system implicitly coordinates traffic flow for the users;
- An extension of "Self-Aware Traffic Route Planning" [27] to account for real world traffic dynamics and road-network intersections;
- A prototype demonstration system that can handle thousands of queries and illustrates how features such as destination prediction can be incorporated; and
- Multiple validation experiments using microscopic traffic simulation, including an experiment derived from real-world census data.

The rest of the paper is organized as follows. In Section 2, we discuss related work to our approach. In Section 3, we give an overview of our prototype mobile system. Section 4 presents an overview of the mathematical framework we build upon and our new algorithms for participatory route planning. Finally, we discuss the implementation, experimentation and validation of this novel approach in Section 5. We conclude by discussing future research directions.

2. PRIOR WORK

Our system builds off of the theoretical framework proposed by [27], which is discussed in detail in Section 4.1, "Mathematical Framework". This paper presents a routing approach that incorporates updating time-varying, Gaussian density distributions for uncertain routes. Their route planning algorithm is primarily inspired by [17], which presents a planning algorithm using graphs with stochastic, time-invariant edge costs.

A related body of work includes [7], which proposed a multi-agent reservation-based system to replace existing signalized intersections. Our route planning algorithm also takes into account the effect of other users in the system, but, in contrast, our algorithm uses a global, macroscopic approach to alleviating overall traffic congestion, rather than local considerations of intersection throughput. To create the system proposed by [7] in the real world would also require highly accurate sensors at every intersection in the system, whereas our system requires no additional sensing infrastructure.

Another body of related work is the study of Dynamic Traffic Assignment, the problem of distributing flows of traffic from known origins and destinations. A summary of approaches can be found in [20]. Most relevant to our work are the solutions involving simulation, such as [8]. In these approaches, vehicles are iteratively routed and simulated until an equilibrium is reached in which no further reduction in travel time can be obtained.

We have created a mobile system that features our participatory route planning algorithm. Our mobile system is similar to CarTel ([11]), a mobile sensor computing system that utilizes connectivity to the internet and GPS to collect and process data from sensors located on mobile systems. In this respect, our work also resembles that of the Mobile Millennium project, which attempts to use GPS in cellular phones to gather traffic information in real time ([12], [1]). However, in contrast to these projects, which sense live traffic data, our system not only considers current traffic conditions, but also utilizes the plans of the users to coordinate those users by predicting their effect on the road conditions.

3. PROTOTYPE SYSTEM

Participatory routing has performance requirements distinct from typical routing systems due to the heavy volume of queries that are expected. We have therefore created a prototype system to show that such a system can be created, and that it can perform in real time.

Our system implements a client and server architecture (Figure 2). For our client implementation, we have created a mobile application that is capable of asynchronously sending routing queries to

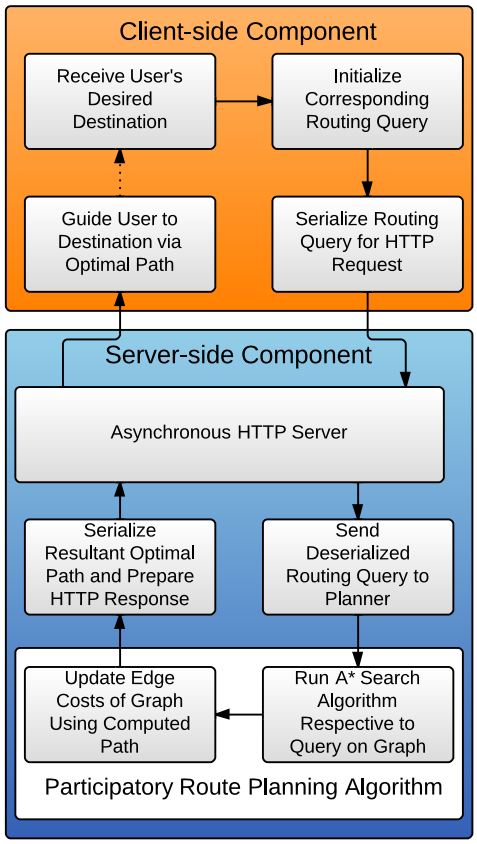

Figure 2: **System overview of our client- and server-side model:** The client-side component (with the orange background) illustrates our mobile app system; the routing query is received from the client, serialized, and sent to the server, which handles the request. The server-side component (shaded in blue background) shows the handling of the routing query and our participatory route planning process. The routing algorithm is ran with respect to the destination request, the stochastic road map is updated, and the optimal route in term of travel time is sent back to the client.

the server and obtaining routes to the desired location. Once a user has chosen a destination on the client, an asynchronous routing query is sent to the server with the most recent information regarding the user's latitude, longitude, and orientation.

In order to handle requests from numerous clients, our server employs an asynchronous and multi-threaded approach. One thread is reserved for routing and update computations, as each route is technically dependent on the previous routes' updates. The remaining threads take queries from clients, process them, and return computed routes, which consist of a list of road segments along with their geometries.

Our client system's interface, which can be seen in Figure 3, is based on common mobile and GPS routing programs on both Android devices and iPhones. However, a participatory system has some distinct requirements. As mentioned before, this system is envisioned to involve a large portion of the traveling vehicles: a typical use case would be someone commuting to work in the morning. As such, the interface for such a system needs to be as unobtrusive and intuitive as possible. One approach to achieving this design goal is having the client work automatically, or as automatically as possible. To this end, our client application features *destination prediction* for the user based on their routing history. This can be seen in 3(a) where an address is suggested on the main screen. The user then needs only to confirm. One could also imagine a completely automatic client built into a car's GPS system.

Our destination prediction system is based on [2] and [16]. In both of these approaches, the authors used streams of GPS data to construct Markov models, which were then used to infer the user's daily movements by estimating the transition probabilities between locations. Since our client program does not run at all times, GPS data is not present continually. Therefore, we adapt the proposed methods to use information from users' past routing queries to forecast their destinations: at the time a query is requested, the client checks to see what the most likely destination is using the user's current location and the frequency of previous destinations from the current location. This model could also be extended to take the time of departure into account or aggregate multiple users' data to identify common trips.

4. ALGORITHMS

In this section, we present the key mathematical foundation and the new algorithms for participatory route planning.

4.1 Mathematical Framework

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: **end if**
9: **for each** edge $e = (u, v)$ in $\mathcal{G}$ **do**
10: $\mu \leftarrow \bar{\tau}_e(t_0 + \bar{\tau}_u)$
11: $\sigma \leftarrow \tilde{\tau}_e(t_0 + \bar{\tau}_u)$
12: **if** $\bar{\tau}_u + \mu + w(\tilde{\tau}_u + \sigma) < \bar{\tau}_v + w\tilde{\tau}_v$ **then**
13: $\bar{\tau}_v = \bar{\tau}_u + \mu$
14: $\tilde{\tau}_v = \tilde{\tau}_u + \sigma$
15: pred$(v) \leftarrow u$
16: $OPEN \leftarrow OPEN \cup \{v\}$
17: **end if**
18: **end for**
19: **end while**

Figure 4: The A* algorithm from [27] to compute an optimal route with respect to the cost metric between start node s and goal node g at time t_0. τ values are travel time estimates/costs, and h is the heuristic function.

Our system is inspired by the mathematical framework introduced in [27]. This work suggests a theoretical approach to routing vehicles that considers the system's own planned routes as a new source of information. The method routes cars individually, and then uses the planned routes as estimates for the cars' trajectories in the near future. These trajectories, in aggregate, form an estimate of the future traffic pattern. This method is composed of (a) a *route planner* that computes paths for cars through a time-dependent density field defined on the road network and (b) an *updater* that modifies the density field according to the calculated route.

The *route planner* makes use of a stochastic A*-search algorithm through a time-dependent density field. This field is composed of

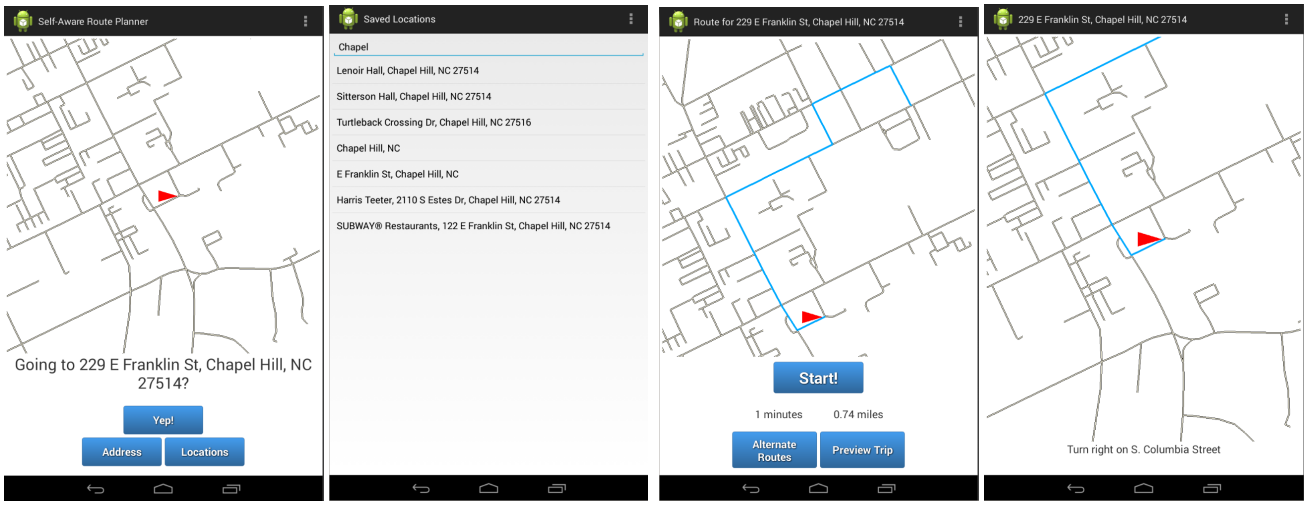

Figure 3: **Different screens featured by our Android App** (from left to right): (a) the Main Screen, hosting the interactive map display and destination prediction output; (b) the Locations Screen, showing previous destination queries; (c) the Route Confirmation Screen, detailing the route to destination; (d) the En Route to Destination Screen, portraying turn-by-turn directions for the user to follow and the interactive map display.

Gaussian distributions, discrete in time and space, defined over the road network graph, $\mathcal{G}$. The algorithm is shown in Figure 4. For each road explored, the cost of traversing the road is the estimated travel time:

$$\tau_e(t) = \frac{\ell_e}{f_e(\rho_e(t))},$$

which is the length of the road divided by the estimated velocity, $f_e(\rho_e(t))$, which is a function of the estimated density. The function uses values for the maximum density and maximum velocity to determine the current velocity, which can be estimated using a number of models, as discussed in [9, 28]. In our work, we use a version of the function for the equilibrium velocity presented in the *Aw-Rascle-Zhang* traffic model,

$$f_e(\rho_e(t)) = (v_{max} - v_{min}) * \left(1 - \frac{\rho_e(t)}{\rho_{max}}\right)^{\gamma} + v_{min}$$

Once a route has been planned, that route is considered an estimate for where that car will go in the future. For each car routed, the method adds a marginal amount of density to the road network along the planned path. For each edge of the route, the travel time estimates are used to calculate the probability that the car is on that edge at a certain time,

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

where $q_{(u,v)}(t)$ is the probability that a particular car is on edge $e = (u, v)$ at time t. This car is then added to the density field for that road segment, taking into account the length of the road:

$$\rho_e(t) = \rho_e(t) + q_{e=(u,v)}(t)/\ell_e.$$

This approach, though promising, has several shortcomings. First, the method ignores traffic flow dynamics that can lead to the spread of traffic congestion and jams. Second, the method ignores the effect of traffic intersections, which can be a significant cause of congestion and delay in urban scenarios. Third, there is not sufficient validation: the preliminary results were given for cars moving through a simple 5x5 grid without realistic intersections, and the simulator used for experimentation is a highly abstract method. Finally, the method does not make use of the "participatory sensing" framework to update routes and conditions.

We introduce a novel concept of "participatory route planning" that exploits both "participatory sensing" [4], for continuous real-time traffic updates, and "self-aware route planning" [27], on the premise that the planned routes become an integral part of realistic traffic prediction. Furthermore, we also introduce stochastic models to handle complex urban scenes as encountered in real-world scenarios. As discussed in Section 5, "Experiments", we further undertake a rigorous investigation on the performance of this new *participatory route planning* algorithm with a commonly-used, state-of-the-art microscopic traffic simulator.

4.2 Participatory Route Planning

"Participatory route planning" synthesizes 'participatory sensing' to refine and update the traffic patterns and self-aware route planning [27] to handle real-world urban conditions and to improve the predictive capability. Compared to prior work, this algorithm can handle traffic flow dynamics and takes into account delays caused by signalized intersections.

4.2.1 Traffic Flow Dynamics

In the original formulation, the travel time estimate for a road is calculated as

$$\tau_{travel} = \frac{\ell}{f(\rho(t))},$$

where $\rho(t)$ is the time-dependent density, ℓ is the road length, and $f(x)$ is a function that maps density to velocity using a fundamental diagram. However, this ignores the dynamics of real world traffic. If a downstream road is heavily congested, that congestion limits the amount of flow that can enter the road, propagating the congestion upstream.

To account for this, we add a model of the flow between roads. Given two connected roads, at some time t, we know their densities, velocities, and speed limits, which are all considered constant

during the time interval. To find the flow between these roads, then, we can use a Riemann solver, as is done in continuum and hybrid traffic simulation [23, 22]. This flow thus limits how many cars can enter the downstream road during the time interval.

However, we do not have a continuum traffic simulator: we need to calculate travel times for individual vehicles. To accomplish this, we build a *queuing model* over our continuum traffic representation using the above flow calculation. For this, we first calculate the expected number of cars on the road as $c = \rho(t) * \ell$, where $\rho(t)$ is the time-dependent density, and l is the road length. We then calculate the *outflow*, $o(t)$, or the number of cars per time interval that can depart, using a Riemann solver for the *Aw-Rascle-Zhang (ARZ)* system [23], as our routing system models traffic in the same manner as ARZ[1]. For a simple connection of two roads, we can now calculate how long it takes for the current cars to exit the road, and thus how much time it would take a new car to traverse the road.

However, at most intersections there are multiple downstream roads, and it is not generally known to which the cars currently on the road will go. This means we do not know what road or roads to use to calculate the outflow. The current cars, represented by the density value, could have been previously routed, sensed, or estimated from historical conditions. To address this, we make a *worst-case* assumption that the preceding cars on the road will choose the downstream road with lowest outflow. Alternatively, the model could choose a best-case outflow or an average outflow. However, we have observed that the worst-case assumption works best in practice.

The time required for the existing cars to leave the road is $\tau_{drain} = \frac{c}{o}$. This cost is only taken into consideration if it is greater than the previously estimated travel time,

$$\tau_{est} = max(\tau_{travel}, \tau_{drain})$$

as a car either freely traverses a road and departs or it reaches the end of an existing queue and waits for it to empty.

4.2.2 Intersection Delay

In the earlier work on traffic-predictive routing, intersections were only considered for their topological properties: they only created the connections of the network. In real world scenarios, especially urban scenarios, intersections cause delays that need to be modeled.

Above, we calculated a delay time due to the need for a queue of traffic to empty. However, that model assumes an uninterrupted flow out of the road. If the road has a signalized intersection, this flow will be broken by the traffic signal's cycles. Therefore, for signalized intersections, we modify the τ_{drain} equation to incorporate the cycles of the traffic signal. The cars can only leave the road during the green light portion of the cycle, which is τ_g seconds long. Due to reaction time and acceleration constraints, only a portion of this will be usable, τ_g^* [6]. To empty the queue of cars will require a number of cycles,

$$c_g = \frac{\tau_{drain}}{\tau_g *}$$

where c_g is the number of effective green lights needed for the outflow to finish.

Each of these cycles takes $\tau_g * i = \tau_f$ seconds, where i is the number of independent green cycles. The final time estimate then is $\tau_{light} = c_g * \tau_f$, and thus the overall model is

$$\tau_{final} = \max(\tau_{est}, \tau_{light}) + \tau_c$$

where τ_c is a constant cost for the intersection regardless of the current queue length, $\tau_c = \tau_f(1-\frac{1}{i})\frac{1}{2}$, representing the probability of a red light when no other cars are present.

4.3 Analysis

For each car, an A* algorithm computes the path to the goal. As the error between our heuristic and the actual distance, $h * (v)$, is not bound by $O(\log h * (v))$, the time complexity of the search will be exponential in the path length, like most A* computations [21]. We use a simple shortest-distance heuristic, and it is likely a more sophisticated heuristic could significantly prune the number of edges that get explored.

Each route also causes the traffic density representation to be updated. For each of the n edges of the route, the density values for d time steps need to be updated. The number of timesteps depends on the probability threshold used, i.e. when the marginal addition of density is ignored, the uncertainty of the car's arrival and departure times, and on the size of the time discretization, Δt. The cost of the update is then $O(nd)$. To reduce the memory requirement of the density field, a sparse vector representation can be used.

5. EXPERIMENTS

As a proof-of-concept, we conduct city-scale experiments to demonstrate the functionality of our system. These experiments start by illustrating simple behaviors and then build up to more complex scenarios.

To perform these experiments, we use the Simulation of Urban Mobility (SUMO) simulator [13], a state-of-the-art microscopic traffic simulator. Our experiments consist of comparing the performance of our participatory routing system against baseline routing systems in various scenarios. For each experiment, we create a population of vehicles, each with an origin, a destination, and a departure time. The cars are populated at a specified rate using a Poisson instantiation process, creating a relatively uniform distribution of cars over time. We create routes for these vehicles using both our method and baseline methods. These sets of routes are independently simulated, and we compare the resulting statistics, particularly the travel time, for each vehicle. The road networks used for these experiments are from Open Street Maps. The speed limit parameter for each road segment was assigned by SUMO, the maximum density for each road was calculated from the road and car lengths, and the average length of the green-light cycles for intersections are assigned by SUMO. We consider the exact cycle timings for the intersections to be unknown: i.e. for a road at a given time, we do not know if the light is green or red.

The remainder of this section is organized as follows. In Section 5.1, we compare our approach with "Self-Aware Traffic Routing" [27]. In Section 5.2, we discuss the baseline approaches we use for comparison. In Section 5.3, we describe the simulation scenarios. In Section 5.4, we discuss the results of simulating those scenarios. In Section 5.5, we discuss the system's performance. And finally in Section 5.6, we provide some discussion and analysis of the results.

5.1 Comparison with Self-Aware Routing

[1]Though our system has zero relative flow.

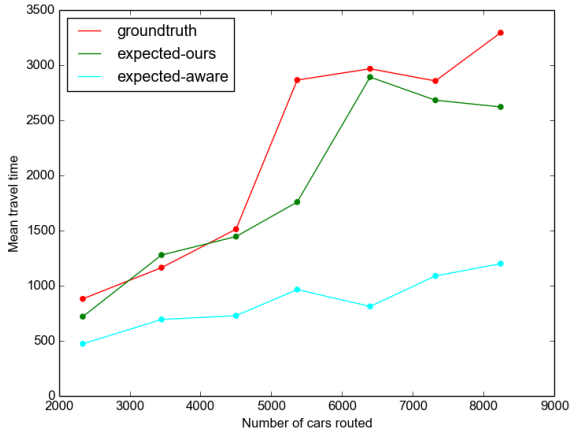

Figure 5: The mean travel time for the simulated ground truth (red), our method (green), and the self-aware routing algorithm (blue) for the MSMD scenario, described below. Our method can closely track the ground-truth travel time, while the self-aware algorithm systematically underestimates the actual travel time due to its inability to account for intersection delay.

Our system builds on the theoretical framework of "Self-Aware Traffic Routing", [27]. In this work, a stochastic graph algorithm enables paths to be aggregated statistically, allowing future searches to take into account their effects on future traffic. However, this work was not experimentally validated; it only considered a highly abstract 5x5 grid-based graph with no lanes, no traffic lights, no highways, etc.; it ignored traffic jam formation; and it did not address any real-world considerations of creating such a system. In this paper, we consider and address all of these complex issues observed in real-world traffic for common urban scenes.

One of the most significant advantages of our approach is that we can more accurately predict the travel time of cars than [27] (as shown in Figure 5), and thus we can use the planned routes themselves to better predict future traffic patterns. This capability enables our method to be used in conjunction with real-time sensing and planning. An example of this can be seen in Figure 10, in which our method can predict patterns of congestion, while the Self-Aware framework predicts fast travel and little to no congestion.

5.2 Baselines

Our simulations compare our method with the following baseline systems:

5.2.1 Shortest-Path Router (SP)

The simplest baseline system we use is a shortest-path router. The router we use for this is *Duarouter*, part of the SUMO package. We provide this router with the start and goal roads for each query, and it returns a route that we then provide to the simulator.

5.2.2 Sensor-Data Router (SD)

This baseline system models existing, state-of-the-art commercial systems. These systems can receive sensor readings from highway loop-detectors, mobile devices, and other sources to create velocity estimates for the road network. These velocity estimates can then be used to find the fastest path to the goal. Similarly, this baseline router receives the mean velocity for every road in the network every 60 seconds, which represents the aggregation period used with real sensors. For each vehicle, this baseline plans the fastest route

Figure 6: **An example of the *MSSD* scenario:** in which vehicles are spawned at various origin points and drive to a single destination. On the left is the mean velocity field of the fastest-path baseline router, and on the right is the mean velocity field resulting from our method. The destination point was chosen randomly inside the destination region, designated by the red rectangle. The origin points were chosen randomly outside this region with the additional constraint that they be at least 1km away from the destination. The mean velocity is shown as a color ranging from red, 0 m/s, to green, 13 m/s, the speed limit for most of the roads.

given the current velocities and road geometries. To implement this, we simulate all cars up to the end of the current time window, t_i, and then export the mean velocities of that time window, $\Delta t = t_i - t_{i-1}$, back to the router. We then route vehicles that depart between t_i and t_{i+1} using the reported velocities, simulate all cars up to t_{i+1}, and so on.

It should be noted that real world systems would have noisy sensor measurements and would likely only have measurements on highways and major roads. In these regards, this **SD** baseline system is more powerful than existing navigation systems and routing alternatives.

5.3 Scenarios

In this section, we discuss our experimental set-up for various scenarios. These scenarios use two road networks, one of the lower portion of Manhattan and one of the city of Sioux Falls. In both of these scenarios, we simulate a population of cars that are assigned routes to follow and record the simulated travel times. The travel times for our method and the baselines are later compared.

5.3.1 Manhattan

These experiments show the behaviors and characteristics of the routing system in a realistic urban setting. They feature several traffic flows intended to illustrate the performance of the system in different scenarios. The map used for these experiments is of lower Manhattan from Open Street Maps. It is composed of 4,073 edges and 3,135 vertices. The total road length is 392,397 meters. A discretization of 5 seconds was used for the time domain.

Single-Source, Single-Destination (SSSD). The simplest routing test is between a single origin and destination (OD). Car queries were generated between a randomly chosen origin and a randomly chosen destination over a 20 minute period at a fixed rate using Poisson instantiation.

Multiple Source, Single Destination (MSSD). This scenarios are modeled on a typical commuting pattern, in which vehicles from a surrounding area travel to a downtown area. The downtown area is defined by a bounding box, seen in Figure 6. The destination must be within this bounding box, while the origins must be outside and

at least 1000m away. For the scenario, there are 5 origin locations generated and one destination. Traffic queries were generated using Poisson instantiation over a 20 minute period.

Multiple Sources, Multiple Destinations (MSMD). This scenario simulates a larger, more realistic traffic flow throughout a city. The traffic is generated by combining 10 **MSSD** scenarios, each of which has 5 origins and 1 destination. We again use a bounding box to restrict the destinations. This scenario is motivated by traffic patterns during the morning rush hour, when residents leave their homes and commute to workplaces. Traffic queries were generated using Poisson instantiation over a 20 minute period.

Using real-time sensing with our system. In this experiment, we demonstrate how real-time sensor data could be combined with our system. Currently, sensor data is only available on a limited portion of the road network, primarily on highways where loop detectors are present. This data is also not typically available in real time. As our system assumes a large number of vehicles are participating, the vehicles themselves can report conditions in real time by providing velocity and position updates. These updates can be filtered to create an estimate of the velocity for each road, as was done in [28].

To simulate this, we iteratively route batches of vehicles. Each batch consists of vehicles spawned during a time window of 60 seconds. For each batch, the mean velocity of each road in the simulation during the preceding batch is used as an input by our participatory routing system. This is done in a direct way by setting the speed-limit, v_{max}, for each road in our system to the corresponding simulated velocity. Historical data could be incorporated into our system in a similar manner.

5.3.2 Sioux Falls

These experiments are intended to show the behavior of the system in as realistic a scenario as we can create. For these experiments, we use a set of trips, created by [5], that match real-world conditions as defined by U.S. census data for the city of Sioux Falls. The origins and destinations for the morning rush hour portion of these trips can be seen in Figure 7. Versions of this scenario have been used by many traffic engineering studies, including [15, 14, 24]. One difference between our experiments and previous work is that we use a realistic street map from Open Street Maps data, whereas prior studies used a more abstract road network graph, representing only the major arterial roads and highways. This map is composed of 28,608 edges and 26,352 vertices. The total length of roads is 5,232,662 meters. A discretization of 10 seconds was used for the time domain.

In this experiment, we compare our routing approach against two baselines: (a) the basic shortest-path router and (b) the sensor-data router. The set of vehicle queries used is a portion of the Sioux Falls morning rush hour traffic, from 7 am to 9 am, and limited to specified numbers of cars in order to show the system performance at different load levels. In Figure 9, we can see the mean travel time of our method and of the two baselines for various numbers of vehicles. Each data point represents a separate experiment with the specified number of total vehicles routed.

5.4 Results

This section details and analyzes the results of the experiments described above. First the results of the Manhattan experiments are discussed, followed by the results of the Sioux Falls experiment.

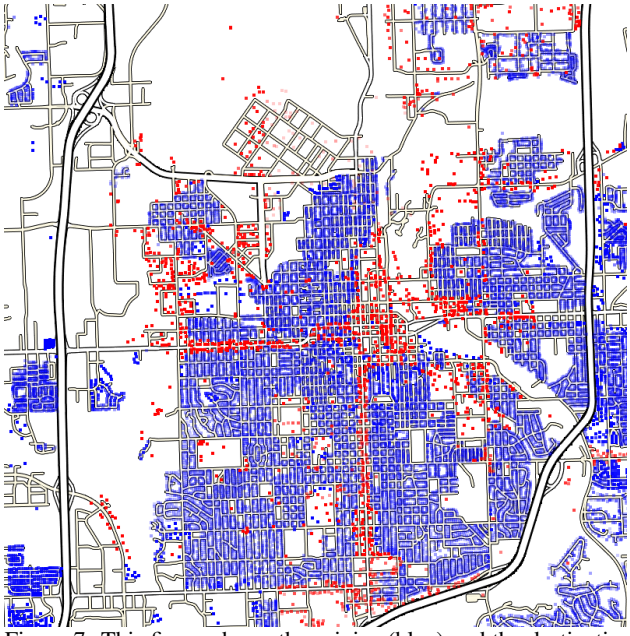

Figure 7: This figure shows the origins (blue) and the destinations (red) for the Sioux Falls morning rush hour commute.

SP Baseline	Speedup mean	std. dev.	max	min	# cars	minutes saved
SSSD	1.58	0.34	2.81	0.85	194	10.14
MSSD	1.99	1.14	7.64	0.35	599	14.51
MSMD-sparse	1.04	0.21	2.04	0.50	269	0.21
MSMD-medium	1.48	0.77	6.01	0.32	1231	5.21
MSMD-dense	2.02	1.76	18.35	0.10	2375	13.97
SD Baseline						
SSSD	1.13	0.33	2.00	0.63	194	1.98
MSSD	1.38	0.78	4.80	0.20	599	2.85
MSMD	1.16	0.53	5.51	0.37	1231	1.19
ours+sensors over ours	1.23	0.178	1.612	0.824	2400	NA

Table 1: Performance speedup of our approach over the Shortest Path (**SP**) and Sensor-Data (**SD**) baseline route planning. Our method achieves up to a maximum speedup of 18.35 over SP baseline in congested traffic for the MSMD benchmark.

Please see **http://gamma.cs.unc.edu/PRP** for an appendix with additional data and a video demonstration.

5.4.1 Manhattan

A summary of the experimental results for the Manhattan scenarios can be seen in Table 1. The top half of the table shows the speed up statistics and the average time saved over the shortest-path baseline (**SP**). The bottom half of the table shows the same results for the sensor-data baseline (**SD**).

- **SSSD.** Our planner is able to get a mean speedup of 1.58 over **SP** by utilizing the spare capacity of the road network, saving an average of 10 minutes per car. One interesting aspect of this simple experiment is that our system was able to achieve a speedup even though the total number of cars was less than 200! For the **SD** baseline, using the same car population, we achieve a more modest speedup of 1.13.
- **MSSD.** For this experiment, the mean speedup of our system over the **SP** baseline was 1.99, and the mean time saved was 14.51 minutes. One reason this scenario has a higher speedup than **SSSD** is that the routes chosen must converge

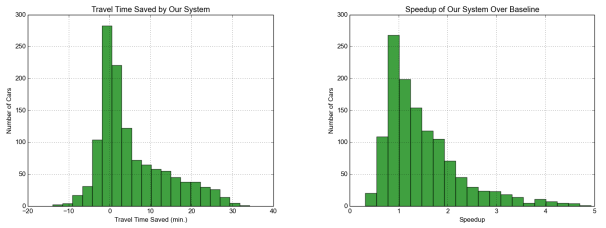

Figure 8: **MSMD Scenario:** These histograms show the performance of our system relative to the **SP** baseline router for a city-traffic scenario. On the left is shown the time saved by each vehicle by using our system, and on the right is shown the speedup of each car achieved by our system. For this example, the mean speedup of our system over the baseline was 1.48, and the mean time saved was 5.21 minutes.

on a single destination, creating areas where traffic flows together and creating congestion. For the **SD** baseline, we have a speedup factor of 1.38.

- **MSMD.** In the case of the highest traffic demand, our planner was able to achieve an average speedup of 2.02 over **SP** saving over 13 minutes of travel time. The sparse demand scenario shows the behavior when there is insufficient flow to cause congestion – there is no speedup and traffic lights cause random variations in travel time. Finally, in the mid-range scenario, our system achieved a speedup of 1.48, with 5 minutes of travel time saved. Our method achieved a small speedup of 1.16 over the **SD** baseline in this case. (For the dense case, the **SD** baseline created gridlock and its cars could not execute their routes.) A histogram of the speedup over **SP** for the middle density scenario can be seen in Figure 8.
- **Real-time Sensing.** We performed 20 randomized trials to investigate this system. For these trials, an average of 2400 cars were routed over a 40 minute period through the Manhattan road network using the same City-Traffic scenario set up described above. In Table 1, we present the speedup of our system with this real-time sensor data over our system without real-time sensor data, *Ours no sensor*. We can see that on average using this real-time sensing data provides an additional speedup. We believe this could be even greater if the vehicles in the system were able to transmit position updates, which is a functionality that cannot be simulated using SUMO.

5.4.2 Sioux Falls

The mean travel times for various flow levels can be seen in Figure 9. We can see that our method outperforms both baselines, achieving a mean time of 1189.99 seconds for twenty-five thousand cars, while **SP** had a time of 1996.89 and **SD** had a time of 1573.06, the speedups for which are 1.68 and 1.32. We can also see that at low flows, negligible speedup is present as there is less possible congestion.

Routing as traffic prediction. Our approach allows the routes of the vehicles to be used to estimate the future traffic pattern. This is possible because our routing system functions like a mesoscopic traffic simulator, i.e. a simulator in which individual vehicles are propagated while a macroscopic density field is maintained. Being able to predict the traffic pattern that results from the vehicles

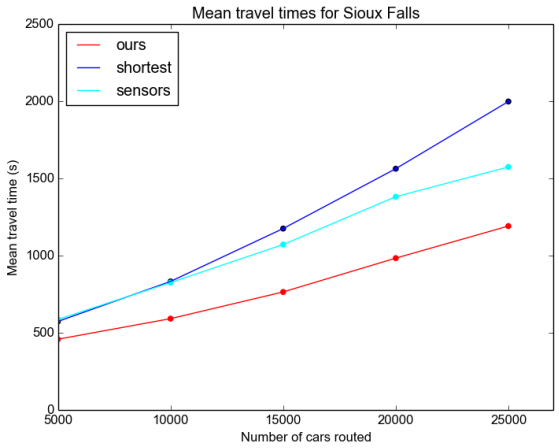

Figure 9: This figure shows the mean travel times for varying numbers of vehicles using our method, a shortest-path baseline (**SP**), and a commercial-like system using sensor data (**SD**). We observe that our method (red) outperformed both the shortest-path baseline (navy) and the commercial-like system with sensor data (sky blue).

following their routes is important: this enables the approach to be enhanced using real-time sensor data and historical prediction. If the system has a good prediction for the traffic state at a time, it can then fuse that prediction with incoming sensor data.

However, using routing to predict the traffic state is also difficult. Traffic is a complex dynamical system with continuous and discrete aspects. Only the average behavior of the traffic lights is known. A single point of congestion can grow outward through the network and cause large deviations.

In Figure 10, we can see our system's prediction for the evolution of the traffic state based on its routes. Two velocity fields are shown for our system's prediction (middle) and for the simulated ground truth (top). Each figure shows the mean velocity for each road over a 10 minute period. We can see that the system predicts many areas of congestion correctly. However, the ground truth has some congested jams of greater severity than predicted. These jams are caused by intersections with stop signs, which can create an arbitrarily large delay for vehicles, causing congestion that grows through the network. This level of delay is difficult to predict, and so the travel cost for these intersections can be easily underestimated.

5.5 System Performance

In these experiments we demonstrate the performance and responsiveness of our server. In this experiment, the server was run on a 4-core, 3.33 GHz machine with 6 GB of memory. The simulated clients were run on a separate machine with the same specifications. The experiment consists of simulated clients generating queries, sending them to the server, and receiving responses. The queries were generated by uniformly sampling random origin and destination positions.

In Figure 11 we can see the number of requests that the server can handle within varying periods of time. In this scenario, a single simulated client continuously sent queries to the server. We can see in Figure 11 that the server was capable of processing 6,000 routing requests within a period of ten minutes, which is a rate of 0.1 second per request.

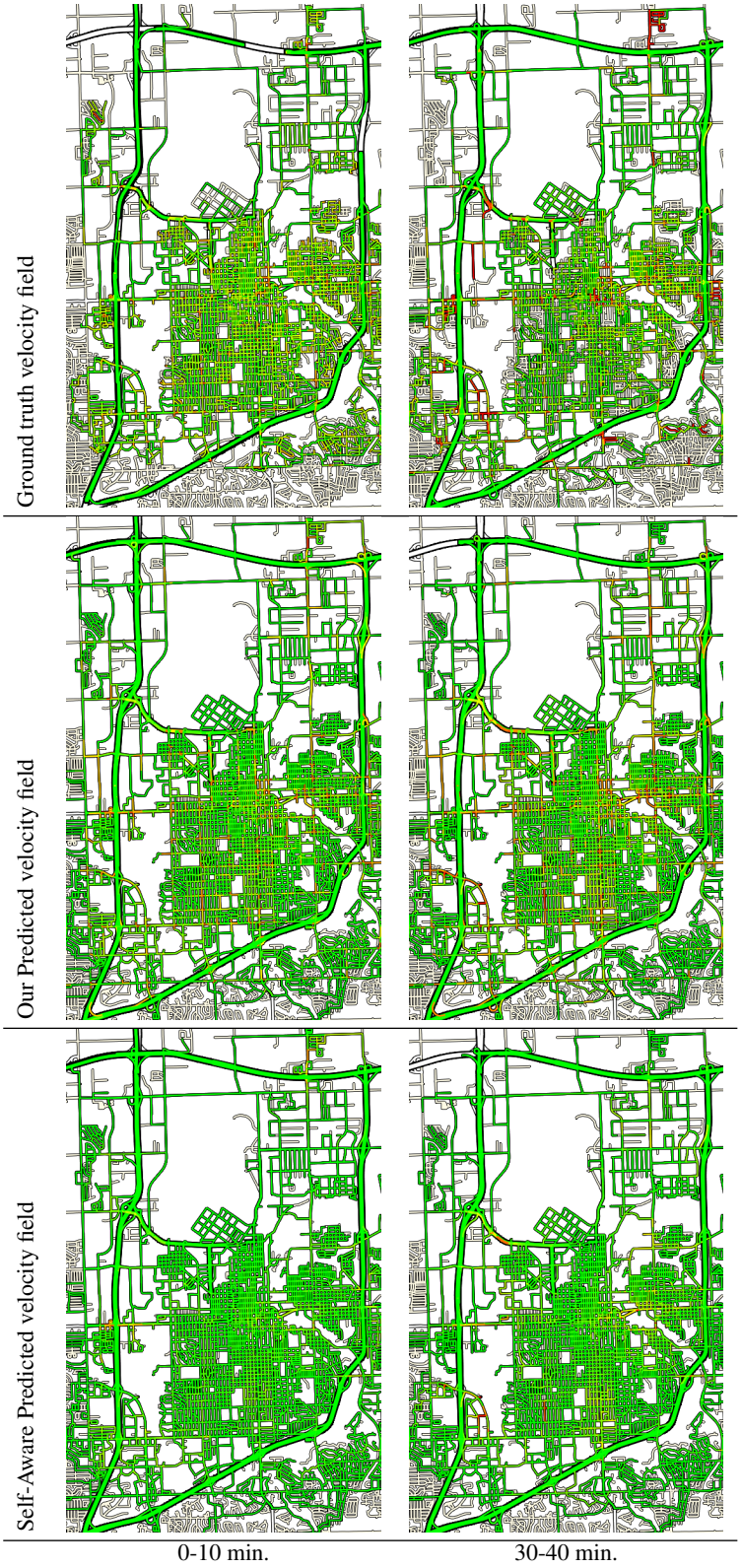

Figure 10: Each plot shows the mean velocity field over a ten minute time window, with a color ranging from red, 0 m/s, to green, 13 m/s. The top row is the ground truth; the middle row is the velocity field predicted solely by aggregating the routing requests of participating cars; the bottom row is the velocity field from using [27].

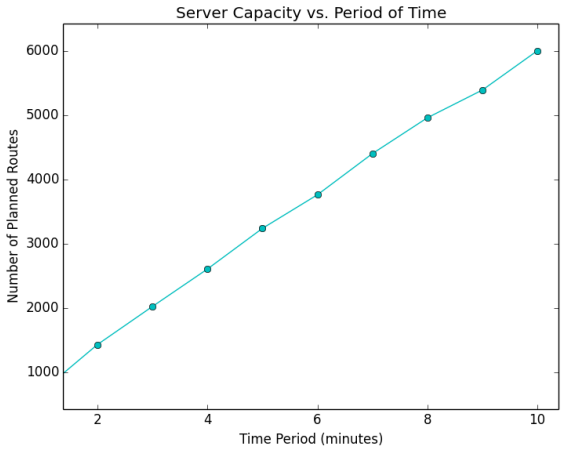

Figure 11: This figure shows the number of routes the server can generate in a period of time.

5.6 Discussion

The experiments show a mean speedup for all the scenarios, though a few vehicles do have a slower travel times. One factor that leads to this is the random effects of traffic lights: the actual timings of traffic signals are unknown. An identical route will perform better or worse than its equivalent depending on how many red lights a vehicle needs to stop at. Another factor is the traffic jam growth. A vehicle that is stopped by an unexpected traffic jam will have a much larger travel time than a vehicle traveling an otherwise identical route.

An interesting question is why the sensor-aware routing baseline does not do perform better, as knowing the current velocity instantaneously would seem to give a great advantage to routing. However, planning a route given the *current* conditions can actually be worse than using no traffic information at all. If the conditions for some road change before a vehicle reaches it, then its plan was based on faulty knowledge. For example, consider planning a trip from California to New York: certainly a traffic jam that exists in Ohio at the time of the query should not influence the planned route.

The experiments we performed have some limitations. First, the noisiness of real-world traffic and sensing are unknown and therefore not accurately accounted for in our simulation. Second, the traffic simulator, SUMO, has difficulty simulating highly congested urban traffic. In these cases, the simulator can experience gridlock, which prevents the simulation from terminating. This limited the density of cars we could generate in some experiments. We believe that our system would perform even better at higher demand levels.

6. CONCLUSION

We have presented an approach to coordinating vehicles using participatory route planning. With this method, planned vehicle routes are used as an additional source of information for estimating future traffic conditions, enabling our system to plan routes for a large portion of vehicles, or even for every vehicle, and achieve a speedup in travel time over planners that only use static conditions or only use historically-based predictions of traffic conditions. We have presented novel algorithmic contributions that enable our system to work in real-world conditions. We have also implemented and demonstrated a prototype mobile client-server system. Fur-

thermore, we have presented experiments that validate the performance and effectiveness of our system in terms of improved travel time and reduced traffic congestion. One possible future direction is combining this method with traffic flow reconstruction from real-time sensor data [25, 26] and historical prediction. We would also like to conduct in-depth investigations into the effect of varying the proportion of vehicles participating in the system, higher traffic loads, and having simulated vehicles providing real-time updates, which was not possible with SUMO. Finally, we plan to further improve the computational efficiency of our approach.

Acknowledgements: This research is supported in part by National Science Foundation, Award #IIS-1247456.

7. REFERENCES

[1] S. Amin, S. Andrews, S. Apte, J. Arnold, J. Ban, M. Benko, R. M. Bayen, B. Chiou, C. Claudel, C. Claudel, T. Dodson, O. Elhamshary, C. Flens-batina, M. Gruteser, J. carlos Herrera, R. Herring, B. Hoh, Q. Jacobson, T. Iwuchukwu, J. Lew, X. Litrico, L. Luddington, J. Margulici, A. Mortazavi, X. Pan, T. Rabbani, T. Racine, E. Sherlock-thomas, D. Sutter, and A. Tinka. Mobile century using gps mobile phones as traffic sensors: A field experiment, 2008.

[2] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

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

[4] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. *Proc. of ACM WSW at SenSys 2006*, 2006.

[5] A. Chakirov and P. Fourie. Enriched sioux falls scenario with dynamic and disaggregate demand. Working paper, Future Cities Laboratory, Singapore - ETH Centre (SEC), Singapore. 2014.

[6] R. Church and C. ReVelle. Modelling an oversaturated intersection. *Transportation Research*, 12(3):185–189, 1978.

[7] K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, pages 530–537, Washington, DC, USA, 2004. IEEE Computer Society.

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

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

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

[11] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: A distributed mobile sensor computing system. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 125–138. ACM, 2006.

[12] W. Jariyasunant, S. Kerkez, and B. Glaser. Mobile transit trip planning with real-time data. In *Transportation Research Board 89th Annual Meeting*, 2010.

[13] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[14] L. J. LeBlanc. An algorithm for the discrete network design problem. *Transportation Science*, 9(3):183–199, 1975.

[15] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, 9(5):309–318, 1975.

[16] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311 – 331, 2007.

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

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

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

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

[21] S. Russell, P. Norvig, and A. Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25, 1995.

[22] J. Sewall, D. Wilkie, and M. C. Lin. Interactive hybrid simulation of large-scale traffic. *ACM Transaction on Graphics (Proceedings of SIGGRAPH Asia)*, 30(6), December 2011.

[23] J. Sewall, D. Wilkie, P. Merrell, and M. C. Lin. Continuum traffic simulation. In *Computer Graphics Forum*, volume 29, pages 439–448. Wiley Online Library, 2010.

[24] C. Suwansirikul, T. L. Friesz, and R. L. Tobin. Equilibrium decomposed optimization: a heuristic for the continuous equilibrium network design problem. *Transportation science*, 21(4):254–263, 1987.

[25] D. Wilkie, J. Sewall, and M. Lin. Transforming GIS data into functional road models for large-scale traffic simulation. *IEEE Trans. on Visualization and Computer Graphics*, 18(6), 2012.

[26] D. Wilkie, J. Sewall, and M. Lin. Flow reconstruction for data-driven traffic animation. *ACM Trans. on Graphics (Special Issue of ACM SIGGRAPH)*, 2013.

[27] D. Wilkie, J. P. van den Berg, M. C. Lin, and D. Manocha. Self-aware traffic route planning. In *AAAI*, 2011.

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