Traffic Light Scheduling, Value of Time, and Incentives [*]

Argyrios Deligkas[1], Erez Karpas[2], Ron Lavi[2], Rann Smorodinsky[2]
[1] Leverhulme Research Centre, University of Liverpool
[2] Industrial Engineering and Management, Technion
argyrios.deligkas@liverpool.ac.uk, karpase@technion.ac.il, rann@technion.ac.il, ronlavi@ie.technion.ac.il

Abstract

We study the intersection signalling control problem for cars with heterogeneous valuations of time (VoT). We are interested in a control algorithm that has some desirable properties: (1) it induces cars to report their VoT truthfully, (2) it minimizes the value of time lost for cars waiting at the intersection, and (3) it is computationally efficient. We obtain three main results: (1) We describe a computationally efficient heuristic forward search approach to solve the static problem. (2) We extend the solution of the static problem to the dynamic case. We couple our algorithm with a carefully designed payment scheme which yields an incentive compatible mechanism. (3) We describe simulation results that compare the social welfare obtained by our scheduling algorithm, as measured by the total value of waiting time, to the social welfare obtained by other intersection signalling control methods.

1 Introduction

Traffic congestion cost Americans $124 billion annually in direct and indirect losses and this number will rise to $186 billion in 2030 [INR, 2014]. This claim is supported by data [INR, 2014; Christidis and Rivas, 2012], which forecast that cumulative costs between 2013 and 2030 across four countries – UK, France, Germany, and USA – will sum up to $4.4 trillion. This cost accounts for fuel value, indirect costs, and the value of wasted time. As [INR, 2014] points out, the largest of the three is undoubtedly the opportunity cost of the time wasted whilst delayed in congested traffic. In this paper, we study this problem when it occurs on an intersection and we adopt an economic approach. We are interested in a control algorithm that has some desirable properties: it minimizes the value of time lost for cars waiting at the intersection, it induces cars to report their value of time (herein after, VoT) truthfully, and it is computationally efficient.

Using various economic mechanisms to reduce congestion is becoming increasingly popular. In Israel, there is a Fast Lane on one of the entrances to Tel-Aviv, allowing faster access to the city to drivers who are willing to pay. London has a Congestion Charge for driving a vehicle within the charging zone. Many mega cities in China widely implement a vehicle ownership control program, releasing a fixed number of vehicle license plates every month, priced and allocated via various economic mechanisms. Singapore also has a vehicle quota system, and many more examples exist. Compared to these existing mechanisms, we propose a dynamic pricing mechanism which is advantageous since the resulting tolls depend on the actual route being taken, and driver payments increase when using more congested junctions.

In a seminal paper, [Becker, 1965] introduces the concept of *value of time*. Since then, economists and civil engineers tried to estimate the VoT for commuters [Abou-Zeid *et al.*, 2010; Bento *et al.*, 2015], and proposed and *applied* approaches based on VoT in order to decrease the VoT wasted on congested traffic on highways by pricing fast-lanes on them; see [Tsekeris and Voß, 2009; de Palma and Lindsey, 2011] for a survey. Yet, for intersection management, theoretical models that take VoT into account have appeared only recently [Dresner and Stone, 2006; Schepperle *et al.*, 2007; Schepperle and Böhm, 2008; Raphael *et al.*, 2015; Sharon *et al.*, 2017]. A main reason for this is the fundamental difference between intersections and highways, as crossing an intersection requires *coordination* among cars arriving at different lanes. Hence, economic mechanisms for junction control and for highway control are fundamentally different.

[Dresner and Stone, 2004; 2005] were the first who studied the intersection management problem from the Multi Agent System point of view and provided a framework for coordinating cars in the intersection. Although their framework can be the base for intersection management in the future, it requires every car to be *autonomous*. Another solution to intelligent intersection management, which does not require massive infrastructure changes, is SURTRAC [Smith *et al.*, 2013]. SURTRAC uses cameras to measure traffic flow, and adaptively changes the traffic light schedule to maximize flow. However, it does not take VoT into consideration.

If heterogeneous VoT is taken into account, a second issue arises from the introduction of agents in the intersection model; *incentive compatibility* (IC). IC guarantees that no car can benefit from misreporting its true VoT in order to cross the intersection faster. Whether or not IC is a crucial requirement from a system is a subjective issue. We strongly believe IC is critical for the obvious reason — a mechanism that is

[*] This research was supported by the Technion Autonomous Systems Program; Ron Lavi was supported by a grant from the MINERVA foundation (ARCHES 2011) and by the ISF-NSFC joint research program (grant No. 2560/17).

not IC may be optimal in terms of the reported VoT but not the actual VoT. IC is a central concept in mechanism design in general and to our problem in particular. We combine techniques from planning and scheduling with techniques from game theory to construct a computationally-efficient traffic light scheduling algorithm that minimizes the wasted VoT and incentivizes the cars to report their true VoT.

Our main contributions are threefold. First, we describe a computationally efficient heuristic forward search approach to solve the static problem (no new incoming cars). Simulation results show that this method is significantly faster than the dynamic-programming approach [Bellman, 1954] to solve the static problem. This feature is crucial if we ever hope to apply our mechanism in the real world, as it must be able to control a traffic light in real time. Second, we extend the solution of the static problem to the dynamic case, where new cars do arrive at the intersection during execution. We couple our algorithm with a payment scheme which yields an IC mechanism, guaranteeing that it is in the best interest of each car to truthfully report its VoT. Third, we describe simulation results that compare the social welfare (total value of time wasted) obtained by our scheduling algorithm to the social welfare obtained by other intersection signalling control methods. We also give a theoretical impossibility result, showing that no algorithm can obtain a close-to-optimal social welfare in the worst-case, and in addition observe that the welfare that our algorithm produces is on the Pareto-frontier. To the best of our knowledge, this is the first approach that satisfies all the above properties.

Further Related Work. Following [Dresner and Stone, 2004], several papers studied intersection management from a different aspect than ours, adopting the *reservation-based* approach. This approach requires every car to be autonomous, hence it is not supported by the current infrastructure yet. For example, [Vasirani and Ossowski, 2008; 2010] study reservation based (RB) models on networks of intersections, [Vasirani and Ossowski, 2012] study an RB approach and implement a combinatorial auction in a single intersection, [Levin and Boyles, 2015] apply first-price auctions in RB intersections to increase efficiency in the dynamic traffic assignment problem, and [Levin *et al.*, 2017] try to optimize an RB approach using some heuristics without dealing with IC constraints. Another approach, similar to ours, studies intersections controlled by traffic lights. [Isukapati and Smith, 2017] study traffic signal control that takes into account heterogeneous VoT, but the proposed approach is not IC. [Lloret-Batlle and Jayakrishnan, 2016] study traffic signal control that incorporates user heterogeneity on value of delay savings, which is different than VoT, and allow monetary transfers among the cars. They propose a mechanism that is economically efficient, budget balanced and Pareto efficient, but not IC. [Xie *et al.*, 2014] consider pedestrians in addition to cars. [Pourmehrab *et al.*, 2017] study signalised intersections and optimize the time spent on the intersection. [Xie *et al.*, 2012] study real-time dynamic flow optimization of vehicle traffic through a network of signalized intersections. [Gulić *et al.*, 2016] follow an automated planning approach for signalized intersections and [Vallati *et al.*, 2016] introduce a planning system for signalised traffic management in networks. However, none of these consider heterogeneous VoT.

2 The Intersection Model

The intersection. We focus on a single intersection with a finite set of incoming *lanes* L. A lane $l \in L$ can be either a single physical lane or a set of physical lanes that move together, e.g., two lanes for going straight and turning left controlled by a single light. Lanes l_1, l_2 are *interfering* if cars from l_1 interfere cars from l_2. A *light assignment* is a set of pairwise non-interfering lanes. A *schedule* σ is a sequence of pairs $\langle a, d \rangle$ meaning that the lanes in the light assignment a have a green light for a duration d.

The status of every lane in a light assignment is either `Move` or `Wait`. Hence, for every lane, a schedule can be decomposed into *phases*: a maximal period of time with the same status for the lane. When a lane l changes status from `Wait` to `Move`, there is a switching time δ until the first car from lane l enters the intersection. This accounts for the time spent on "yellow light", or the time needed for the lane to clear from cars of other lanes.

Cars. Every car i arrives at the intersection at some lane l. If there are k cars at lane l when car i arrives, its position is $k+1$. It crosses the intersection in position 0. We make two realistic simplifying assumptions. First, when a lane moves, every car needs the same time to advance one position. Second, every car needs the same time, t_c , to cross the intersection (i.e., to move from position 1 to position 0).

Each car is associated with a *private* value of time (VoT) v_i. The total cost of car i for waiting t time units in the intersection is $c_i(t) = v_i \cdot t$. A car is willing to pay up to $c_i(t)$ for saving t time units in crossing the intersection.

Mechanisms and Optimal Schedules. Our goal is to compute schedules that minimize the total cost of the cars. Formally, if $t_i(\sigma)$ denotes the time car i needs to cross the intersection under schedule σ, an *optimal schedule* σ^* minimizes $\sum_i v_i \cdot t_i(\sigma^*)$. Since values v_i are private, car i might report $b_i \neq v_i$ if this results in a schedule in which it crosses faster. We will motivate the cars to bid their true VoT by implementing a *mechanism*, which receives a bid vector b and produces a schedule $\sigma(b)$ and a *price vector* $p(b)$ where $p_i(b) \geq 0$ is the price that car i pays. We use $b = (b_i, b_{-i})$ where b_{-i} is the bids of all cars except i. The total utility of car i under schedule $\sigma = \sigma(b)$ is $v_i \cdot t_i(\sigma) + p_i(b)$. A mechanism is *incentive compatible* (IC) if no car i can decrease its total utility by reporting $b_i \neq v_i$. A car might refuse to participate in the mechanism, e.g., by not installing the app. We therefore additionally require *individual rationality* (IR) which means in our setting that $b_i = 0$ implies $p_i = 0$. With both IC and IR, a car can never benefit from not participating, or from participating but misreporting its true VoT.

3 Static Instance

We start with the case where no new cars arrive. We describe a polynomial-time computation of an optimal schedule for our model, and add payments to incentivize the cars to bid their true value of time. Throughout, we fix an arbitrary tie breaking rule among optimal schedules. This will be crucial for the dynamic case.

Computing an Optimal Schedule for our Model

Finding an optimal schedule for a given intersection can be seen as a deterministic planning problem. For a given intersection, a state s consists of the sequence of cars at each lane, as well as the current light assignment. Our convention is that the first car in each lane is the first in the sequence describing that lane. In the static case, cars only leave the intersection. Thus, our terminal states are those with no cars in the intersection. The actions in this planning problem are the possible light assignments. We define the successor state s', which is the result of applying light assignment a in state s, to be the state right after the next car crosses the intersection when the light assignment is a. More precisely, we compute the time t required for the next car in each lane in a to cross. If the chosen light assignment a is different from the previous one (the light assignment in s), the time t includes the switching time, and is $\delta + t_c$. Otherwise, the time is t_c. The transition then corresponds to setting the light assignment a for the duration of t. Multiple cars can cross if they cross at the same time; this will happen as we assume cars travel at the same speed.

The cost of such a transition is the total value of time of the cars in the intersection, which is the time t it requires multiplied by the total VoT of the cars in the intersection in s.

A sequence of transitions consisting of the same light assignment corresponds to a longer phase of the same light assignment. Thus, it is easy to see that a path through this state space corresponds to a valid schedule, and the sum of costs of the transitions in the sequence corresponds to the total costs incurred by the cars following the schedule. While not all possible schedules are covered by this search space, all *non-wasteful* schedules are. A schedule is non-wasteful if it does not contain some interval of time where no cars are currently crossing the intersection. Any optimal schedule must be non-wasteful, and so all optimal schedules can be found by searching through this state space.

Another optimization we make is to only choose among *maximal* light assignments. A light assignment a is maximal iff there is no feasible light assignment a' such that $a \subset a'$. It is easy to see that ignoring non-maximal light assignments never prunes an optimal solution, while decreasing the branching factor of the search space.

We bound the number of possible states in this search space. We assume the number of lanes in the intersection, k, is a constant, and that there are n cars, of which there are n_i cars in each lane i. Since cars in each lane cross in the order they are waiting in the lane, it is enough to count how many cars have crossed in each lane to know the exact state of the cars in the intersection. Therefore, the number of possible states of the cars is $\Pi_{i=1}^{k} n_i \leq n^k$. Since our state also includes the current light assignment, this number is multiplied by the number of possible light assignments, which is at most 2^k. Thus, the maximum number of states is $2^k n^k$.

This choice of state representation and transition function has several advantages. First, as we have seen, the size of the state space is polynomial in n, because time is neither part of the state nor of the actions. In other words, we do not choose a duration for each light assignment, which would have resulted in an infinite branching factor. Furthermore, this formulation can be extended to handle extra constraints imposed by the scheduler like bounds on the minimum and the maximum time of a phase, or the presence of pedestrians. Every pedestrian crossing can be modelled as a "pedestrian" lane with its own minimum phase, and interferes with the rest of the lanes of the intersection. The number of the pedestrians waiting to use the crossing can be counted via a camera and be treated as a single vehicle that wants to cross the intersection via the specific lane. The scheduler can assign a VoT for each number of pedestrians. In addition, pedestrians could actively participate in the bidding, to cross faster.

Lemma 1. *For any instance I and bidding vector b we can compute the optimal schedule, w.r.t. b, in polynomial time.*

While it is possible to use blind search or dynamic programming to find an optimal schedule, it might have to examine the entire state space. To avoid this, we design an admissible heuristic for use with A^* [Hart *et al.*, 1968]. It is based on a relaxation, similar to delete-relaxation from automated planning [Bonet and Geffner, 2001], by assuming that it is possible to assign a green light to all lanes at once, and computing the total cost for all cars. This is easily computed by considering each lane in isolation. For each lane, we compute the time each car in the lane will cross, assuming this lane has a green light until all cars have crossed. If the lane in question has the green light in s already, the car at position i in the lane crosses at time $t_c \cdot i$. This car contributes $t_c \cdot i \cdot b_i$ to the heuristic value. If the lane in question has a red light in s, we must also account for the switching time, hence it contributes $(\delta + t_c \cdot i) \cdot b_i$. This heuristic is clearly admissible, and thus yields an optimal solution when used with A^*. Our empirical evaluation shows that this yields significant time savings in practice.

VCG payments

The polynomial-time computation of the optimal schedule allows us to use VCG payments: p_i for car i is defined as the difference between the cost incurred by the rest of the cars under the optimal schedule $\sigma^*(I)$ and the cost of the optimal schedule when car i is not present, $\sigma^*(I_{-i})$. However, in I_{-i} car i cannot be simply removed from the intersection, since this would change the costs of the other cars as they will advance one position in the lane, violating the no-externalities assumption of the VCG framework. We thus define I_{-i} as the instance where we replace b_i with zero. This way, the cost of any car $j \neq i$ does not change between I and I_{-i}, and VCG payments continue to satisfy IC and IR.

Myerson's payments

Myerson's payments [Myerson, 1981] yield another IC and IR mechanism which for the static case coincides with VCG. However, VCG can be used only for an optimal schedule while Myerson's method yields IC and IR payments for any *monotone* algorithm, i.e., any algorithm with the property that increasing one's bid must result in a weakly shorter crossing time. Formally, for any b_{-i}, if t is i's crossing time when the bid vector is $b = (b_i, b_{-i})$ and t' is i's crossing time when the bid vector is $b' = (b'_i, b_{-i})$, where $b'_i > b_i$, then $t' \leq t$. In the dynamic case, when the algorithm is not optimal, we use Myerson's method.

Lemma 2. *The optimal schedule is monotone.*

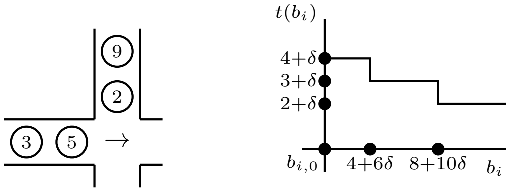


Figure 1: Left: An instance of a two-lane intersection with switching time $\delta < 0.1, t_c = 1$ and initially the horizontal lane is on status Move. The optimal schedule is $s, \downarrow_2, \downarrow_9, s, \rightarrow_5, \rightarrow_3$, where the arrow denotes the lane that moves and the subscript the car that crosses, and s denotes the switch. Right: The crossing times for the car with VoT 9, as a function of its bid. Using this, we compute Myerson's payment for this car.

In Myerson's method, fixing some monotone light-assignment algorithm, i's payment when the bid vector of the other cars is b_{-i} is as follows. Let $t_i(x)$ be the time car i crosses the intersection when the bid vector is (x, b_{-i}). Given b_i, recursively define a sequence of bids $0 = b_{i,0} < b_{i,1} < \ldots < b_{i,m} \leq b_i$ where $b_{i,j}$ is the minimal bid s.t. $t_i(b_{i,j}) < t_i(b_{i,j-1})$. Myerson's payment $p_i^M(b_i)$ is

$$p_i^M(b_i) = \sum_{j=1}^{m} b_{i,j} \cdot \big(t_i(b_{i,j-1}) - t_i(b_{i,j})\big). \qquad (1)$$

Lemma 3. *For any bid vector $b = (b_i, b_{-i})$ the number m is bounded by n, the number of cars in the intersection.*

This is because the number of cars present in the junction at i's crossing time when i's bid is $b_{i,j}$ strictly decreases as j increases; if i crosses faster, there exists another car that crossed before i in $b_{i,j-1}$ and crosses after i in $b_{i,j}$.

Assuming a finitely divisible value of time (say, up to cents), the naive way to compute Myerson's payments is to exhaustively search over all possible bids of car i (keeping the other bids fixed), computing $b_{i,m} \in [0, b_i]$ and $t_i(b_{i,m})$ for every possible m. A more efficient way is to use binary search, which we use in our experiments. The binary search can be thought of as iterating over a binary tree of ranges of bids. The root node is $[0, b_i]$, where b_i is the bid submitted by car i, and we start by computing $t_i(0)$ and $t_i(b_i)$. We then expand the tree as follows. To expand a node $[a, b]$, we compute $t_i(\frac{a+b}{2})$, and generate the children $[a, \frac{a+b}{2}]$ and $[\frac{a+b}{2}, b]$. The benefit comes from pruning any node $[a, b]$ where $t_i(a) = t_i(b)$, since we know that the crossing time would also be the same for any bid in the range $[a, b]$. The height of the search tree is at most $\log b_i$ and by Lemma 3 we have to find $m \leq n$ leaves of this tree.

Theorem 1. *The mechanism that implements the optimal schedule and uses Myerson's payments is IC, ex post IR, and requires polynomial time.*

4 Dynamic Instance

Next we study the dynamic version of our problem, where cars arrive online at the intersection and the mechanism has no knowledge about the cars that will arrive later.

As we will show, it is impossible to achieve optimality with no knowledge about the future. Thus, our goal is to design a time-efficient online IC mechanism that achieves low total cost for the cars. In this setting we cannot apply VCG payments in order to derive an IC mechanism, because mechanisms that do not implement optimal solutions cannot become IC by using VCG payments [Nisan and Ronen, 2001].

Thus, we take a planning and execution perspective, and propose two different approaches.We assume we have the optimal schedule for the cars that were in the intersection when the mechanism started. Whenever a new car arrives, we must decide whether to compute a new schedule immediately, or keep executing the current schedule. We consider the two extremes: never compute a new schedule until the current one has finished executing (termed *statically optimal*), and always compute a new schedule (termed *locally optimal*).

4.1 Statically Optimal Mechanism

Under the statically optimal mechanism, we compute a schedule for the cars in the intersection at time 0, and execute that schedule until it ends. When the schedule has finished, we compute an optimal schedule for the cars currently in the intersection, execute it until it finishes, and so on. If we complement this mechanism with the Myerson's payments, or VCG payments, we make it IC and IR.

Theorem 2. *The statically optimal mechanism is IC and IR.*

However, this is not always a good idea. Consider for example a two-lane intersection, where l_1 interferes with l_2. Assume that initially there are n cars in l_1 all of them with value zero and no cars in l_2. Clearly, the optimal schedule for this instance is to allow all cars in l_1 to cross without any interruptions. Suppose now that a car with value x arrives at time-step 1 at l_2. Then, it would have to wait for $(n-1) \cdot x$ until it crosses and its total cost would be $n \cdot x$. The optimal schedule in hindsight for this instance is to allow one car in l_1 to cross, then the car in l_2, and finally the rest of the cars in l_1. The optimal cost is x, so the statically optimal mechanism can perform arbitrarily badly.

4.2 Locally Optimal Mechanism

Under the locally optimal mechanism, we compute a new schedule whenever a new car arrives at the intersection. We will study two different scenarios for this mechanism, depending on whether the switching time δ is 0 or positive.

For zero switching time and two-lane intersections we prove a stronger notion of monotonicity. Intuitively, it states that bidding higher results in *moving* faster.

Lemma 4. *Let $b_i < b'_i$ and let (b_i, b_{-i}) and (b'_i, b_{-i}) be two bidding vectors for a static instance of a two-lane intersection with zero switching time. Furthermore, let $\pi_{i,t}$ and $\pi'_{i,t}$ be the position of car i at time-step t under the optimal schedule for (b_i, b_{-i}) and (b'_i, b_{-i}) respectively. Then, for any t it holds that $\pi_{i,t} \geq \pi'_{i,t}$.*

With Lemma 4 we can prove that the locally optimal mechanism is monotone, and thus we get an IC mechanism using Myerson's payments.

Theorem 3. *The locally optimal mechanism is monotone on two-lane intersections with zero switching time.*

When switching time is positive, the locally optimal mechanism is no longer monotone even on two-lane intersections, and thus *not* IC under Myerson's payments. Consider the example of Figure 1, extended by the car with VoT $x > 100$ that arrives at time 1 at the horizontal lane, while the car with VoT 2 is crossing the intersection. The locally optimal mechanism implements the schedule $s, \downarrow_2, s, \rightarrow_5, \rightarrow_3, \rightarrow_x, s, \downarrow_9$. The car with VoT 9 crosses at time $5 + 2\delta$, and has positive payment. If it bids zero, the mechanism implements the schedule $\rightarrow_5, \rightarrow_3, \rightarrow_x, s, \downarrow_2, \downarrow_9$, it crosses at time $5 + \delta$, and its payment is zero. So, it is better off by lowering its bid.

Computing Payments

Locally optimal mechanism requires knowing the future for computing Myerson's payments. This is because computing the time car i which bid b_i would have crossed had it bid $b'_i < b_i$ requires simulating the mechanism with the bid b'_i. Because bidding lower means crossing later, cars that arrived after car i actually crossed (with a bid of b_i) might affect the crossing time of i under bid b'_i. However, we do not have to compute payments immediately when car i crosses. Rather, we only have to compute optimal schedules with b_i online, and we can compute the payments offline, when we already know the "future" which occurred after car i crossed. Because our mechanism is IR, we can tell each car how much it has to pay later, while preserving the participation incentives.

4.3 Pareto Optimality

We now discuss the theoretical guarantees of our proposed mechanisms. We take a competitive analysis perspective, where for every car that arrives online an adversary chooses the lane and the time it joins, and its VoT. We show that any online algorithm (not just our proposed mechanisms) can be arbitrarily far from optimal in the worst case.

Firstly, observe that the cost $v_i \cdot t_i(\sigma)$ car i suffers under schedule σ can be decomposed to *base* cost and *extra* cost. Car i will suffer the base cost under *any* schedule due to its initial position on its lane; if car i has position k on its lane, then it will suffer $(k + 1) \cdot t_c \cdot v_i$ cost irrespective from the implemented schedule since every car in front of it has to cross the intersection. The extra cost depends on the schedule σ and it is the difference between $v_i \cdot t_i(\sigma)$ and the base cost of car i. Observe, the optimal schedule minimizes the sum of the extra costs of the cars.

Consider a two-lane intersection with $t_c = 1$ and no switching time. Initially, there is one car in every lane each with value 1. At time $\epsilon < 1$ a new car with value x arrives on one lane. If the mechanism moves lane l_1 first, then the adversary locates the new car on lane l_2. Else, it locates it on l_1. Thus, the extra cost of any schedule is $(1 - \epsilon) \cdot x + 1$ while the optimal schedule in hindsight has extra cost 2.

Theorem 4. *For the dynamic setting there is no deterministic algorithm that achieves bounded competitive ratio with respect to the extra cost.*

On the other hand, we can prove that there is no mechanism that achieves strictly smaller cost than our mechanisms over every possible instance, hence they are pareto-optimal. Specifically, our mechanisms are optimal when no new cars arrive. Thus, when all futures (including this one) are possible, our mechanisms are on the pareto-frontier.

Theorem 5. *Both the statically optimal mechanism and the locally optimal mechanism are pareto-optimal.*

5 Empirical Evaluation

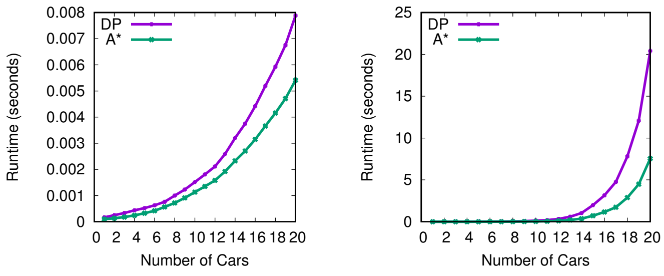
In order to empirically evaluate our mechanism, we implemented it in Python, and performed a series of simulated runs. We used two types of intersections in our experiments: a *simple* four way intersection (with four incoming lanes, which all continue straight), and a more *complex* four way intersection with separate lanes for continuing straight and for turning left (for a total of 8 incoming lanes). For the complex intersection, we randomly choose $\frac{2}{3}$ of the cars coming from each direction to continue straight, and $\frac{1}{3}$ to turn left. The switching time we used in our experiments was 0.

For every intersection, we considered different distributions on traffic volume and on VoT of car. First, we consider a symmetric distribution, where cars are equally likely to arrive from all four cardinal directions, and all cars have their VoT drawn from the same distribution. We sampled VoT according to the empirical study of [Abou-Zeid *et al.*, 2010], which estimated the VoT of real drivers.[1] We also considered asymmetric distributions, parameterized by S, in which $\frac{1}{S}$ of cars arrive from north or south (equally likely), and the rest from east or west (also equally likely). However, the VoT of cars coming from the north or south is S times higher on average than of those coming from the east or west, which we achieved by drawing a VoT from the same distribution as above and multiplying it by S.

We first report on runtime results, comparing the time it takes to find an optimal schedule using A^* with our heuristic compared to using dynamic programming. For each type of intersection, we sample 30 random instances with n cars in the intersection, where n varies between 1 and 20. These experiments were run on a laptop with an Intel i7-6700HQ processor running at 2.6 GHz. Figure 2 shows the wall-clock running time of A^* with our heuristic and of dynamic programming on a simple intersection and on a complex intersection. In both cases we used symmetric volume of traffic and VoT, but the results for the other settings are very similar. As Figure 2 shows, A^* is much more efficient, saving about 30% time for the simple intersection, and about 70% of the time for the complex intersection, compared to dynamic programming. We also remark that we believe an efficient implementation in C++, which avoids copying all of the state when generating the successors, will improve on these wall-clock times by at least an order of magnitude.

Next, we evaluate our mechanisms for the online setting. We performed simulations starting from a random initial configuration of the intersection with 10 cars, and simulating forward for 100 time steps. At each time step, cars arrive with a Poisson distribution, where we vary the arrival rate between 0 and 1. We compared both of our proposed online mechanisms: the locally optimal mechanism (denoted local opt),

[1] We actually used a piecewise-linear approximation of the CDF described in the paper, as the full numerical data was not available.



(a) Simple Intersection

(b) Complex Intersection

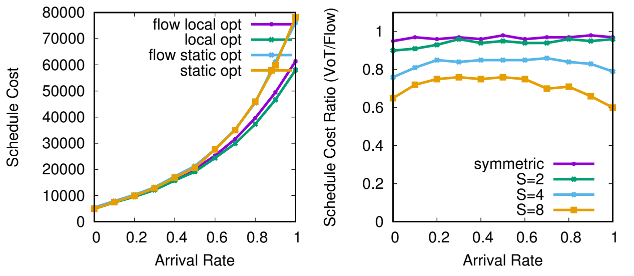
Figure 2: Comparison of runtimes of A^* vs. Dynamic Programming

and the statically optimal mechanism (denoted static opt). In order to evaluate the benefit of taking VoT into account we compare against algorithms that optimize traffic flow and ignore VoT. For flow algorithms, the optimal schedule is computed by running A^* with a bid of 1 for all cars, and either computing a new schedule whenever a new car arrives (denoted flow local opt), or only when the current schedule finished executing (denoted flow static opt). We did not compare our approach with reservation-based approaches as they rely on a different type of intersection management infrastructure. We also did not compare with non IC mechanisms as it is not clear how to implement a *reasonable* bidding strategy for them, and thus the comparison would be meaningless.

Figure 3a shows the schedule cost — the total value of time all cars spent in the intersection under the realized schedule — for all four mechanisms we evaluate, in the complex intersection with symmetric traffic. Each point is the average of 100 random simulations. Results for all other scenarios we tested are similar.We also compared to many round-robin traffic light schedules (with different cycle times), but these results are much worse than any of the abovementioned mechanisms, so we omit them to make the plots more legible.

These results indicate that the locally optimal mechanisms perform much better than the statically optimal ones, especially when the arrival rate is higher. We can also observe that when the arrival rate is high, optimizing for VoT can do slightly worse than optimizing for flow when using the statically optimal mechanism. This is because as arrival rate increases, so does the number of cars in the intersection whenever we search for a new schedule, and thus the reaction time of the statically optimal mechanism is higher. Thus, it is very likely that a high value car will arrive while the previous schedule is executing, but will have to wait until the schedule finishes executing. However, averaging across all arrival rates, optimizing for VoT is better than optimizing for flow with the statically optimal mechanism. Finally, we remark that with the locally optimal mechanism, optimizing for VoT is always better than optimizing for flow.

To evaluate the benefit of optimizing for VoT relative to optimizing for flow with the locally optimal mechanism more closely, we compared the ratio of the schedule cost when optimizing for VoT and the schedule cost when optimizing for flow, using different ratios of asymmetry in the traffic distribution. These results are shown in Figure 3b, based on the same data described above. The results clearly show how optimizing for VoT becomes much better than optimizing for flow, the more asymmetric traffic is, with almost 60% of the cost of wasted time when $S = 8$.



(a) Schedule Cost for Complex Intersection, Symmetric Traffic

(b) Ratio of Schedule Costs between the Locally Optimal Mechanisms for VoT and for Flow

Figure 3: Empirical Results of Online Mechanisms. Schedule Cost is the Total Value of Time for All Cars.

6 Discussion

In this paper we studied the intersection signalling control problem for cars with heterogeneous VoT. We derived incentive compatible mechanisms and showed that they waste significantly less value of time compared to flow algorithms. We believe that our approach could be implemented without massive infrastructure changes. For example, car drivers can install an application on their smartphones which they can use in order to bid and pay. The application can use the GPS of the phone to obtain the location of the car and its position on the lane. The application will send the information to an intersection-dedicated server where the schedule is produced.

Several interesting questions stem from our paper both theoretically and practically, that we plan to study in the future. We conjecture that the statically optimal mechanism is IC for any intersection with zero switching time. Theorem 4 shows that there is no deterministic algorithm that is boundedly suboptimal when the scheduler has no information about the future traffic. On the other hand, if there is stochastic information about future traffic, then we can formulate the signalling problem as an MDP and utilize the framework of [Cavallo *et al.*, 2009]. Our preliminary results show that we can get IC mechanisms via this framework that are not ex-post IR. We plan to construct ex-post IR mechanisms for this setting, or prove that no such mechanism exists.

Another interesting direction we plan to study is a network of intersections. There, we have to consider how to coordinate the *flow* of the traffic between the intersections alongside the possibility of route-manipulations by the drivers.

Finally, throughout this paper we assumed that drivers know their VoT. In order to deal with this issue, we plan to study the existence of incentive compatible mechanisms that offer “menus” of options to the drivers alongside the option of bidding their VoT, if they know it. For example, such a menu can give to the drivers the option to pay x in order to cross the intersection at 30 seconds or to pay $2x$ in order to cross the intersection in 20 seconds. Then, we expect that drivers will choose the more profitable option for them.

References

[Abou-Zeid *et al.*, 2010] M. Abou-Zeid, M. Ben-Akiva, M. Bierlaire, C. Choudhury, and S. Hess. Attitudes and value of time heterogeneity. *Applied Transport Economics-A Management and Policy Perspective.*, pages 523–545, 2010.

[Becker, 1965] G. Becker. A theory of the allocation of time. *The Economic Journal*, 75(299):493–517, 1965.

[Bellman, 1954] Richard Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6):503–515, 11 1954.

[Bento *et al.*, 2015] A. Bento, K. Roth, and A. Waxman. The value of urgency: Evidence from congestion pricing experiments. Technical report, 2015.

[Bonet and Geffner, 2001] Blai Bonet and Héctor Geffner. Planning as heuristic search. *Artificial Intelligence*, 129:5–33, 2001.

[Cavallo *et al.*, 2009] Ruggiero Cavallo, David C Parkes, and Satinder Singh. Efficient mechanisms with dynamic populations and dynamic types. 2009.

[Christidis and Rivas, 2012] Panayotis Christidis and J Nicolás Ibañez Rivas. Measuring road congestion. *Institute for Prospective and Technological Studies, Joint Research Centre, Brussels*, 2012.

[de Palma and Lindsey, 2011] Andre de Palma and Robin Lindsey. Traffic congestion pricing methodologies and technologies. *Transportation Research Part C: Emerging Technologies*, 19(6):1377 – 1399, 2011.

[Dresner and Stone, 2004] Kurt M. Dresner and Peter Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *AAMAS*, pages 530–537, 2004.

[Dresner and Stone, 2005] Kurt M. Dresner and Peter Stone. Multiagent traffic management: an improved intersection control mechanism. In *AAMAS*, pages 471–477, 2005.

[Dresner and Stone, 2006] Kurt Dresner and Peter Stone. Human-usable and emergency vehicle-aware control policies for autonomous intersection management. In *International Workshop on Agents in Traffic and Transportation*, 2006.

[Gulić *et al.*, 2016] Matija Gulić, Ricardo Olivares, and Daniel Borrajo. Using automated planning for traffic signals control. *PROMET-Traffic&Transportation*, 28(4):383–391, 2016.

[Hart *et al.*, 1968] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, SCC-4(2):100–107, 1968.

[INR, 2014] The future economic and environmental costs of gridlock in 2030. Technical report, INRIX, 2014.

[Isukapati and Smith, 2017] I. K. Isukapati and S. F. Smith. Accommodating high value-of-time drivers in market-driven traffic signal control. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1280–1286, 2017.

[Levin and Boyles, 2015] Michael W. Levin and Stephen D. Boyles. Intersection auctions and reservation-based control in dynamic traffic assignment. *Transportation Research Record: Journal of the Transportation Research Board*, 2497:35–44, 2015.

[Levin *et al.*, 2017] M. Levin, H. Fritz, and S. Boyles. On optimizing reservation-based intersection controls. *IEEE Trans. Intelligent Transportation Systems*, 18(3):505–515, 2017.

[Lloret-Batlle and Jayakrishnan, 2016] R. Lloret-Batlle and R. Jayakrishnan. Envy-minimizing pareto efficient intersection control with brokered utility exchanges under user heterogeneity. *Transportation Research Part B: Methodological*, 94:22 – 42, 2016.

[Myerson, 1981] Roger B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6(1):58–73, 1981.

[Nisan and Ronen, 2001] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1-2):166–196, 2001.

[Pourmehrab *et al.*, 2017] M. Pourmehrab, L. Elefteriadou, S. Ranka, and M. Martin-Gasulla. Optimizing signalized intersections performance under conventional and automated vehicles traffic. *arXiv preprint arXiv:1707.01748*, 2017.

[Raphael *et al.*, 2015] Jeffery Raphael, Simon Maskell, and Elizabeth Sklar. From goods to traffic: First steps toward an auction-based traffic signal controller. In *PAAMS*, pages 187–198, 2015.

[Schepperle and Böhm, 2008] Heiko Schepperle and Klemens Böhm. Auction-based traffic management: Towards effective concurrent utilization of road intersections. In *CEC / EEE*, pages 105–112, 2008.

[Schepperle *et al.*, 2007] Heiko Schepperle, Klemens Böhm, and Simone Forster. Towards valuation-aware agent-based traffic control. In *AAMAS*, pages 185:1–185:3, 2007.

[Sharon *et al.*, 2017] G. Sharon, J. Hanna, T. Rambha, M. Levin, M Albert, S. Boyles, and P. Stone. Real-time adaptive tolling scheme for optimized social welfare in traffic networks. In *AAMAS*, pages 828–836, 2017.

[Smith *et al.*, 2013] Stephen F. Smith, Gregory J. Barlow, Xiao-Feng Xie, and Zachary B. Rubinstein. Smart urban signal networks: Initial application of the SURTRAC adaptive traffic signal control system. In *ICAPS*, 2013.

[Tsekeris and Voß, 2009] T. Tsekeris and S. Voß. Design and evaluation of road pricing: state-of-the-art and methodological advances. *NETNOMICS: Economic Research and Electronic Networking*, 10(1):5–52, Apr 2009.

[Vallati *et al.*, 2016] M. Vallati, D. Magazzeni, B. De Schutter, L. Chrpa, and T. McCluskey. Efficient macroscopic urban traffic models for reducing congestion: A pddl+ planning approach. In *AAAI*, pages 3188–3194, 2016.

[Vasirani and Ossowski, 2008] M. Vasirani and S. Ossowski. Towards reservation-based intersection coordination: an economic approach. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 1–6, 2008.

[Vasirani and Ossowski, 2010] M. Vasirani and S. Ossowski. A market-based approach to accomodate user preferences in reservation-based traffic management. *Int. Conf. Auton. Agents Multiagent Syst.*, pages 103–110, 2010.

[Vasirani and Ossowski, 2012] Matteo Vasirani and Sascha Ossowski. A market-inspired approach for intersection management in urban road traffic networks. *JAIR*, 43:621–659, 2012.

[Xie *et al.*, 2012] X. Xie, S. Smith, and G. Barlow. Schedule-driven coordination for real-time traffic network control. In *ICAPS*, 2012.

[Xie *et al.*, 2014] Xiao-Feng Xie, Stephen F. Smith, Ting-Wei Chen, and Gregory J. Barlow. Real-time traffic control for sustainable urban living. In *ITSC*, pages 1863–1868, 2014.