The Effect of Congestion Frequency and Saturation on Coordinated Traffic Routing

Melanie Smith and Roger Mailler

Computational Neuroscience & Adaptive Systems Lab
Department of Computer Science
University of Tulsa, Oklahoma
{melanie,mailler}@utulsa.edu
http://www.cnas.utulsa.edu

Abstract. *Traffic congestion is a widespread epidemic that continually wreaks havoc in urban areas. Traffic jams, car wrecks, construction delays, and other causes of congestion, can turn even the biggest highways into a parking lot. Several congestion mitigation strategies are being studied, many focusing on micro-simulation of traffic to determine how modifying road structures will affect the flow of traffic and the networking perspective of vehicle-to-vehicle communication. Vehicle routing on a network of roads and intersections can be modeled as a distributed constraint optimization problem and solved using a range of centralized to decentralized techniques. In this paper, we present a constraint optimization model of a traffic routing problem. We produce congestion data using a sinusoidal wave pattern and vary its amplitude (saturation) and frequency (vehicle waves through a given intersection). Through empirical evaluation, we show how a centralized and decentralized solution each react to unknown congestion information that occurs after the initial route planning period.*

Keywords: routing, coordination, constraint optimization

1 Introduction

Traffic congestion is a familiar and frustrating occurrence for anyone who drives on the road. In fact, congestion is continually getting worse, affecting more of the road system (secondary, tertiary roads), during more times of the day, and with rising penalties for extra travel time [5]. Except for consistent and repetitive physical bottlenecks (causing 40% of congestion), congestion occurs with maddening irregularity where nothing is the same from one day to the next. Studies show that the unpredictability of traffic incidents or inclement weather cause the majority of congestion because no one can accurately determine how different factors such as time-of-day, weather conditions, number of vehicles on the road, etc, will affect traffic [5]. A domino effect also occurs due to road congestion [5].

Because congestion causes are all related, significant payoffs can be expected by treating them. By decreasing the effects of congestion, the total delay experienced by travellers is reduced, time and fuel are saved, vehicles cause less emissions, safety for travelers is increased, the benefits for just-in-time manufacturing are increased, overall transportation costs are decreased, and benefits are provided to the national economy [5].

An obvious approach to mitigating congestion is changing road structures and governing rules to increase traffic flow and/or lighten the demand for a particular road. The problems are the physical, financial, and political limits that cap the effectiveness of these kinds of solutions. Instead, we ask the question: how can we better utilize what is already in place, making our roadways more efficient? This is the impending problem as our society continues to grow and physical expansions of roads become less and less feasible [3].

Some Global Positioning System (GPS) devices have the ability to warn drivers of upcoming congestion and automatically suggest alternative routes. However, GPS units typically use similar path planning algorithms to reroute users, meaning that drivers will often be advised to take the same detour. Thus, detouring does not always help the individual driver as others receive the same advice, moving the jam to secondary/tertiary roads where drivers will again be faced with the same congestion, but on a smaller road.

A separate, but substantial, issue related to route coordination is not only how the vehicles coordinate with one another, but also how they mitigate congestion due to vehicles on the road who do not use any routing or coordinating to get from one place to another. Essentially, this type of congestion travels in waves across the road network. The "green phase" of a traffic light is analogous to the frequency of the wave and the amount of traffic flowing determines the amplitude of the wave (or saturation level).

In most cases, approaches to traffic management only deal with how to distribute and use global information in order to make local (per vehicle) routing decisions. To our knowledge, very few address issues on how to best use all the information that is shared in the system and how to make routing decisions to satisfy both local and global objectives (e.g. individual driver route preferences and minimizing overall network congestion) while also reacting to external congestion.

In this paper, we discuss the traffic domain as a constraint optimization problem and use our coordination algorithm for vehicle routing decisions. We then generate congestion in the network over varying frequency (waves per hour) and saturation levels to compare a centralized and decentralized technique.

2 Background

Traffic modelling takes place on three distinct levels: microscopic, mesoscopic, and macroscopic [1]. Microscopic models consider each vehicle individually and model each according to its own position, headway gap, and velocity (e.g. [11]). Essentially, vehicle trajectories are produced as the vehicles move through the system using rules that dictate how they move and interact, including acceleration, deceleration, lane changing, and passing maneuvers [1]. Macroscopic models aggregate the data depicted by the micro models and use average velocity and density numbers to determine the flow over a particular roadway. Mesoscopic models are considered to have an intermediate level of detail, modelling individual vehicles, but describing their interactions and actions from a macroscopic perspective.

Balbo [2] also uses an agent-based approach to design a Transportation Regulation Support System called SATIR. SATIR integrates a decision support system with an automatic vehicle monitoring system. This is similar to our work in that it combines different architectural units in the simulation environment. However, SATIR focuses on public transit bus system management and routing them in real-time in order to minimize the delay of the bus schedule.

Another similar work investigates the effect of route information sharing over lattice, radial-ring, and a real road network [13]. Their results show that as the number of vehicles that share route information increases, that congestion within the system decreases. However, they only consider a highly centralized approach and vary the percentage of vehicles that participate in the information sharing.

Wunderlich, et al [12] investigate link travel time for decentralized route guidance and find that predictive route guidance can be supported within the constraints of a decentralized architecture. Both the SAVaNT and SAVaNT-CNV approaches are studied, but no comparison is shown between different degrees of centralization as they focus more on the percentage of guided versus unguided vehicles, or how congestion changes as more vehicles adopt their guidance technology. The DIVERT simulation system deals with large-scale environments and vehicle-to-vehicle communication [7]. DIVERT focuses on learning the road network and on the ability and feasibility of networked communication between vehicles.

Centralized systems are generally infrastructure-based and require roadside equipment to upload information into the centralized server [10]. Because all information is aggregated in one place, there is a complete global view that allows the entire system to be evaluated, and given infinite time and computational power, could garner an optimal routing solution. However, optimality is prohibitively expensive. Otto found that having a global perspective allows a greater time savings in travel times than the distributed, local perspective, but that savings comes at an additional cost in route length to drivers [10].

Distributed systems totally rely on V2V communications instead of V2I (vehicle-to-infrastructure). They take advantage of direct exchanges between participating vehicles to achieve higher scalability, but at the risk of losing data consistency. By using V2V communication, deployment and maintenance costs are less. However we are then faced with questions of user adoption and how much effect imprecise information will have on the vehicles. Taken from a local perspective, Otto shows an improvement in reliability, scalability, and cost without negatively impacting travel time and distance relative to a centralized version [10].

Additionally, Otto compares centralized and distributed methods using several routing algorithms and several different neighborhood sizes for the distributed environment: information is gathered from either a two block radius around the vehicle's current location, from the full route within two intersections, or from a quarter of the route within four intersections. The study finds that looking at the congestion information for the entire route was pointless because congestion levels may change before the vehicle reaches that point in the route, which wastes significant bandwidth communication what becomes useless information. Even though the centralized algorithm assigns routes to vehicles upon request, it does not use the route request data to help predict how congestion will be in the future, it only factors in the current congestion levels being experienced.

3 Modeling Traffic as a Constraint Optimization Problem

To model the traffic domain within a constraint optimization problem, we define the road map to be a directed graph with a set of nodes representing intersections and a set of edges that represent road segments, or links. A route is a path, or a sequence of connected intersections, that is traversed in order and does not contain cycles.

Let $M(P, Q)$ be a directed graph representing a road map with a set of nodes, P, representing intersections and a set of edges, Q, representing road segments, or links. Each $q \in Q$ connects two intersections, $p_i \in P$ and $p_j \in P$, and is denoted as $(p_i, p_j) \in Q$. A route is a path, or a sequence of connected intersections, that is traversed in order and does not contain cycles. For any route, $r =< p_1, p_2, \ldots, p_n >, \forall i \mid 1 \leq i < n, p_i \in P \bigwedge p_{i+1} \in P \bigwedge (p_i, p_{i+1}) \in Q$.

The weight of the elements in Q is the degree of saturation ($saturation = flow/maxflow$) on that element. This is defined as $w(x) = x.saturation \mid x \in Q$. For $p \in P$, the weight is the queue length, or the number of vehicles waiting to pass through the intersection.

When cast as a constraint optimization problem, following from Yokoo's original definition [14], each vehicle acts as an agent with start, destination, and departure time variables. There exists a set of possible routes for each vehicle, and each are evaluated using a cost function chosen by that vehicle. We define the traffic routing component as:

- a set of n vehicles $V = \{v_1, \ldots, v_n\}$.
- a set of start places $S = \{s_1, \ldots, s_n\}$, where each s_i is the start place for v_i.
- a set of goal places $G = \{g_1, \ldots, g_n\}$, where each g_i is the goal place for v_i.
- a set of start times $T^0 = \{t_1^0, \ldots, t_n^0\}$, where each t_i^0 is the start time for v_i.
- a finite, discrete set of possible routes for each of the vehicles $R = \{R_1, \ldots, R_n\}$ where each R_i contains the set of routes to which the associated vehicle, v_i, may be assigned. Each $r \in R_i$ is a path from s_i to g_i, and are ranked according to the anticipated cost that the route will entail.

– a set of cost functions $f = \{f_1, \ldots, f_n\}$ where each $f_i(r) \mid r \in R_i \bigwedge r = v_i.currentRoute$ is function

$$f_i(r) : \sum_{j=1}^{r.length-1} travelTime(r_j, r_{j+1}) \quad (1)$$

For this work, we use the same cost function for all vehicles that minimizes total travel time. Each vehicle is then assigned to a coordinator agent, who is responsible for coordinating routes among the vehicles it controls. Thus, there is a set of k coordinator agents, $C = \{c_1, \ldots, c_k\}$ and a mapping function $\alpha : C \to V$, implying that $\alpha(c_i) \to \bar{V}$ ($\bar{V} \subset V$) states that it is coordinator c_i's responsibility to assign routes to vehicles in the set $\bar{V}$. In the centralized approach, all vehicles are assigned to one coordinator, so $C = \{c_1\}$ and $\bar{V} \equiv V$. Distributed approaches have one coordinator per vehicle, so in that case, $k = n$.

The problem is to find an assignment $R^* = \{r_1, \ldots, r_n \mid r_i \in R_i\}$ such that the global cost, called F, is minimized. We define F as follows:

$$F(R^*) = \sum_{i=1}^{n} f_i(r_i) \quad (2)$$

4 Coordinated Routing

The A* algorithm is a best-first graph search algorithm that is used for finding the optimal (least-cost) path from a start point to an end point in a directed, acyclic graph [8]. A* uses a combination of the predicted distance to the goal and the known cost to the current position instead of just a cost heuristic: $f(x) = g(x) + h(x)$, where $g(x)$ is the known cost from the start point to the current position and $h(x)$ is the estimated distance to the end point.

We set $h(x)$ as the euclidian distance from the current point to the destination divided by a constant speed limit that ensured admissibility to give us an estimated cost in time units. The straightline distance proves to be much less than the actual distance travelled between two points because typically taking roads means traveling in a grid-like manner. The generic equation for $g(q)$ is as follows.

$$g(q) = g(p) + runningTime(p, q) + queueingTime(q) \quad (3)$$

Given that q is a successor to p, meaning there exists a road segment that flows out of p and into q, the actual cost of traveling from p to q is the time it takes for a vehicle to drive from p to q. The time spent waiting at an intersection is also added in.

By combining congestion information with the routing specifics of each vehicle, we can coordinate routes between vehicles, giving us a more efficient use of the existing road network. Routing is by itself an expensive operation; coordinating efforts between several entities that are searching for a route together only magnifies the cost.

The other modeling construct facilitates data dissemination. Studies have looked at centralized versus distributed data dissemination [10], but have not paired that completely with vehicular control. Others have also looked at how quickly information can be accumulated as vehicles drive through a network [6]. We incorporate congestion data using the concept of neighborhoods, where information can be shared among neighbors. Two vehicles can be considered neighbors if they are assigned to the same coordinator (siblings) or if they are physically located on the same intersection or link (local), and in some cases we consider all vehicles to be neighbors (global). Each vehicle can share its own route plan and destination, but does not pass along other information it may have learned previously from other vehicles. Each coordinator also has congestion information knowledge for the any reporting entities on the network, for example, this information would include saturation information that summarizes vehicles that are not being simulated.

4.1 Centralized

Algorithm 1: Centralized Coordination Algorithm

cong ← initial congestion knowledge;
foreach *vehicle* $v_i \in \bar{V} | \alpha(c) \to \bar{V}$ **do**
    routes ← `route`(v_i, *cong*);
    `update_congestion_info`(*routes*, *cong*);
end

The centralized algorithm takes an initial snapshot of the environment's congestion and then begins the routing process. Then it uses an A* search that takes into account the known congestion and the anticipated congestion of all the previously routed vehicles. In order to do this, we must make the assumption that order is preserved, meaning vehicles plan their routes in the same order they are moved forward during the simulation. For example, if we know v_1 is the first vehicle, we can accurately predict the amount of congestion it will encounter. Order preservation is restrictive, however to solve without that restriction, there is need to run a solver before the simulation can begin and again every time rerouting is desired. It is conceivable to prioritize vehicles in order of submitting their information to the coordinator, which is what we have simulated. To consider different orderings and find an optimal solution, the A* search grows to be on the order of $(3^n)^h)$ where 3 is the typical number of outgoing links, n is the number of vehicles, and h is the maximum number of route hops over all vehicles. For a set of 25 vehicles with an average of 8 hops, it is about $O(10^95)$ each time routes are calculated. In algorithms where rerouting is allowed, the ordering is different each time routes are recalculated, so over time, order preservation becomes less restrictive.

4.2 Decentralized

While the Centralized Algorithm plans routes for all vehicles before the simulation begins, the Decentralized Algorithm updates routes as the vehicles are traveling in much the same fashion as the Distributed Stochastic Algorithm (DSA) [15]. As previously mentioned, the decentralized approach assigns a separate coordinator to each vehicle in the simulation. This means that instead of having information on all vehicles in the simulation, this algorithm only considers information from a vehicle's neighbors. A neighbor is considered any vehicle that is co-located at the same intersection or link. At each time step where the vehicle is at an intersection, this algorithm (see Algorithm 2) determines who its neighbors are, which we define as any other vehicle that is stopped at the same intersection at the same time. At each time click, the neighborhood can contain a different set of vehicles and thus a different set of knowledge.

The *update_congestion_info* method is the same as with the Centralized algorithm except that instead of updating it from the beginning of all the vehicle's routes, it picks up at the current position of the vehicles in order to keep from calculating congestion information that is in the past. It also only updates known congestion info that has been collected from its neighbors. If stale information is passed, it is ignored.

Algorithm 2: Decentralized Coordination Algorithm

```
neighbors ← current congestion info from neighbors;
update_congestion_info(neighbors, cong);
foreach vehicle v_i ∈ V̄|α(c) → V̄ do
    neighbors ← route(v_i, cong);
    update_congestion_info(neighbors, cong);
end
foreach cycle do
    foreach vehicle v_i ∈ V̄|α(c) → V̄ do
        if random < p then
            neighbors ← route(v_i, cong);
            update_congestion_info(neighbors, cong);
        end
    end
end
```

5 Simulator & Experimentation Setup

Our vehicles operate in a simulated queue based traffic model (see Figure 1) that runs inside the FARM simulator [9], meaning that each road segment, or link, is split into two parts, one where traffic is flowing normally and the other where traffic is queueing at the intersection. Each link has a maximum number of

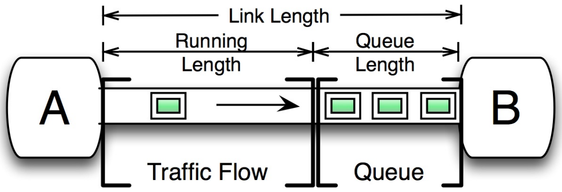

Fig. 1. Queue Based Traffic Model.

vehicles that are allowed to flow over the link simultaneously, and as the number of vehicles approaches the maximum, congestion is experienced. The effective length of a link is shortened by the length of the queue, meaning that if the queue contains the maximum number of vehicles, there is no movement. Each intersection has one queue per incoming link. Each queue is given a "green light" that allows a certain number of vehicles to pass through the intersection. Within each single queue, vehicles leave in the same order they arrived, however across all queues flowing into one intersection, that may or may not be the case.

To determine the velocity of the vehicles traveling over the link, we use a Weibull distribution (see Equation 5) for any saturation level that is over the free flow density ($k_{free} = 60\%$ saturation) and less than the jam density ($k_{jam} = 95\%$ saturation). $Maxflow$ is the capacity on the link, which was 10 vehicles for these simulations The Weibull distribution choice allows us to take advantage of a multistage model, which is very similar to the DynaMIT model [4].

$$\begin{gathered} 1 + (avgvel(k_{free} * maxflow) - 1) * \\ 1 - ((saturation - k_{free})/(k_{jam} - k_{free}))^2)^2 \end{gathered} \tag{4}$$

This equation is also contingent upon the average velocity that a vehicle travels while on a link. The distance a vehicle travels during each cycle of the simulator is calculated based on the flow experienced on that link at that time. The average velocity is calculated as the current running length (total length (len) minus the queue length (ql)) of the link divided by the time taken to accelerate (ta), decelerate (td), and cruise (tc) from the time the vehicle began its journey (ti).

$$avgvel = \frac{len - ql}{ta + td + tc - ti} \tag{5}$$

Congestion data is generated in a wave pattern that is routed between the start and destination regions. The wave travels from one intersection to the next, and if the congestion arrives from several incoming links, it is combined on the outgoing link. The phases are related, and an example can be found in Figure 2. The thickness of the edges indicates the saturation levels and the thickness is proportional to the amount of congestion on that link. The sinusoidal wave propagation formula is as follows:

$$saturation * (0.5 + 0.5 * sin(phase - \frac{\pi * time * wph}{1800})) \quad (6)$$

When a new vehicle is routed, it can calculate what its average velocity and wait times will be as it progresses through its route, based on the congestion data and the knowledge of the routes other vehicles are planning to take.

By including the knowledge of other vehicles and any other existing congestion on the system in the evaluation function, each subsequent vehicle to be rerouted can make the best decision based on that information. The $update\_congestion\_info(routes, cong)$ method creates a table containing traffic flow information for each link and queue length for each intersection, keeping counts for each entity over time. When a new vehicle is routed, it can calculate what its average velocity and wait times will be as it progresses through its route. Generating these information tables is a similar process to building the optimal search tree: all vehicles are initially scheduled to move at their start time, and as the table is filled out for each time click, the counts are updated when vehicles are expected to leave or arrive at a new intersection.

Rerouting occurs when the encountered congestion is greater than the expected congestion level that was identified during the route planning phase. The expected congestion is not applicable for the cases where no congestion data is used, and is therefore set to 0. The rerouting is triggered by encountering congestion along the route, but congestion data is not known for any other intersection besides the current location of the vehicle in this case, simulating the visual perception of congestion that a driver can see ahead of him.

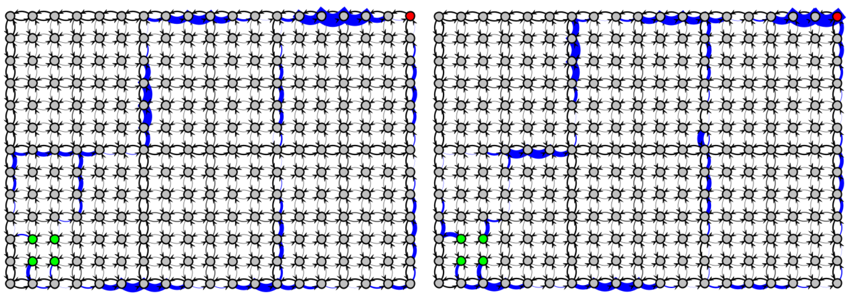

Fig. 2. Wave Propagation as Time Changes from $t = t_i$ (left) to $t = t_{i+k}$ (right).

The simulations presented in this paper are based on a 25 to 10 density ratio of vehicles to the maximum number of vehicles that can be on a link at a given time, and is evaluated over a small-area grid map (∼6 square miles). Vehicles start from a four-intersection cluster and travel to the same destination intersection. This would be analogous of an evacuation scenario where everyone in a particular area gets sent to the same shelter location. We use the map shown in Figure 2 that has arterial roads each mile with speed limit of 45mph, secondary streets each half mile with speed limits of 35mph, and tertiary roads with speed limits of 25mph.

All vehicles attempt to start at the same time, however this is staggered by the saturation and frequency of the external congestion and the queueing of the vehicle at the initial intersection. Thus, actual start times are dependent on whether another vehicle can fit on the road given the external congestion and whether the intersection will let the vehicle through at a given time step. Time steps of the simulation roughly represent an actual second. However, it is also the case that when rerouting occurs, it occurs within one time step and generally causes the time step to take much longer than an actual second. Simulation continues until all vehicles have arrived at their destination.

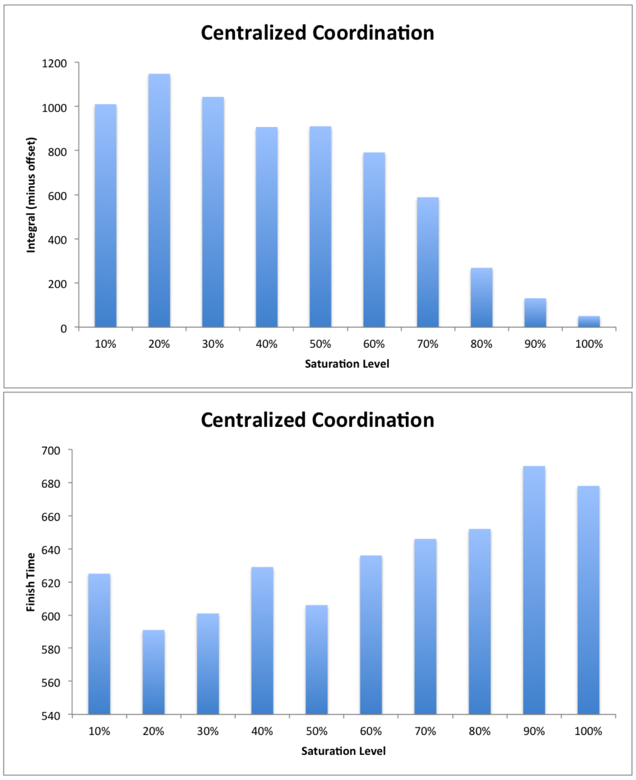

Fig. 3. Averaged Centralized results for Saturation Variance. Top: Integral of the Number of Vehicles that have reached their destination over time. Bottom: Finish time for the last vehicle.

6 Results

We show results for varying the saturation level from 10% to 100% in 10% increments at 20 waves per hour and varying the waves per hour from 5 to 60 in increments of 5 at 60% saturation for a centralized approach and a decentralized approach.

To measure the effects that varying the saturation and wave frequency, we keep track of the number of vehicles that have arrived at their destination at each time click. As the finished vehicle count approaches the number of vehicles in our simulation, we can compare two different aspects of time spent traveling. First, there is the obvious comparison of which approach got ALL the vehicles to their destination in the least amount of time. This is equal to the finish time for the last vehicle to arrive at its destination.

However, that metric does not consider the rate at which the vehicles finish, or whether more vehicles finish faster regardless of when the last vehicle arrives. For instance, which is superior, a route scheme that takes everyone 10 minutes to reach their destination or a route scheme where it takes some vehicles 5 minutes, another set 7 minutes, and the last vehicle takes 11 minutes? This requires a different metric, and is the one we emphasize as showing a more realistic view of how well the approach solved the problem. To calculate this, we take the integral under the curve of vehicles finished over time, so the higher the number, the more vehicles finish faster. In the graphs below, we do offset the numbers for ease of graphing because the difference between them is the important part, not the actual value.

Figure 3 shows the centralized results for the variations on saturation level. On the top is the last vehicle's finish time, which increases (predictably) as saturation increases in a somewhat linear fashion. However, the story left untold by that graph is that up until 70% saturation, the algorithm doesn't notice a dramatic drop in effectiveness. The integral (bottom) results show that there's an increase in the rate of decline as saturation rises towards the fully saturated end. This means that the vehicles are ALL taking longer as a whole to complete their route and having some trouble dealing with congestion. For example, compare the Integral results of 60%, 70%, and 80%. The finish time of the last vehicle does not change significantly, but the integral numbers show that a significantly larger number of vehicles finish sooner in the 60% scenario versus 70% and in the 70% versus 80% scenario. This result shows us that even a tiny drop in congestion can make a substantial impact on many drivers.

Figure 4 shows centralized results for the variations on frequency, or waves per hour. This is like saying a wave of congestion comes through the network every 1 minute when there are 60 waves per hour. The top shows the integral results and the bottom contains the last vehicle's finish time. The graphs visually break into three sections: 5-15 wph, 20-25wph, and 30-60wph.

Likely the first batch (5-15wph) are not going to be seen very often in real life, as it is rare to see a 12 minute long green light. The next batch deals with the 2.4-3 minute wave frequency, which for large intersections is common. Our results show that having congestion waves at this frequency has a largely negative impact on getting vehicles to their destination, emphasized by the fact that the integral values are lower than the rest and the finish times are much higher. The last batch (1-2 minute frequency range) shows relatively similar results except for the 60wph case. In this case, the graphs show a high integral value (the highest disregarding the first batch) and one of the lowest overall finishing times. This shows a nice balance of the two, and a marked improvement over the 55wph case that boasts the same low finish time but a significant drop with the integral.

Another reason for seeing improvement with increased frequencies is that the centralized algorithm's initial congestion snapshot used for initial route formulation and because of the higher frequency, it is never too far off from reality. However, with lower frequencies the consequences of inaccurate or outdated congestion information rise. Additionally, the higher the frequency, the more often there is a lower-congested gap for our coordinated vehicles to enter into.

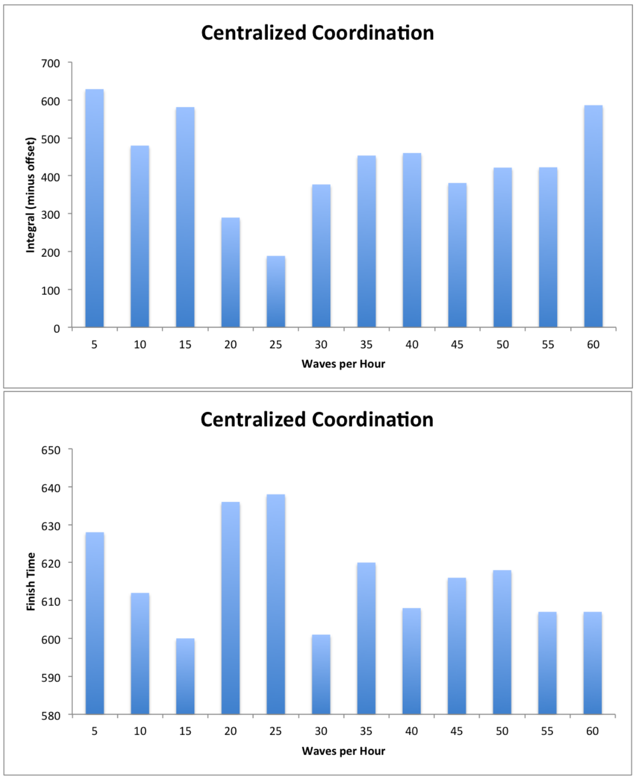

Fig. 4. Averaged Centralized results for Frequency Variance. Top: Integral of the Number of Vehicles that have reached their destination over time. Bottom: Finish time for the last vehicle.

Figure 5 compares the saturation variance between the centralized and decentralized approaches for one run (the previous graph in Figure 3 were averaged over all runs). The top graph shows the raw data and the bottom shows the difference between the two. What we are highlighting here is the shift from how the decentralized approach is better for the extreme ends and the centralized approach works better for the 30%-50% saturation levels. Although the differences are relatively small, the trend appears in the bottom graph where a 0 value indicates both approaches are equivalent, centralized outperforms decentralized when the bars are below the axis, and decentralized does better when the bars are above the axis. We graphed these numbers like this to show how as saturation increases in the 70% - 100% range, the benefits of using a decentralized approach become more pronounced.

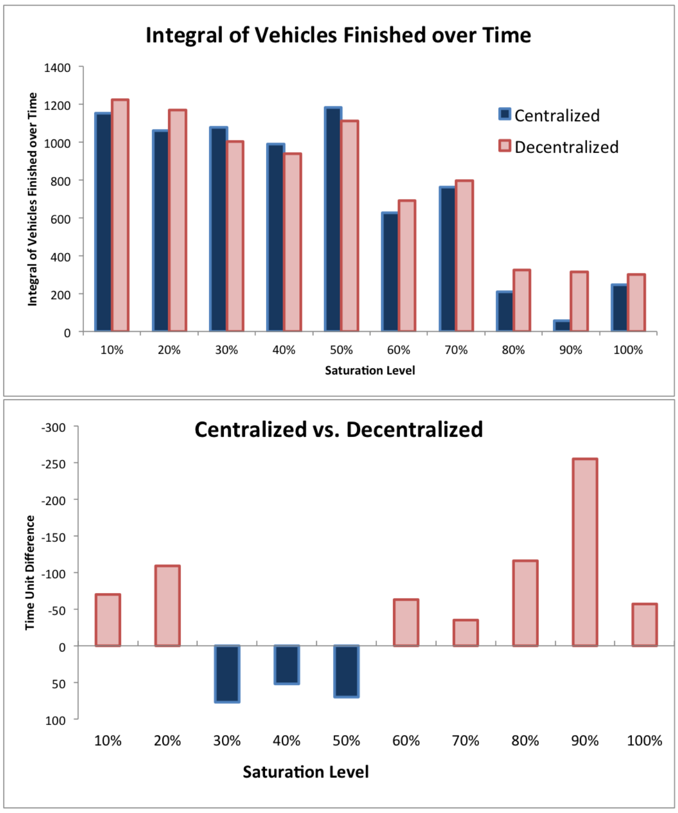

Fig. 5. Comparison of Centralized and Decentralized Solutions: Integral of Vehicles Finished over time for varying the Saturation Level (top) and the difference between centralized and decentralized integrals (bottom). The bars indicate the difference between each solution. The x-axis is the "break-even" point, and for instances where the centralized solution is better, the bar goes downward and where the decentralized solution is better, the bar goes upward.

7 Conclusion

In this paper, we discuss how coordinating vehicle routes improves the ability for vehicles to maneuver through the system. We discover that the amount of dynamic congestion information impacts how quickly a set of vehicles traverses through the system due to their ability to communicate and dynamically route around congestion.

In any case where a road network is close to fully saturated with vehicles, there does not exist a good solution when you consider congestion that is outside the route sharing and coordination environment. No matter how good the algorithm is, it is limited by the participation of the vehicles in the system (i.e. how much congestion is due to vehicles that are not being coordinated and sharing route information) and by the physical capacity of the road. However, by using a decentralized approach, we find that vehicles are more equipped to react to encountered congestion. We also find that small changes in congestion saturation can positively impact the system as a whole, and that having 60 waves per hour at intersections allow vehicles to move more effectively throughout the road network.

Adopting a technology like coordinated routing that involves a human taking direction from a computer device in order to help solve a problem also creates another problem. By coordinating a driver's route with the known routes of other drivers so that everyone gets where they are going at a better time, we also run the risk of the system being frustrating to use. For instance, if a route keeps changing every 3 minutes, the driver will get frustrated and turn it off or disregard its instructions. Thus, as this work continues, an important consideration is how well the technology will be received by its users. This means that no significant amount of time should pass causing the user to wait for a route – the user should be able to get in the car, find a route quickly, and get on their way. Second, there should be some stability in the route over time so as not to confuse the user or cause unnecessary lane changes and maneuvers in order to quickly get in the correct turn lane. Lastly, the system should balance the emotional effects that encountering congestion has on drivers with the ability to mitigate that congestion. If it is more frustrating to put up with the technology than it is to deal with the congestion, there is no purpose to deploying this type of technology. However, if a system can guarantee users to reliably get them to their destination in the least possible time, then this has the possibility of opening up a new market for GPS manufacturers to those people who may not need directions, but would benefit from the intelligence built into the system.

References

1. Administration, F.H.: Traffic analysis toolbox volume iii: Guidelines for applying traffic microsimulation modeling software. Tech. Rep. FHWA-HRT-04-040, USDoT (2004)

2. Balbo, F., Pinson, S.: An agent oriented approach to transportation regulation support systems. Proceedings of the 5th Workshop in Agent in Traffic and Transport (2008), http://www.lamsade.dauphine.fr/FILES/publi931.pdf
3. Bazzan, A.L.C.: Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. AAMAS 18(3), 342–375 (Jun 2009), http://www.inf.ufrgs.br/~bazzan/%23pub
4. Ben-Akiva, M.: Development of a deployable real-time dynamic traffic assignment system, task d interim report: analytical developments for dta system. Tech. rep., MIT ITS Program, Cambridge, MA (1996)
5. Cambridge Systematics Inc.: Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation pp. 1–140 (Sep 2005)
6. Conceição, H., Damas, L., Ferreira, M., Barros, J.: The divert project: Development of inter-vehicular reliable telematics. OSGeo Journal 3, 51–56 (2007), http://www.dcc.fc.up.pt/~michel/homepage/publications/2007-OSGeo-2.html
7. Conceição, H., Ferreira, M., Barros, J.: On the urban connectivity of vehicular sensor networks. LNCS 5067, 112–125 (2008), http://www.dcc.fc.up.pt/~michel/homepage/publications/2008-DCOSS.html
8. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions of Systems Science and Cybernetics 4(2), 100–108 (Aug 1968)
9. Horling, Bryan; Mailler, R., Lesser, V.: Farm: A scalable environment for multiagent development and evaluation. 2nd Int'l Wksp on Software Eng for Large-Scale Multi-Agent Systems (SELMAS 2003) pp. 171–177 (May 2003)
10. Otto, J.S., Bustamante, F.E.: Distributed or centralized traffic advisory systems - the application's take. SECON pp. 1–10 (Sep 2009)
11. Wahle, J., Schreckenberg, M.: A multi-agent system for online simulations based on real-world traffic data. 34th Hawaii Int'l Conf. on System Sciences (2001), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=926332
12. Wunderlich, K.E., Kaufman, D.E., Smith, R.L.: Link travel time prediction for decentralized route guidance architectures. IEEE Transactions on Intelligent Transportation Systems pp. 4–14 (Mar 2000)
13. Yamashita, T., Kurumatani, K.: New approach to smooth traffic flow with route information sharing. In: Bazzan, A., Klugl, F. (eds.) Multi-Agent Systems for Traffic and Transportation Engineering. pp. 291–306. IGI Global (2009)
14. Yokoo, M., Durfee, E.H.: Distributed constraint optimization as a formal model of partially adversarial cooperation. Tech. Rep. CSE-TR-101-91, University of Michigan, Ann Arbor, MI 48109 (1991)
15. Zhang, W., Wang, G., Wittenburg, L.: Distributed stochastic search for constraint satisfaction and optimization: Parallelism, phase transitions and performance. In: Proceedings of the AAAI Workshop on Probabilistic Approaches in Search (2002)