When to Leave Your Office? A Schedule-Driven Approach to Demand Shifting

Hsu-Chieh Hu and **Stephen F. Smith**
Carnegie Mellon University, Pittsburgh, PA, USA
hsuchieh@andrew.cmu.edu
sfs@cs.cmu.edu

Abstract

In this paper, we propose an approach that alleviates rush hour congestion through scheduling vehicles' access to the bottleneck road segments. We consider a setting in which each participating driver has a window of feasible departure time, and has to access several bottleneck road segments (e.g., tunnels or bridges) during the driver's itinerary. Our goal is to produce a timetable, which provides a specific departure time for each driver so that the peak traffic through the considered bottlenecks can be minimized. We formulate this as an optimization problem with a set of constraints, each representing a temporal range in which a participating driver can start to commute. Our objective is to minimize the combined peak volumes of all bottleneck road segments. We provide a precise definition to this problem and propose an integer linear programming (ILP) approach to solve this problem. The ILP formulation proposed here uses two types of constraints to direct the optimization: (1) the window constraint, which corresponds to each driver's temporal flexibility, and (2) the capacity constraint, which caps each bottleneck's peak usage. This approach is analyzed through simulations on real-world map data of Pittsburgh, Pennsylvania. Compared to a baseline case where each vehicle departs at its earliest time and a randomized approach using a random departure time, our scheduling approach is found to reduce the mean travel time by 15 to 20%. We also explored the impact of different penetration rates, i.e., percentages of vehicles participating in the scheduling service, and found that the improvement is still substantial even with just 40% of adoption.

Introduction

Traffic congestion in urban areas is a serious problem, resulting in significant economic costs for drivers through wasted time and fuel, and environmental cost through increased vehicle emissions. The primary cause of traffic congestion is limited roadway capacity, which is highlighted during the rush hours when the number of vehicles on the road can exceed the capacity of "bottleneck" road segments. One straightforward approach to addressing this problem is to increase infrastructure capacity through road building. However, studies on traffic and transportation have shown that widening or building more roads might not be the best way to alleviate congestion and could in some cases create more traffic (Downs 1962; Duranton and Turner 2011). Even when the capacity building does help, the construction could be infeasible due to physical constraints and the costs could be prohibitive. For these reasons, we believe that the better way to create more capacity should be through better control and coordination and not infrastructure building.

Due to a large number of vehicles traveling during peak rush hours, roadways are especially susceptible to congestion during this period. Over the past decade, major cities are increasingly turning to a demand management approach, which regulates the number of vehicles on the road to relieve the traffic congestion (Meyer 1999; Pluntke and Prabhakar 2013). Most of these existing demand management strategies rely on road pricing or usage restriction; however, these measures usually need infrastructural supports and law enforcement efforts and are thus not always feasible.

In this paper, we propose a scheduling-based approach to coordinate the road usage and relieve traffic congestion. This approach explores the possibility of reducing peak traffic volumes by scheduling vehicles' departure times according to drivers' self-reported departure time windows. For each participating driver, we assume that we could collect this driver's origin, destination, and a time window specifying feasible departure time. Given inputs from all participating drivers, we would suggest a personalized schedule for each driver to depart. The objective of our scheduling algorithm seeks to minimize the combined peak congestion levels at all identified bottleneck road links.

To demonstrate the potential of this demand shifting approach, we provide a corresponding integer linear programming (ILP) model. This model uses a time-indexed formulation and utilizes two types of constraints: 1) the window constraints, and 2) the capacity constraints. The window constraints specify the temporal flexibility that the participating drivers have for their times of departure. The capacity constraints allow us to identify the maximal number of vehicles that simultaneously arrive at the considered bottleneck segments. To analyze the effectiveness of our proposal, we identify several main arterial roads of Pittsburgh as our bottlenecks and assume that each vehicle may pass through several bottlenecks according to their chosen routes. As a comparison, we consider a baseline strategy in which drivers depart at their earliest possible departure time and a simple randomized approach (each driver would choose a departure

time uniformly randomly within the time window).

We evaluate the effectiveness and the efficiency of these approaches by measuring the mean travel time (MTT) and the computational cost, under several different test conditions. Across all comparisons, the scheduling approach achieves significant performance advantages, ranging from 15 to 20% in travel time reductions. Our approach does not need 100% adoption to work; in our experiments, we show that the system congestion begins to improve even with only 20% adoption rate. Our contributions can be summarized as follows:

- We define a new scheduling problem that realizes demand shifting to reduce peak traffic volumes.
- We propose a ILP model with iterative travel time update to solve the proposed scheduling problem.
- We evaluate performance of the ILP model through a detailed simulation. Empirical results show that our schedule-driven approach reduces delay by 20% at most.

The remainder of the paper is organized as follows. First, we briefly review some existing regulating methods for relieving traffic congestion. Following that, we describe the setting considered and provides a formulation to the problem. An ILP-based method is then proposed. Following that, we present the results from simulations and discuss the effectiveness of the proposed approach. We then look at some related computational approaches from the literature. Finally, we conclude this paper.

Traditional Traffic Engineering

Since traffic congestion has been a substantial problem in our society, researchers from different fields such as traffic engineering (Strickland and Berman 1995) and economics (Arnott, De Palma, and Lindsey 1990) have been persistently working to eliminate the cost of congestion. Current approaches to congestion follow three basic paradigms:1) capacity addition, 2) road rationing, and 3) congestion pricing.

Capacity addition solves congestion from the supply side. Increasing the number of lanes and deploying new public transit services are two approaches of capacity addition in transportation networks. However, there are several challenges with increasing the capacity of road networks. For example, the fundamental law of expressway congestion (Downs 1962; Duranton and Turner 2011) claims that the traffic volume of an expressway increases with additional road capacity, and construction of new capacity does not reduce congestion for existing commuters in the long run.

From the demand side, road rationing and congestion pricing (Liu, Yang, and Yin 2014; Han, Yang, and Wang 2010; Viegas 2001) are two other approaches for alleviating traffic congestion. For road rationing, certain types of vehicles are prohibited from being driven on certain days. Similarly, congestion pricing means that vehicles are charged for the usage of the road. However, both approaches run into public resistance due to cost of deployment and unfair pricing. For example, New York City abandoned its congestion pricing policy for Manhattan after encountering significant opposition from neighboring suburbs, who considered the scheme to be unfair.

The above demand-side congestion mitigation methods modify user behavior by disallowing or restraining drivers from using the road. On the other hand, the schedule-driven approach we propose in this paper exploits each driver's commute flexibility to search for departure times that disperse congestion. Compared to other methods for demand management, the proposed approach is more closely aligned with human behaviors and easily applied in cities due to the development of mobile computing (Thiagarajan et al. 2009) and crowd sourcing applications(Pluntke and Prabhakar 2013).

Demand Shifting Traffic Scheduling Problem

In this section, we give a precise definition to the scenario that we consider in this paper. This scenario comprises a set of commuters (or vehicles) $\mathcal{K}$ and a set of bottleneck road segments $\mathcal{B}$. Each vehicle $k \in \mathcal{K}$ has a specific route connecting its starting location and its final destination. When vehicles are traveling from the starting location to the destination during rush hours in the morning (and in reverse direction in the evening), most vehicles will pass through several popular road sections and accumulate significant volumes at these sections. We refer to these road sections as bottlenecks (Helly 1900). For example, for the Pittsburgh area, the Fort Pitt tunnel, connecting downtown and the suburban South Hills area, is one of the most important bottleneck. When the amount of traffic exceeds the capacity of the bottleneck, traffic congestion will emerge and delay will be incurred. In the following sections, we consider a practical setting in which multiple bottlenecks are considered.

Each vehicle k is assumed to have some flexibility as to departure time. By exploiting the collective flexibility of all drivers, vehicles can be directed to arrive at bottlenecks at different times and alleviate traffic congestion without constructing new infrastructure. In contrast to the concentrated traffic volume that would pour into the bottlenecks during the short periods of peak hours, scheduling vehicles based on drivers' demands spreads the commute patterns of vehicles across a longer duration, and thus vehicles encounter less competition for the road capacity.

Let $K = |\mathcal{K}|$ represent the number of vehicles that must traverse multiple bottleneck road segments (e.g., the Fort Pitt tunnel) during the rush hours. Each vehicle k has:

- A time window $W_k = [e_k, l_k]$ during which the vehicle k must depart from its starting point.
- A set of bottlenecks $\mathcal{B}_k$ that vehicle k needs to pass through sequentially. (Note that $\mathcal{B} = \cup_k \mathcal{B}_k$).
- A specified route connecting its starting location, bottlenecks in $\mathcal{B}_k$ and its final destination.
- A set of estimated travel time $\{d_b^k\}_{b \in \mathcal{B}_k}$, each corresponding to an estimate for traveling from k's starting point to a bottleneck b. d_b^k will be updated according to change of traffic.

In the following sections, we assume d_b^k is updated iteratively and the most recent d_b^k is applied. In addition, assume

that time horizon T_{rush} containing all W_k is discretized and indexed by the multiples of a basic time unit ΔT. We then round down the W_k to the nearest time index within T_{rush}.

The goal is to form a schedule $\mathbf{S} = (s_1, s_2, \ldots, s_K) \in \mathbb{N}^K$ that amortizes the vehicle passages of the bottlenecks, where each s_k specifies the departure time for the commuter k. We define the number of vehicles that arrive at the bottleneck as the *traffic volume*. The objective is then to reduce the traffic volume per unit time ΔT at the bottlenecks while satisfying each commuter's departure window constraint. Note that once we have decided an $\mathbf{S} = (s_1, s_2, \ldots, s_K)$, we can estimate for each vehicle k, the arrival time at a bottleneck $b \in \mathcal{B}_k$ as $s_k + d_b^k$.

In the following sections, we will formulate the problem in terms of $\mathbf{S}$. A valid (feasible) schedule $\mathbf{S}$ must have the following properties:

- Each element of $\mathbf{S}$ should be a valid **departure time**. The domain of s_k should be the discretized indices to the assumed time horizon, i.e., $t = 1, 2, \cdots \in T_{rush}$. Note that $T_{rush} = \cup_k W_k$.
- Each s_k should respect commuter k's **window constraint** W_k. Assuming the consecutive time slot indexes covered by W_k are the integers, $e_k, \ldots, l_k$, then the following constraints must be satisfied:

$$e_k \leq s_k \leq l_k, k = 1, \ldots, K. \tag{1}$$

Given the above descriptions of valid schedules, the target optimization problem is to minimize the quantity C_b at each bottleneck, where

$$C_b = \max_t \sum_{i=1}^{K} \mathbf{1}(s_k + d_b^k = t), \tag{2}$$

t ranges over T_{rush} and $\mathbf{1}(\cdot)$ is the indicator function. By minimizing C_b, we lower the number of vehicles that is scheduled to travel through the bottleneck segment b simultaneously, and the fewer the number of vehicles that arrive simultaneously, the shorter the delay in traversing through the bottleneck segment. Formally, we define this problem as the following:

Definition 1. *Given a set of vehicles and a set of bottlenecks, the Demand Shifting Traffic Scheduling Problem (DSTSP) is to search for a schedule that minimizes the maximum of the number of simultaneous arrival vehicles at each bottleneck.*

Figure 1 illustrates the overall picture of our setting. In this example, we have four vehicles with their preferred time windows W_k. The task is to specify the value of s_k so as to determine the exact departure time for each vehicle that satisfies the above specified definition.

Schedule-Driven Demand Shifting

To relieve traffic congestion, the departure times of all drivers must be diverse. Therefore, we seek a feasible schedule that minimizes the upper bound of the constraint (2) at each bottleneck. Through the schedule, the number of vehicles that arrive at the bottleneck simultaneously is decreased. In this section, we propose an ILP model to solve the DSTSP.

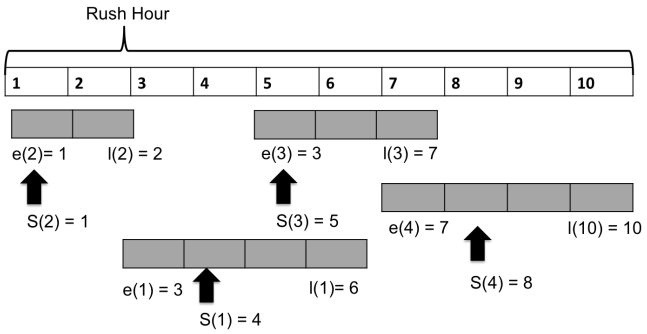

Figure 1: A four-vehicle case for the DSTSP

Objective Function

Traffic congestion in a transportation network can be quantified in terms of the statistics of the arrival processes of the network bottlenecks. These statistics determine the distributions of congestion level and waiting time at each road segment. It is evident that desirable scheduling is associated with a small mean and variance of delay at each bottleneck. In queueing theory, the delay is proportional to the arrival rate of incoming traffic according to Little's law. C_b could be seen as upper bound of arrival rate. However, it is usually difficult to express the objective by using a single figure for optimization.

A convenient alternative is to measure congestion in terms of average traffic carried by each bottleneck. More precisely, we assume that the arrival rates at each bottleneck change due only to schedule update, and we measure congestion on bottleneck b via the upper bound C_b. An useful expression $\sum_{b \in \mathcal{B}} D_b(C_b)$,where D_b is a function of arrival rate, is often appropriate as an objective for optimization. A frequently used formula, average link utilization, is $D_b(C_b) = \frac{C_b}{U_b}$, where U_b is the physical capacity of bottleneck b. In the following sections, we assume the capacity of each bottleneck is same, and the objective can be simplified to $\sum_{b \in \mathcal{B}} C_b$.

ILP Model

This section presents an ILP model for the DSTSP. The model uses a binary variable x_t^k to denote which starting time s_k is selected, where $s_k = t$ is equivalent to $x_t^k = 1$. In order to convert the DSTSP defined previously into an ILP formulation, we introduce the following **capacity constraint**,

$$\sum_{i=1}^{K} \mathbf{1}(s_k + d_b^k = t) \leq C_b, \tag{3}$$

where now C_b upper bounds of the number of vehicles that arrive at bottleneck b at time t simultaneously. With this an ILP model is formulated as follows:

Table 1: Problem Variables

Param.	Description
$\mathcal{K}$	a set of vehicles
$\mathcal{B}_k$	a set of bottlenecks that vehicle k passes through
$\mathcal{B}$	all bottlenecks in the city ($\mathcal{B} = \cup_k \mathcal{B}_k$)
$\mathcal{W}_k$	time indices that vehicle k is willing to depart
e_k, l_k	$\mathcal{W}_k = \{t \in \mathbb{N} : e_k \leq t \leq l_k\}$
d_b^k	the time that vehicle k takes to bottleneck $b \in \mathcal{B}_k$
Var.	**Description**
C_b	congestion level for bottleneck $b \in \mathcal{B}$
x_t^k	time-indexed variable for driver k

$$\min \quad \sum_{b \in \mathcal{B}} C_b \tag{4}$$

$$\text{s.t} \sum_{\left\{ \substack{k:k\in\mathcal{K}, b\in\mathcal{B}_k, \\ e_k + d_b^k \leq t \leq l_k + d_b^k} \right\}} x_{t-d_b^k}^k \leq C_b \ , \qquad \forall b \in \mathcal{B}, t \in T_{rush} \ , \tag{5}$$

$$\sum_{t \in \mathcal{W}_k} x_t^k = 1, \quad x_t^k \in \{0,1\} \ , \qquad \forall k \in \mathcal{K}. \tag{6}$$

Constraints (6) ensure that the window constraints (1) are fulfilled, constraints (5) correspond to the capacity constraint defined above, while the objective (4) minimizes the summation of arrival rate at each bottleneck.

Lower Bound Cut

To speed up the process of solving the ILP model, we introduce a cut constituted of $C_1 \cdots, C_{|\mathcal{B}|}$ to eliminate feasible solutions. Since the lower bound constituting of C_b can be obtained by solving each bottleneck separately, we introduce the cutting plane

$$\sum_{b \in \mathcal{B}} C_b^* \leq \sum_{b \in \mathcal{B}} C_b, \tag{7}$$

where C_b^* is the optimal solution for each bottleneck independently. C_b^* can be acquired by solving a single bottleneck under assumption that the schedules of different bottlenecks are independent. According to our results, adding this cut improves running time by 50% for complex instances, i.e, more bottlenecks. The running time of solving single bottleneck is far shorter than solving multiple bottlenecks. It only casts a small amount of effort with respect to solving the DSTSP. The following results are presented with applying the cut.

Iterative Travel Time Update

Since the $\{d_b^k\}_{b \in \mathcal{B}_k}$ are affected by the generated schedule, e.g., flow rate (vehicles/hour) is smaller after being scheduled, we propose an iterative method to update the travel times and the schedule, which was applied in route guidance system (Wunderlich, Kaufman, and Smith 2000). The concept is illustrated in Figure 2. Combining a schedule calculation with a traffic simulation in an iterative manner, we can revise the schedule to align with the simulated travel time $\{\tilde{d}_b^k\}_{b \in \mathcal{B}_k}$ to each bottleneck. Once $\{\tilde{d}_b^k\}_{b \in \mathcal{B}_k}$ is approximately equal to $\{d_b^k\}_{b \in \mathcal{B}_k}$ in the model, then we terminate whole procedure and report the latest departure time $\mathbf{S}$ to users. The schedule that terminates iterations can be identified as a fixed point for DSTSP.

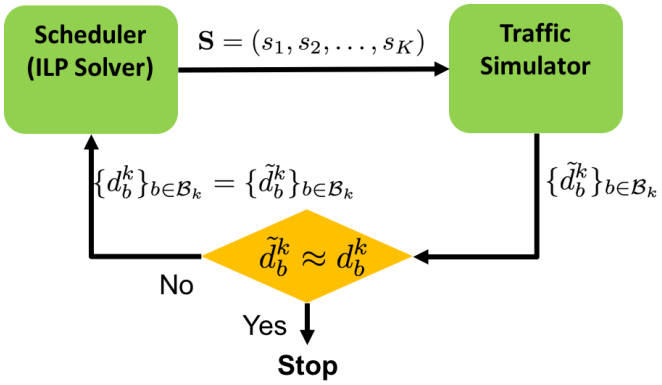

Figure 2: The iterative travel time update.

Experiments

This section presents experimental results of solving the DSTSP for the rush hours of Pittsburgh, PA. We considered the AM rush hours (7AM-10AM) where inbound traffic is dominant. The algorithms were implemented using C++ and CPLEX12.7 and the results were obtained on 64-bit machine with 3.8GHz×8 Intel Core i7-4790 and 16GB of RAM.

Instances

In this section, we explain the procedure that generates the problem instances for the experiments. First, PM traffic of Pittsburgh is simulated. We randomly generate origins nearby downtown and destination around the suburban area. Second, we pick N bottlenecks scattered around the boundary between downtown and suburban area, and each vehicle pass through a random number of bottlenecks from their origin to destination. We defined two different cases: high and low dependency cases. The vehicles in the high-dependency cases pass through more congested bottlenecks than the vehicles in the low-dependency cases, i.e., $E|\mathcal{B}_{h,k}| \geq E|\mathcal{B}_{l,k}|$. Therefore, the starting time of the vehicle in the high-dependency case may have larger impact on the rush hours since it affects more bottlenecks. The complexity of finding a solution for the high-dependency cases is also higher than the low-dependency cases. Third, a route including the assigned bottlenecks is pre-generated for each vehicle, and each vehicle has different departure flexibility, which is realized by assigning a randomly generated window W_k.

Simulation Setup

The microscopic traffic simulation platform SUMO (Simulation of Urban MObility) (Krajzewicz et al. 2012) is used to evaluate the performance impact of our schedule-driven approach. To provide a realistic setting, we import the Pittsburgh map data from the OpenStreetMap into SUMO as shown in Figure 3 and identify 3 to 7 arterial roads as the bottlenecks. The first step of the simulation is to generate problem instances according to the steps from the previous subsection. Subsequently, we provide the solver with the departure times, routes and temporal flexibility and generate the scheduled departure times. Finally, we run the simulation with the scheduled departure time and feed the new simulated travel time into ILP model for the next iteration of the travel time update. The flow chart of the simulation is illustrated in Figure 4. All simulations run for 4 hours of simulated time. To eliminate the effects of simulation start up and termination, we only report the mean travel time (MTT) of vehicles arriving their destination within the middle 2 hours. Results for a given experiment are averaged across ten simulation runs with different random seeds.

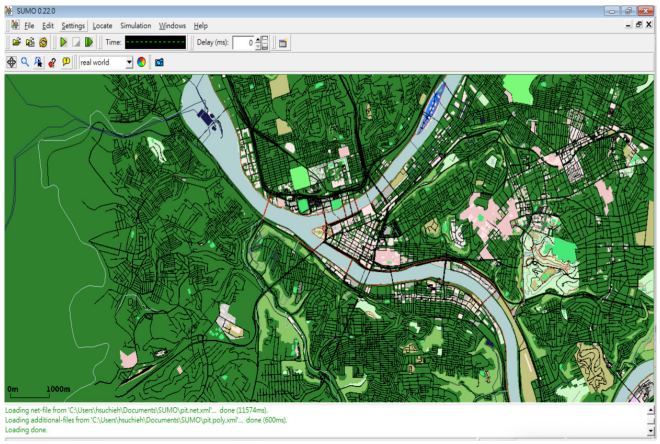

Figure 3: The simulator with a realistic Pittsburgh map.

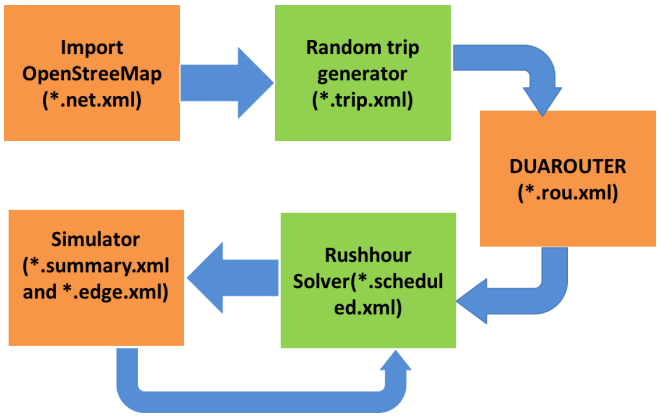

Figure 4: The simulation pipeline based on SUMO.

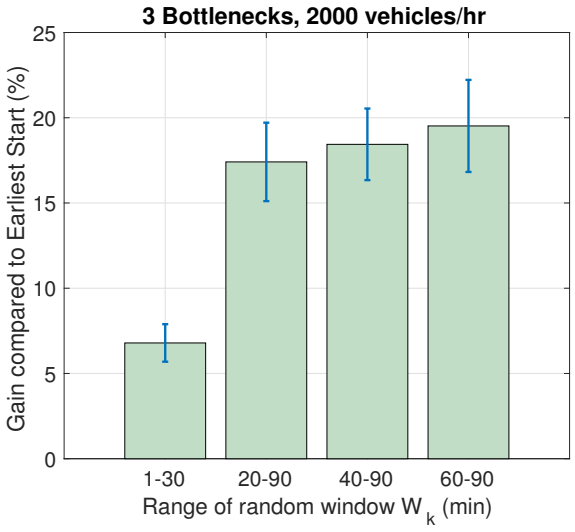

Figure 5: The effect of different window size.

Table 2: MTT comparison for different traffic pattern.

	High dep.,2000/hr		High dep., 4000/hr	
	MTT(s)	Iteration	MTT(s)	Iterations
Earliest Start	3226 ± 21	N/A	7418 ± 21	N/A
Randomized Start	3029 ± 16	N/A	7124 ± 33	N/A
Scheduled Start	2565 ± 11	4	6140 ± 15	3
	Low dep., 2000/hr		Low dep., 4000/hr	
	MTT(s)	Iteration	MTT(s)	Iterations
Earliest Start	1432 ± 25	N/A	5521 ± 27	N/A
Randomized Start	1450 ± 23	N/A	5234 ± 35	N/A
Scheduled Start	1391 ± 14	5	4674 ± 25	4

Urban Traffic Problem

We compare the proposed schedule-driven approach with a baseline case where each vehicle starts at its earliest start time e_k and a simple randomized approach (using a random departure time within each time window). For each scheduled vehicle, the chosen window size range is between 20 minutes and 90 minutes. We ran the test with $\Delta T = 30$ seconds on 5 bottlenecks. We tested two different traffic volumes: 2000 vehicles/hour and 4000 vehicles/hour vehicles and two different scenarios: high and low dependency.

Table 2 shows that MTT of the proposed and two other approaches running over two sets of randomly generated vehicles (i.e., 4000/hour and 2000/hour) and two different traffic patterns (i.e., the high-dependency and the low-dependency cases). In the high-dependency case with 2000 vehicles/hour, the schedule-driven approach has a smaller MTT compared to the earliest start case. When traffic volume is increasing, the gap between two approaches become larger. Specifically, the MTT of the high-dependency case with 2000 vehicles/hour could be reduced by 700 seconds (26%). In the more congested case with 4000 vehicles/hour, the MTT reduction is still 17%. We can also observe that the performance of the randomized approach is usually in the

Table 3: Different number of bottlenecks and dependency with 4000/hour.

	High dependency				
	Earliest Start MTT(s)	Scheduled MTT(s)	Improv.(%)	Iterations	Avg. T_{cpu}(s)
3 bottlenecks	7148 ± 33	7025 ± 14	1.7%	4	421
5 bottlenecks	7418 ± 21	6140 ± 15	17.2%	3	1438
7 bottlenecks	7567 ± 35	6893 ± 31	8.8%	2	3378
	Low dependency				
	Earliest Start MTT(s)	Scheduled MTT(s)	Improv.(%)	Iterations	Avg. T_{cpu}(s)
3 bottlenecks	6423 ± 11	5529 ± 21	14.1%	4	62
5 bottlenecks	5521 ± 27	4674 ± 25	15.3%	4	163
7 bottlenecks	5606 ± 42	5595 ± 48	0.1%	4	387

middle of the baseline and the schedule-driven approach. On the other hand, the low-dependency cases achieve a smaller MTT reduction, which is approximately the same when the traffic demand level is low, and 13% when the traffic demand level is high. It is interesting to note that when the traffic demand level is high, the generated schedule is more effective in coordinating the vehicles. In addition, when the cross traffic between bottlenecks is high, i.e., in high-dependency cases, schedule-driven approach works better as well.

The iterative travel time update is applied on the schedule-driven approach to update $\{d_b^k\}_{b\in\mathcal{B}_k}$. We terminate the procedure if the simulated travel time is within 5% of previous simulated travel time. For high traffic demand, it takes less iterations to terminate since the congestion from previous bottlenecks decreases flexibility of optimizing current schedule. Contrarily, applying iterative update on low demand traffic is able to improve the performance further.

In Figure 5, we study how different window size affects the MTT. Basically, the larger the average temporal flexibility, the larger improvement of MTT scheduling can provide. It gives our model more room to coordinate departure times. We assume that the maximum window size is 1.5 hrs and adjust the lower bound of window size. By increasing the average window size of all vehicles, the 3 bottlenecks with 2000 vehicles/hour can achieve 20% improvement at most.

In Table 3, we profile the schedule-driven approach on different numbers of bottlenecks with 4000/hour volume. For high dependency, the instance of 5 bottlenecks has the highest improvement, while the improvement in 3 and 7 bottlenecks problems is lower. There are two reasons: (1) the traffic in the 7 bottleneck case becomes sparse. The benefit of scheduling vehicles is decaying. (2) 3 bottlenecks case is the most congested scenario. The delay caused by previous bottlenecks is propagated to the latter one. The benefit of scheduling is marginal. The benefit should be enlarged if we provide a more accurate estimated d_b^k. Moreover, the trend of low dependency is decreasing with the increased number of bottlenecks. The instances of 3 and 5 bottlenecks are in the region that scheduling is beneficial.

We also measure the speed of solving the ILP model and find that it depends on two important factors: (a) the number of bottlenecks and (b) the dependency among bottlenecks, which is the average number of bottlenecks each vehicle passes through. For instance, solving the high dependency with 7 bottlenecks takes average 3378 seconds ~ 1 hour for each iteration. However, solving the low dependency with 3 bottlenecks takes only average 62 seconds for each iteration. If the dependency is small, the solver is able to solve the ILP model promptly. The CPU time can be understood in the following way: If it takes too long to generate a solution, it would be impractical as a daily notification service. For instance, drivers may need to change their routes and flexibility just prior to rush hour, and one hour should be a reasonable buffering time.

Effect of Penetration Rates

Understanding the performance under different penetration rates is an essential issue for deploying the proposed methods under realistic scenarios. In the results presented thus far, we have assumed that all vehicles are scheduled by the proposed approach. However, it may be difficult in practice to achieve such a high penetration rate. Figure 6 presents the MTT of the schedule-driven approach under different penetration rates. We assume that the window information of those scheduled vehicles are known. In order to help the scheduled vehicles to avoid rush hours, we assume that they are able to access the traffic information as well. It can be estimated by monitoring the bottlenecks daily. Then the vehicles are scheduled to avoid the estimated rush hours with other unscheduled vehicles by modifying the constraint (5) to $\sum_k x^k_{t,sched.} + \sum_k x^k_{t,unsched.} \leq C_b$. For example, we consider a case of single bottleneck. Penetration rate 40% indicates that 40% of vehicles are scheduled according to their demand and the other 60% start at their e_k. The results show that the proposed algorithm can still provide a considerable 16% improvement compared with the baseline case for a single bottleneck when the penetration rate is 50%. The improvement drops to 12% at a penetration rate of 35% which still represents a reasonable reduction in overall congestion. If drivers provide larger windows for their departure times, we believe performance at lower penetration rates will improve further.

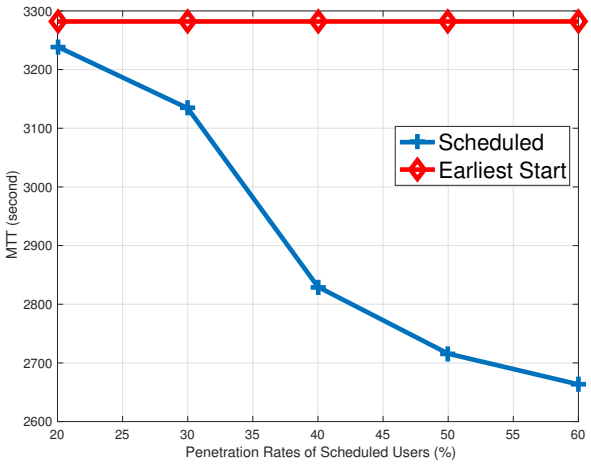

Figure 6: Performance comparison between the schedule-driven approach and the baseline case under different penetration rates

Related Works

The traffic scheduling problem defined in this work is a new problem due to its objective of reducing the upper bound C_b of the capacity constraint. If the upper bound is fixed and without the variable departure times of each user, we can view the problem as an evacuation planning problem. In macroscopic evacuation planning, it models the movement of evacuees as flows in the evacuation graph. The evacuation planning is related to maximum dynamic network flow problem (MDFP) (Ford Jr and Fulkerson 1958; Burkard, Dlaska, and Klinz 1993). MDFP computes the maximum flow over one time period and then determine the paths on the network through flow decomposition. However, MDFP and related problems do not consider *variable departure times* for each flow to compute maximum flow in the context of evacuation. (Even, Pillac, and Van Hentenryck 2015) introduced the concept of convergent evacuation plans, which assigns an evacuation routes to each residential zone. The authors defines a time-expanded graph (Köhler, Langkau, and Skutella 2002) to tackle temporal aspects of evacuation planning, which could correspond to the variable departure times of traffic scheduling problem. However, getting an optimal solution of this problem is restrictive because of its scalability issues. Using it to solve the traffic scheduling problem is even worse, because all drivers have a window, which increases the number of nodes in the graph by at least one order of magnitude. In addition, the graph of the traffic scheduling problem is composed of a few congested bottlenecks and a large number of source nodes that represent the starting locations of drivers. According to commuting behavior, it is preferable to model it as a scheduling problem since the drivers usually intend to select main roads for commuting.

The traffic scheduling problem can be also viewed as a dynamic job shop scheduling problem (Ramasesh 1990) when the vehicles are jobs and the bottleneck is machine. The single machine problem usually assumes the capacity of the machine is constant at a finite value. However, the upper bound of road capacity that leads to free flow is difficult to determine since it depends on the chosen speed and the definition of congestion level. If the upper bound can be minimized and simultaneously fulfill the time requirement of each user, the travel time would be reduced as much as possible and also be able to incorporate more traffic uncertainty.

Developing control strategies to reduce traffic congestion in the research community has been discussed extensively. This includes traffic light control (Boillot, Midenet, and Pierrelée 2006), contraflow (Kim, Shekhar, and Min 2008) and rerouting of cars (Li, Mirchandani, and Borenstein 2009). Compared to those works, this paper considers a more realistic and simpler approach, which only takes user's demand into account without adding controlled infrastructures. Moreover, this approach can be integrated with other approaches to achieve a synergistic effect. To us, the traffic scheduling problem falls into a category of traffic regulation design.

Conclusions

In this work, we describe a demand shifting traffic scheduling problem for relieving traffic congestion. A schedule-driven approach is developed to solve it under the realistic map data of Pittsburgh. Since the level of congestion directly relates to the upper bound of the capacity constraints, we formalized the Demand Shifting Traffic Scheduling Problem (DSTSP) and presented an ILP model for finding a schedule minimizing the arrival rate at each bottleneck based on each driver's temporal flexibility. The proposed ILP model minimizes the bound to spread vehicles' departure times properly. We demonstrate that the approach improves travel time and throughput substantially compared with the baseline case. Furthermore, our solutions provide substantial gain under small penetration rates.

Future work will focus on extending the models to integrate route decisions and build a more accurate model on how delay caused by previous bottlenecks propagate. Route decisions would give additional flexibility to the optimization algorithm to exploit spatial flexibility. Accurate delay model can improve the performance further for larger traffic volumes.

References

Arnott, R.; De Palma, A.; and Lindsey, R. 1990. Economics of a bottleneck. *Journal of urban economics* 27(1):111–130.

Boillot, F.; Midenet, S.; and Pierrelée, J.-C. 2006. The real-time urban traffic control system cronos: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies* 14(1):18–38.

Burkard, R. E.; Dlaska, K.; and Klinz, B. 1993. The quickest flow problem. *Zeitschrift für Operations Research* 37(1):31–58.

Downs, A. 1962. The law of peak-hour expressway congestion. *Traffic Quarterly* 16(3).

Duranton, G., and Turner, M. A. 2011. The fundamental law of road congestion: Evidence from us cities. *The American Economic Review* 2616–2652.

Even, C.; Pillac, V.; and Van Hentenryck, P. 2015. Convergent plans for large-scale evacuations. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15)*.

Ford Jr, L. R., and Fulkerson, D. R. 1958. Constructing maximal dynamic flows from static flows. *Operations research* 6(3):419–433.

Han, D.; Yang, H.; and Wang, X. 2010. Efficiency of the plate-number-based traffic rationing in general networks. *Transportation Research Part E: Logistics and Transportation Review* 46(6):1095–1110.

Helly, W. 1900. Simulation of bottlenecks in single-lane traffic flow.

Kim, S.; Shekhar, S.; and Min, M. 2008. Contraflow transportation network reconfiguration for evacuation route planning. *Knowledge and Data Engineering, IEEE Transactions on* 20(8):1115–1129.

Köhler, E.; Langkau, K.; and Skutella, M. 2002. Time-expanded graphs for flow-dependent transit times. In *European Symposium on Algorithms*, 599–611. Springer.

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

Li, J.-Q.; Mirchandani, P. B.; and Borenstein, D. 2009. Real-time vehicle rerouting problems with time windows. *European Journal of Operational Research* 194(3):711–727.

Liu, W.; Yang, H.; and Yin, Y. 2014. Traffic rationing and pricing in a linear monocentric city. *Journal of Advanced Transportation* 48(6):655–672.

Meyer, M. D. 1999. Demand management as an element of transportation policy: using carrots and sticks to influence travel behavior. *Transportation Research Part A: Policy and Practice* 33(7):575–599.

Pluntke, C., and Prabhakar, B. 2013. Insinc: A platform for managing peak demand in public transit. *JOURNEYS, Land Transport Authority Academy of Singapore*.

Ramasesh, R. 1990. Dynamic job shop scheduling: a survey of simulation research. *Omega* 18(1):43–57.

Strickland, S. G., and Berman, W. 1995. Congestion control and demand management. *Public roads* 58(3).

Thiagarajan, A.; Ravindranath, L.; LaCurts, K.; Madden, S.; Balakrishnan, H.; Toledo, S.; and Eriksson, J. 2009. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, 85–98. ACM.

Viegas, J. M. 2001. Making urban road pricing acceptable and effective: searching for quality and equity in urban mobility. *Transport Policy* 8(4):289–294.

Wunderlich, K. E.; Kaufman, D. E.; and Smith, R. L. 2000. Link travel time prediction for decentralized route guidance architectures. *IEEE Transactions on Intelligent Transportation Systems* 1(1):4–14.