Micro-Auction-Based Traffic-Light Control: Responsive, Local Decision Making

Michele Covell, Shumeet Baluja, Rahul Sukthankar
Google Research
Mountain View, California 94043

***Abstract*—Real-time, responsive optimization of traffic flow serves to address important practical problems: reducing drivers' wasted time and improving city-wide efficiency, as well as reducing gas emissions and improving air quality. Much of the current research in traffic-light optimization relies on extending the capabilities of basic traffic lights to either communicate with each other or communicate with vehicles. However, before such capabilities become ubiquitous, opportunities exist to improve traffic lights by being more responsive to current traffic situations within the existing, deployed, infrastructure. In this paper, we use micro-auctions as the organizing principle with which to incorporate local induction loop information; no other outside sources of information are assumed. At every time step in which a phase change is permitted, each light conducts a decentralized, weighted, micro-auction to determine which phase to instantiate next. We test the lights on real-world data collected over a period of several weeks around the Mountain View, California area. In our simulations, the auction mechanisms based only on local sensor data surpass longer-term planning approaches that rely on widely placed sensors and communications.**

I. Introduction

Traffic congestion is a practical problem resulting in substantial delays, extra fuel costs, and unnecessary harmful gas emissions. In urban areas, traffic is largely controlled by traffic lights; improving their control and responsiveness to existing travel flows holds immense potential for alleviating congestion and its associated problems.

Inefficient configuration of traffic lights remains a common problem in many urban areas. For example, many traffic lights are based on fixed cycles, which means that they are set to green and red for fixed amounts of time. Rarely is this an optimal solution, as the real-time traffic situation is not considered, and may leave cars waiting in long queues to satisfy shorter queues or even no queue at all. Nonetheless, even assuming fixed-length, non-responsive lights, it is possible to optimize the light timings to handle historic knowledge of average flows that have been observed. Such approaches are often handled through the use of genetic algorithms to optimize the light timings [13], [15], [16], [21]. An alternate approach to the optimizations offered through genetic algorithms is a machine-learning approach based on reinforcement learning that attempts to learn optimal policies for the lights [1], [2], [24].

The methods referenced above have not only been applied to fixed-policy lights, but also to lights that use induction-loop sensors to provide real-time traffic-state information to the light controllers. Additionally, many researchers have looked into future possibilities where communication between cars and lights exists [3], [10], [19]. Such communication allows for better light control through the possibility of communicating schedules to cars and arrival times to lights. Further, communication from lights to other lights allow each light to broadcast its schedule and observed flows; with this additional information, alternative planning and scheduling schemes can be created [17], [25], [26].

It should be mentioned that approaches relying on a centralized controller or hierarchies of controllers have also been explored in the research literature. However, the more coordination that is assumed, the greater the difficulty in scalability and system "nervousness" (where small changes in the overall system state require large changes at the local level) [12]. For scalability, we concentrate only on local-decision making.

In Section IV, we present an approach that only changes the logic in the traffic-light controllers. It does not hypothesize the existence of communication between lights nor between cars and lights. The information that we employ to make lights reactive is provided solely through local induction-loop sensors, placed within the framework of an auction system. When a phase change is permitted, the light controller collects bids from all the phases and conducts a micro-auction. The phase bids are set by current readings from local induction-loop sensors, without remote communication. The magnitude of the bid is based on the real-time data from the induction loops. Both positive and negative bids are allowed and will be described in Section IV. To test the ideas, we collected real-world data from the Mountain View, California area over several weeks. The results are promising: in simulations based on the same empirical data, the micro-auction approach outperforms long-term planning approaches on a number of important metrics, including overall capacity.

We begin with a brief section on how to simulate unconstrained transitions between phases, which is used in the subsequent sections. In Section III, we describe an alternative, planning-based traffic controller, derived from [17]. Section IV details our proposed micro-auction controller. All of the methods presented here require optimization of many parameters. The automated optimization procedure (*NASH*) used for all of the approaches is described in V. Section VI presents an overview of the data that was collected — both the routes and the travel tracks. We present our results in Section VII. The paper concludes with ideas for future work in Section VIII.

II. Unconstrained Transitioning between Traffic Phases

In the context of traffic signal terminology, a *phase* specifies which lanes of traffic at an intersection may flow, and

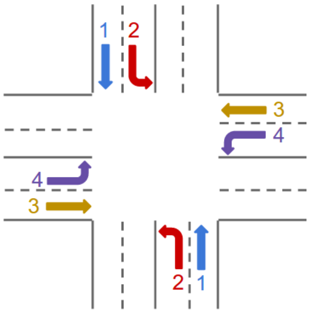

Fig. 1: An example four-phase traffic light

in which directions. For instance, in Figure 1, traffic can flow North-South in phase 1 and East-West in phase 3. Traffic lights typically transition from phase-to-phase in a fixed order, known as *round-robin* sequencing, with phases of constant duration interleaved with appropriate interstage ("yellow") periods. While predictable, such fixed schemes are sub-optimal since they lead to significant under-utilization of the intersection when incoming traffic is unevenly distributed. This has prompted research on *unconstrained* sequencing, where the traffic light can switch to phases out-of-order in order to better service the observed traffic at the intersection.

We use Sumo [14] for all the simulations reported in this paper. Sumo (*S*imulation of *U*rban *MO*bility) is an open-source traffic micro-simulation package [7], Sumo uses discrete time steps (1 msec each) in its simulations but keeps a continuous representation of location, distance, and speed. We selected Sumo since this continuous spatial representation makes it much better for simulations that include congested surface streets, compared to the discrete-space alternatives [20].

Whenever there is a transition from one traffic-light green phase to another, where some of the lights that were green become red, there is the need for an interstage period (that is, a "yellow phase") to provide warning to the affected drivers to stop or to proceed, based on their speed and distance to the intersection. The sequence of green phases is "round robin" by default within Sumo. In the case of round-robin transitions, there is only one yellow phase that will follow any given green phase (since the sequence of greens is fixed and predefined). When we have traffic sensor inputs, we can consider allowing out-of-sequence transitions, based on what traffic is seen at the intersection's local induction loops. However, the potential for improvement in traffic flow requires additional yellow-phase logic.

In both Sections III and IV, we allow lights to make out-of-sequence transitions between green phases. This unconstrained approach makes better use of the intersection capacity, since it can be more responsive to real-time traffic demands. In addition, when the traffic-light logic includes advance planning (as in Section III), there is a second advantage to out-of-sequence transitions. For planning-based logic, forcing a round-robin sequencing necessitates much longer planning horizons to avoid making short-sighted decisions — the light must plan ahead for the combined duration of all the phases in the cycle. In contrast, with unconstrained sequences, the light only needs to plan ahead for the duration of a single yellow phase.

Which signals need to be yellow (that is, which of the displays need to light up yellow, instead red or green) depends on the combination of previous and following green phases. Since we allow out-of-sequence transitions between green phases, we need to create a grid of the correct yellow phases to put in the middle of these transitions. During the Sumo-simulation setup [14], for each pair of green phases, if any green signal needs to change to a red signal, we create the intermediate yellow phase by setting those displays in the new yellow phase as lighting up yellow.

Next, we need to decide the duration of the yellow phase. We can look at the signals that are marked as yellow and trace backwards from them to the maximum of the speed limits for the lanes that those signals control. We then set the yellow duration for that green pair to be that maximum speed divided by the DOT-recommended safe deceleration rate of 3 m/s^2 [22] plus a reaction time of 1 second.

Finally, we added custom logic to Sumo's traffic lights [14] to correctly select the yellow according to the current phase and the requested next-green phase. The additional logic will then follow that selected sequence of phases, from the current green phase, to the correct yellow phase for the green pair, to the selected next-green phase.

III. Planning-based Traffic Light Control

Inspired by the results of the CMU Pittsburgh traffic-light experiment [17], we implemented a traffic light controller that uses remote sensors and planning for deciding traffic phase switching. This planning-based approach provides an alternative against which to compare our auction-based approach.

With the planning-based approach, we solve the phase scheduling problem by collecting the traffic data from a sequence of induction loops that are listed for a given phase of the traffic light, with the loops spaced at about 3 seconds of expected travel time apart from each other. The sensors are placed in the roadway, spaced at distances dictated by the speed limit and this target separation time. In the planning-based approach, we use these sensors for both occupancy and speed data [18]. We assume that they have been placed on all lanes that could lead traffic to the controlled intersection in 15 sec (or less) of travel. When intersections are closely spaced, these non-local sensors may be on the other side of other intersections, leading to a large web of sensors and distributed communication. While this probably includes more sensors (and communication) than the likely minimum require, it enables us to create a detailed speed and occupancy profile for each section of road, which we use in our planning.

Internal to our planning-based traffic lights, we maintain time lines of when we expect cars (observed by local and remote induction loops) to arrive at the controlled intersection. The car counts (from the known induction loops) are scaled by a weighting factor, based on the historically observed turning ratios between the sensor and the traffic light as well as the historically observed rates for which phase will be needed by the incoming traffic (e.g., straight or left turn at the controlled intersection itself). The distance-to-time translation used to put the car onto the time line is based on the observed speed profile for the road (also from the known induction loops). We then use dynamic programming to solve for the best phase sequence and transition times.

The dynamic-programming search starts with a single potential schedule that has the start time of the current phase marked in the past. If there are cars waiting at the traffic light on other phases, it increases the solution space by considering a phase change to each of the other phases that are in demand as soon as is possible, given the yellow-duration lead time needed for scheduling. For example, considering Figure 1, if phase 3 is the current phase and there is traffic waiting for phase 1 and phase 2, the space of possible solutions will expand to three possibilities: (a) remaining on phase 3; (b) changing to phase 1; and (c) changing to phase 2. This expansion of the possible solution space continues with new possible branches being introduced each time a new car arrives.

The timing of the phase changes is restricted so that the minimum time given for each phase of the light (plus the yellow duration) is respected: if phase 4 requires a minimum duration of 3 seconds, then the next possible phase change will be postponed until 3 seconds plus the yellow duration after the start of phase 4. Also, if the current phase is expected to be "empty" before the arrival of the new traffic (that is, no cars waiting as measured by the local induction loop sensors and no cars arriving as measured by the remote induction loop sensors), the proposed phase change for the newly arriving traffic is moved up to be as early as possible, such that the yellow will start just after the current phase is empty. This early change has the advantage of reducing the amount of time that the intersection remains under utilized — for example, in the case where the minimum-duration constraint on the next phase would not allow switching phases a second time as early as would be optimal.

Given how the phases can change (described above), the scheduling solution is then selected based on minimizing a combination of penalties:

- Speed-loss penalty: This is the penalty for forcing a car to stop. The penalty is scaled by the speed limit of the road that the car will be traveling onto. That speed was chosen since it best reflects the acceleration that the car will require, that it otherwise would not have needed had it not been forced to stop.
- Waiting-time penalty: This is a penalty that is linear in the amount of time that each car must stop at a red light.
- Phase-change penalty: This is a penalty that is added for each proposed phase change. It increases the penalty on the schedules that unnecessarily cycle through the different phases when there is no waiting or incoming traffic.

The solution space is repeatedly expanded and then pruned, using the approach described in [26]. It is expanded by moving forward through all of the arrival times of incoming traffic. After each expansion, the solution space is checked for schedules that can be removed, due to a higher partial cost with the same ending phase and the same numbers of waiting cars. We describe our results with planning-based control in Section VII.

The alternative that we propose to planning-based lights is an auction-based system. It is described next.

TABLE I: Definitions used in Auction-Based Control

Minimum duration: The shortest duration that the current phase will be active for.

Priority duration: The duration before which the current phase has priority on the light's control. Before this time, as long as the current phase's "bid" is non-negative, it will remain active. After this time (or if the current phase's bid is negative before this time), the active phase is selected through an "auction" process.

Release duration: The duration after which the current phase cannot continue to be active, unless none of the other phases want the control: that is, unless all the other "bids" are negative. After this time, the active phase is selected through a "handicapped auction" process.

Auction: The process of comparing positive, zero, and negative bids placed by each of the green phases of the light, to determine which green phase should become active (after an intervening yellow phase, if needed). The auction is won by the highest bid (even if that is negative). Ties are broken by proceeding round-robin through the maximum-bid phases.

Handicapped auction: The auction process that is used after "release duration". In a handicapped auction, the current phase's bid is limited to be less than or equal to zero. With that handicap applied, the auction proceeds as before.

Bid: The weighted sum of the current, local induction-loop measurements (s_j): the bid b_i for phase i is $b_i = \sum_j w_{ij} s_j$. The weights (w_{ij}) are selected by the optimization process. When the optimization process sets $w_{ij} = 0$, we refer to that as having removed sensor j from phase i.

IV. Auction-based Traffic Light Control

In contrast with the planning-based traffic-light control [26], our auction-based approach does not require the use of *remote* sensors since no planning is required. Instead, only the typical induction loops, at the entrance lanes to the controlled intersection, are used. It also does not use light-to-light or car-to-light communication. Auction-based traffic-light control takes an approach similar to current on-demand phase switching but makes better use of the local induction-loop data.

Each phase of this traffic light logic has three time-separated behaviors (see Table I). The timing and inputs used for these behaviors are pre-optimized to the general traffic patterns that are expected a given time of day (e.g., morning commute hours). Based on our tests, it is the combination of these three behaviors along with the input optimization that are responsible for the improved efficiency. We describe the behaviors in this section and the optimization process in the next section.

In auction-based logic, each traffic-light phase definition includes a weighted list of sensors that is to be used by that phase to determine its "bid" for the cycle at any given time. The weights can be positive or negative and are used to effectively scale the number of cars observed on that sensor. For example, a phase for a lower-priority road could include in its sensor list a *negative-weighted* induction loop from in front of the higher-priority road, so that the lower-priority road would be more likely to release the phase when cars arrive from the higher-priority direction. Similarly, a more heavily-used phase could use a larger positive weight for some of its own sensors, to allow it to "out-bid" the lighter-traffic direction. Instead of having these weighted lists of sensors be given as a fixed input that is manually created, the automated optimization procedure selects which of the local sensors to use with what weight for each phase. This process is described in Section V.

Since the optimization procedure is allowed to remove any

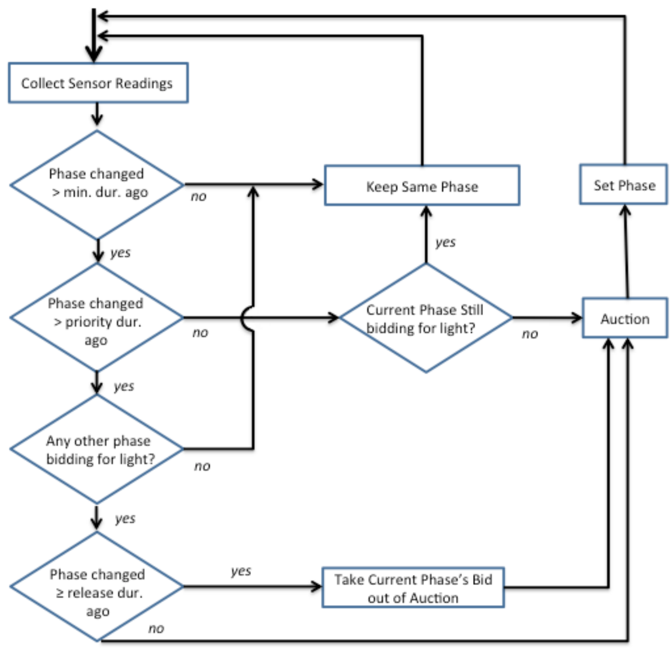

Fig. 2: Flowgraph of phase change decisions within the auction-based traffic light

or all of the sensor inputs to any given phase's bid, we designed our control logic to handle these cases. When a phase has no sensor inputs, it will continually put in a zero bid for the light. If the optimization procedure removes all sensor inputs from *all* phases of the traffic light, the auction logic results in the light behaving as a static light, using round-robin cycling and using each phase's "priority duration". As we will discuss in Section VII, this no-sensor situation does arise in our tests.

As shown in Figure 2, the way in which the traffic light decides whether to change phase depends on how long the current phase has been active. We use the terms, "minimum duration", "priority duration", and "release duration" to separate the time intervals for these different behaviors. As suggested by its name, "minimum duration" is the minimum amount of time that a phase must be green before possibly changing to yellow. The minimum duration is given as an input to the simulation and can be set to be different for each phase of the light. For simplicity, we start with all of the minimum durations as 3 seconds but allow the optimization procedure (Section V) to adjust that to be larger, if needed to avoid "thrashing" between phases in light traffic. No phase changes can occur before minimum duration.

For each second between minimum duration and "priority duration", the current phase has priority on the traffic light. If its bid for the cycle is non-negative (indicating that it would like to have the cycle), then it will keep the cycle, no matter what the bids of the other phases are. While this greedy approach may seem to be suboptimal, it has the advantage of increasing the average duration of the cycles and reducing the amount of time spent switching between green phases (and thereby reducing amount of time wasted on yellow lights). Again, we use parameter optimization to adjust this priority duration according to the expected traffic demands.

For each second between priority duration and "release duration", a micro-auction is held between the different phases. Each phase bids according to the weighted sum of the sensors that have been selected for that phase. If the highest bid is negative, then the current phase is the default winner of the cycle until one or more of the bids change. Otherwise, the phase with the highest bid will get the cycle (after the appropriate yellow). If multiple phases put in the same winning bids, then the winner is selected by proceeding round-robin through the phases, starting with the phase after the current phase.[1]

For each second after the release duration, the same type of auction is held but with the added constraint that the current phase cannot bid an amount above zero. This non-positive bid by the current phase "releases" the cycle. Any other phase that would like to have the cycle will win the auction away from the current phase. If more than one of the other phases have positive bids, the auction process will pick the strongest bidder.

The progression restarts after the start of each green phase, progressing from non-negotiable (below minimum duration) to greedy (below priority duration) to auctioned (below release duration) to a handicapped auction (above release duration).

V. Parameter Optimization for Traffic Light Control

No matter which algorithm is used for traffic light control (fixed-schedule, planning-based, or micro-auctions), each approach has numerous parameters that must be specified to complete the program (Table II). For example, in the simplest case, with fixed schedules, for each light, the length of the phases and their offsets have a large impact on the performance of the system. As mentioned earlier, many machine learning approaches have been used in setting these parameters. Perhaps the most common approach seen in traffic light optimization literature is the use of genetic algorithms (GA) [9] to set the numeric or enumerable values associated with traffic lights [13], [15], [16], [21].

Despite the prevalence of genetic algorithms in this domain, we have found that a much simpler mechanism, *next-ascent stochastic hillclimbing (NASH)* works as effectively as GAs and is simpler to implement and faster in practice. The basic algorithm is described below. This topic is further explored in [4].

For any given approach, we start with specifying the set of parameters that can be modified. With those, NASH operates as follows. A parameter is randomly chosen from the set and the modification operator for that parameter is applied. In the simplest case, if the parameter is a real number, it is perturbed by a small amount (for example $\pm 5\%$). If the parameter can take on a set of distinct values, another value is selected. Once the parameter modifications are made, the schedule is then "repaired", if needed. The repair process ensures that the parameters are consistent with each other and are set within the appropriate ranges. For example, in the case of fixed-schedule light settings, we may want to ensure that the overall cycle of the light remains the constant to keep all lights in synch, but the

[1] It is this round-robin assignment for tie breaking that causes auction-based lights with no sensors to behave like static-lights.

TABLE II: Parameters Optimized for Each Approach

Fixed-Schedule	Planning	Micro-Auctions
Phase Lengths Phase Offsets	Penalty Weights	Detector Weights Detectors to Use Durations (Minimum, Priority, Release)

individual phase lengths can change. In this case, once a phase length perturbation has been made, the repair process ensures that the other other phase lengths are reduced appropriately to compensate and keep the overall cycle length static.

Once any repairs are made, the new schedule is evaluated with the desired *objective function*, where the objective function can be overall/average wait time, maximum wait time, emissions, etc. If the perturbation improved the performance on the objective function over the previous settings without the perturbation, the perturbation is accepted, and the schedule with the perturbation becomes the new baseline. If the perturbation has not performed as well on the objective function, the perturbation is recorded (so that it is not explored again), the perturbations are discarded from the schedule, and the previous baseline remains unchanged.

The exact number of perturbations made in each iteration is chosen stochastically. The maximum number allowed was determined empirically and varied according to the complexity of the schedule being developed. The more parameters the schedule had, the larger the maximum number of perturbations per step that were allowed. This process is iterated until either a satisfactory solution is found or time expires.

VI. Experimental Set-up

Our goal in this paper is to test the auction-based traffic lights against currently deployed traffic lights (and against planning-based traffic lights), using real road and traffic data. First, we describe what is required to import this data into Sumo [14], the simulation package employed.

A. Anonymous Traffic Data

To allow our tests to be relevant to actual traffic situations, we use a demand profile that reflects the reality of that section of the roads by using anonymized location data collected from opted-in Android users [8]. The raw data, which itself does not include personally identifiable information (PII), is also scrubbed to further reduce identifiability risks. For most of our simulation work, we only keep the starting time for when a track enters the mapped area. However, as will be covered in Subsection VI-B, we do use the differences between the time stamps within each individual track to estimate the currently-fielded traffic-light phase durations and offsets.

From the anonymous track data, several steps are required to make them appropriate for use in our simulations. We select data with tracks that intersect with the map area that we will use in the Sumo simulation, based on the road-segment ids to which they have been "snapped." We also filter by time, limiting to data between a given start and end date. We then alias all the given times down to the weekday and time of day (e.g., overlapping data from multiple weeks to bring together an "average" Tuesday 7:00am–11:00am profile). This allows us to combine several weeks worth of data and make up for fact that not every vehicle traveling through our space/time window of interest will have an opted-in Android phone. By aliasing our data on a week-long cycle, we can bring the demand levels up to the DOT-reported levels and do so with more natural variations in routes and start times than would be seen using simple replication.

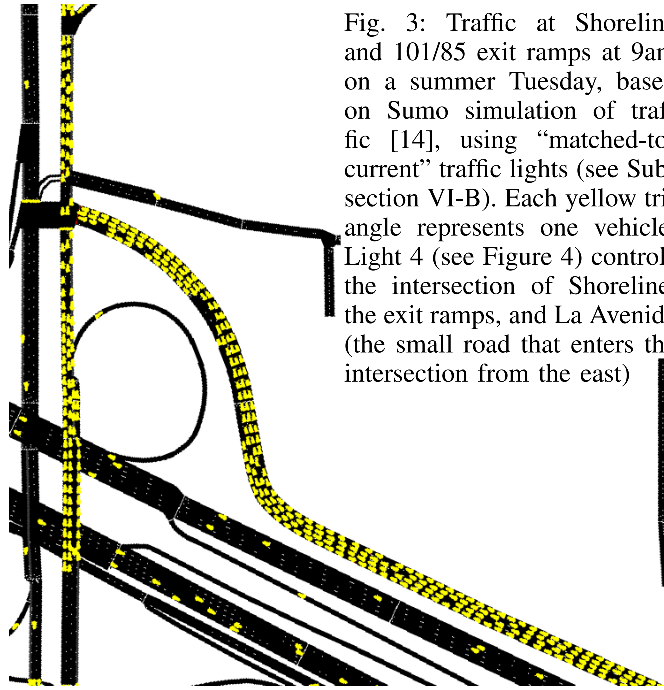

Fig. 3: Traffic at Shoreline and 101/85 exit ramps at 9am on a summer Tuesday, based on Sumo simulation of traffic [14], using "matched-to-current" traffic lights (see Subsection VI-B). Each yellow triangle represents one vehicle. Light 4 (see Figure 4) controls the intersection of Shoreline, the exit ramps, and La Avenida (the small road that enters the intersection from the east)

Finally, we find all the tracks that begin or end at a traffic light within our simulation. Most of the tracks that begin/end at controlled intersections do so as part of the PII filtering: the PII filtering does not distinguish between long stops due to traffic queues versus personal destinations, so many tracks actually begin/end at congested intersections. We can allow cars to enter and exit the simulation at other uncontrolled intersections (similar to cars entering or exiting the roadway from parking areas) but, since we are trying to model the effects of the traffic lights on the flow, we extend these tracks so that, as a group, they have the same turning ratios as are otherwise seen from cars entering or leaving that intersection onto their observed direction.

Figure 3 gives an example of what our simulation estimates as the typical traffic pattern for Tuesday 9am at Shoreline and the 101 and 85 exit ramps.

B. Street and Intersection-Control Data

In order to use real street layouts, we import Google's map data into Sumo, using two map-reduces [6] to complete the import. In the first map-reduce, we select a lat-long region and output an intermediate format for the non-private roadways in that region, re-organizing connection information (e.g., incoming-to-outgoing lane connections and disallowed turn sequences) to bring it in line with what the Sumo import functions require. We also included in these intermediate structures information about stop sign and traffic light locations, using a combination of Google maps and OpenStreetMap [11] data. From this intermediate format, we then complete the

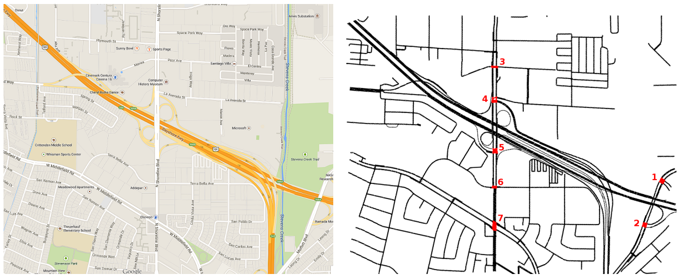

Fig. 4: The area used for the traffic tests, as seen in Google maps (left) and Sumo (right).

translation to Sumo node files, edge files, and connection files. Separating the process into two steps allows us to run over large geographic areas (e.g., thousands of square miles within California) and to then focus our simulation on a much smaller geographic area (e.g., the Shoreline-101 interchange area).

We also want to determine the currently-fielded programming for traffic lights during morning-commute hours. In Section VII, we use that baseline simulation to evaluate whether our approach will improve traffic flow, relative to the current situation or not. We cannot simply use the observed travel times, directly from the anonymous track, since there are many strong sources of variation between those times and what would actually be achievable in our simulations including: possibly incorrect total load estimates (we do not have exact numbers of cars for the time period in question), time-aliased data (we are folding several weeks worth of data over onto itself to provide additional track data, to make up for the subsampling done by needing to have an opted-in Android phone without reducing variability in the final estimated traffic mix), and systematic biases within the simulation software (Sumo's virtual drivers tend to be more law abiding than our local morning commuters). We use the same optimization procedure described in Section V to estimate the phase durations and offsets of the traffic lights in our area, with the objective function forcing a match between the simulated travel times and the observed travel times. We refer to these traffic light programs as *matched-to-current* lights in Section VII. Extensive details about this approach and the accuracy of the results are given in [4].

Figure 4 shows the map area that we used for our tests. It includes seven traffic lights, at the intersections shown in red on the right subfigure. We number these intersections, to allow further discussion in Section VII.

VII. Results

In Tables III and IV, we compare different approaches to traffic light control over different commuter-traffic profiles. We perform the comparison by considering as our baseline the travel times and traffic load under the matched-to-current light controls. For that baseline, the number of cars and the distribution of routes is set to match the processed route data from Subsection VI-A over the indicated time window (e.g., 8:00am–9:30am) on a Tuesday morning in the summer of 2014. We then run the Sumo simulation, and get the average travel time for that set of cars.

When comparing an alternative approach (e.g., optimized static-phase controls) to that baseline, we run the simulation using the alternative control approach, always using the same *distribution* of routes as was used for the baseline. For Table III, we keep the same number of cars as well and just compare the mean travel-time (MTT) changes. For Table IV, we scale the *number* of cars up or down (with the same distribution of routes), until we match its average travel time to the baseline travel time, giving us a measure of the change in capacity. Comparing Tables III and IV, we find that the percent change in MTT is exaggerated compared to the change in capacity. Since promised strong improvements to travel times have seldom remain true for long, due to reactively-increased demand [23], we emphasize the capacity changes in our discussion.

The first lines of Tables III and IV report the results of our comparison of optimized static control to the matched-to-current control. The optimized static control uses the phase durations and relative offsets that were found to be best for an average-morning commute period (weekday 7am–11am), using NASH (Section V). There are improved MTT and capacity throughout the commute cycle, with the largest improvements seen when it is most needed, during the 8:00-9:30am peak period. For example, we could reduce travel time during the peak period by nearly half, without changing demand, using this optimized static controller instead of the matched-to-current controller. Alternatively, we could increase the peak period load by as much as 15% without increasing the travel times. This improvement demonstrates the power of the optimization procedure. It is interesting to see this level of improvement using static light timing, even after the light synchronization work was done on Shoreline Blvd in 2012–13 [5]. It may be that the increase in traffic in those two years is enough to change the best choices for the traffic light durations.

For planning-based lights (second line of Tables III and IV), we did not see a consistent change in MTT or capacity across the commute periods that were studied. The MTT and capacity changes during the peak-rush period were positive (21% and 3%, respectively) but were still less than half of the improvement seen under optimized static lights or auction-based lights, both of which need less infrastructure and less computation. The MTT and capacity changes during the average-rush and low-rush periods were slightly negative compared to the matched-to-current traffic lights. These disappointing results were most likely due to the situation that was studied. The majority of the traffic that was involved in the simulation were cars exiting from a freeway onto a congested arterial road. Most of the traffic was not in "clusters", as observed in the Pittsburgh study [17]. We also did much less optimization on this set of controls, changing only three parameters (the penalty weights) across the full simulation. This choice was because of the obvious physical correlates for our other parameters (observed speed profiles for sensor-to-light delays and turning ratios for sensor-to-phase weights). Even so, it would be interesting to see if we could improve on these results using NASH optimization.

Finally, our auction-based traffic lights provided the largest gains both in speed and in capacity over the matched-to-current controls (third line of Tables III and IV). Of these results, the biggest gains were during the peak-rush period, when it is most

TABLE III: Mean-travel-time (MTT) changes under matched demand

	Peak rush (8:00–9:30 am) 21861 observed cars 668.68 sec observed MTT		Average rush (9:00–10:30 am) 20139 observed cars 173.65 sec observed MTT		Low rush (9:30–11:00 am) 18154 observed cars 110.37 sec observed MTT	
Optimized static lights	44% faster	(376.56 sec MTT)	30% faster	(121.37 sec MTT)	6% faster	(103.58 sec MTT)
Planning-based lights	21% faster	(525.82 sec MTT)	8% slower	(187.18 sec MTT)	6% slower	(116.54 sec MTT)
Auction-based lights	79% faster	(140.87 sec MTT)	38% faster	(108.01 sec MTT)	10% faster	(99.57 sec MTT)

TABLE IV: Capacity changes under matched MTTs

	Peak rush (8:00–9:30 am) 21861 observed cars 668.68 sec observed MTT		Average rush (9:00–10:30 am) 20139 observed cars 173.65 sec observed MTT		Low rush (9:30–11:00 am) 18154 observed cars 110.37 sec observed MTT	
Optimized static lights	+16%	(25306 cars)	+9%	(22037 cars)	+8%	(19661 cars)
Planning-based lights	+3%	(22473 cars)	-4%	(19253 cars)	-2%	(17736 cars)
Auction-based lights	+47%	(32082 cars)	+46%	(29499 cars)	+11%	(20204 cars)

needed. When we used auction-based traffic controls compared to the matched-to-current lights, we were able to reduce the MTT during peak-rush hours to about *one fourth* of the time currently needed for that time window, to a time that is close to the current MTT for the average-rush period. Looking at the capacity changes, we were able to allow 47% more traffic into the road network (using the same distribution of routes) during the peak-rush times, without increasing travel times. While the travel times during peak hours are pretty awful (nearly 4 times longer than just an hour later, for similar travel distributions), being able to hold the travel times to this level even with 47% more traffic is a significant accomplishment. If we were to increase the traffic load during peak hours by 47% using the matched-to-current controls, the travel times would go up by 230% (to 2203.6 sec).

The choices made by the optimization procedure applied in auction-based control are interesting. In four of the seven lights, the optimization procedure removed all sensors from all the phases, resulting in the lights acting as static lights: lights 1, 2, 5, and 7 (see Figure 4 right for numbering). The phase durations seen on all of these lights except light 5 were nearly identical to the durations found when we optimized static-control lights: the other three were within 2 sec (out of 86 sec total cycle length) of the optimized static-control light durations. At least this small amount of variation (if not more) is expected from a stochastic search process, such as NASH.

The optimization procedure also found several different ways to in effect remove dedicated-left phases, especially at the most congested intersections (for example, lights 3 and 6 in Figure 4).[2] This behavior seems especially interesting, given the reduced capacity that is generally seen at intersections with dedicated lefts [23]. One approach that the system found to sideline a dedicated left is to remove all the sensors from that phase (forcing that phase to always bid zero) and to mirror a full set of non-zero sensor weights between the opposing direct-through phases. For example, at light 3, the optimization removed all sensors from phase 4 (see Figure 1) and made the weights for phase 1 and phase 3 mirrors of each other (e.g., if phase 1 had w weight for a sensor, then phase 3 had $-w$ weight for that same sensor). Whenever there was any traffic at the intersection, one of the either phase 1 or 3 would have a positive bid (with the other having an equally negative bid). That positive bid will always win the auction away from the zero bid of the dedicated left, without any ties. Another approach that the optimization process found to remove dedicated lefts mostly from the less-used road was to match the bids of the dedicated left to the direct-through phase that was just before it, in the round-robin ordering. Specifically, on light 6 (Figure 4), phases 1 and 2 share a matched set of sensor weights and phases 3 and 4 share another matched set of sensor weights. This means that the direct-through/dedicated-left pair will always tie on any auctions. Since the direct-through phases are always before the dedicated left phases in the round-robin cycle when starting from an opposing direction, the direct-through will win the tied bidding away from the dedicated left.

The optimization also used the time differences between the release durations to favor traffic on the phases that were more prone to long lines. For example, at light 4 (Figure 4), the priority duration given to the 101 exit ramp was ten times longer than the priority duration given to La Avenida (the small road that enters the intersection from the east). With this 10x weighting, we see the traffic delays suffered by the La Avenida traffic approximately equal (on a per-car basis) the delays seen by the 101-exit-ramp traffic.

Finally, the optimization used the number of incoming lanes allowed to pass through a phase as another way to bias the auction towards the dominant direction of traffic. For example, at light 3, the combined weight given to the sensors of Shoreline traffic is five times that given to the Pear-Ave traffic (the east-west road at that intersection). This was done by giving all of the sensors the maximum-allowed positive or negative weights and simply relying on the number of sensors (which equals the number of lanes) to provide priority to larger streets.

VIII. Conclusions and Future Work

Traffic light optimization can significantly reduce traffic congestion. In this paper, we propose several novel ideas for improving phase switching in traffic lights, relying solely upon local information. Our primary contribution is the idea of using independent micro-auctions at each light to incorporate current

[2]The largest amount of congestion is at light 4. However, due to the complexity of that intersection, there was no dedicated-left phase to remove.

traffic conditions and enable unconstrained phase transitions that optimize traffic flow. We demonstrate the feasibility of the proposed methods in simulation using real-world traffic data gathered from anonymized opt-in location readings from mobile phones.

Optimizing light settings based on expected traffic distributions was shown to improve flow in the Shoreline area two years ago [5]. Even following this earlier optimization, we observe that re-optimizing those settings for current conditions could further improve capacity by up to 16%. More importantly, we note that this type of frequent optimization is both worthwhile and feasible, given the amount of anonymized traffic data that can be obtained from opt-in mobile phone locations.

Among the approaches we evaluated in this paper, the planning-based approach was the least successful. It also was the one to which we applied the least amount of optimization, due to the strong physical interpretation of each of the parameters. A future research direction is to test this approach using fully optimized parameters that may not correspond to physical interpretations. Our preliminary experiments suggest that, if planning-based approaches are to be combined with optimization, they may require a more sophisticated optimization procedure to account for the complex temporal interactions induced by interlocked planning.

This paper demonstrates that local micro-auctions hold a great deal of promise. Not only are they simpler to implement, they require less physical infrastructure than planning-based approaches. When traffic lights are already set up for on-demand switching, the auction-based approaches only require changes in the light's software. We studied this approach in conditions dominated by the confluence of a freeway exit ramp with a congested arterial road. We are interested in evaluating whether the same gains are possible in intra-city grids. In any case, the observed improvement in capacity in our experiments is sufficiently large so as to encourage active exploration of optimized, auction-based traffic light control.

ACKNOWLEDGMENTS

We would like to thank Quarup Barreirinhas and Gueorgi Kossinets for their help on this work and with our write-up.

All of our results were obtained using Sumo [14]. We are grateful to the Sumo team and user group for their development work and support, with special thanks to Jakob Erdmann.

REFERENCES

[1] Abdulhai, B., Pringle, R., Karakoulas, G.J.: Reinforcement learning for true adaptive traffic signal control. Journal of Transportation Engineering 129(3), 278–285 (2003)

[2] Arel, I., Liu, C., Urbanik, T., Kohls, A.: Reinforcement learning-based multi-agent system for network traffic signal control. Intelligent Transport Systems, IET 4(2), 128–135 (2010)

[3] Asadi, B., Vahidi, A.: Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. Control Systems Technology, IEEE Transactions on 19(3), 707–714 (2011)

[4] Baluja, S., Covell, M., Sukthankar, R.: Approximating the effects of installed traffic lights: A behaviorist approach based on travel tracks. In: IEEE Intelligent Transportation Systems Conference (2015)

[5] City of Mountain View and Caltrans' Metropolitan Transportation Commission: Program for arterial system synchronization (pass) fy12/13 shoreline blvd traffic signal timing project. http://files.mtc.ca.gov/pdf/-pass/2012-13_Factsheets/City_of_Mountain_View_Caltrans.pdf (2013)

[6] Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: USENIX Operating Systems Design and Implementation (2004)

[7] DLR Institute of Transportation: Sumo - simulation of urban mobility. http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000 (2015)

[8] Dobie, A.: Understanding Google's Android location tracking. http://www.androidcentral.com/understanding-googles-android-location-tracking (2014)

[9] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc. (1989)

[10] Gradinescu, V., Gorgorin, C., Diaconescu, R., Cristea, V., Iftode, L.: Adaptive traffic lights using car-to-car communication. In: Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th. pp. 21–25. IEEE (2007)

[11] Haklay, M., Weber, P.: Openstreetmap: User-generated street maps. IEEE Pervasive Computing 7 (2008)

[12] Heisig, G.: Nervousness in material requirements planning systems. In: Planning Stability in Material Requirements Planning Systems, pp. 21–64. Springer (2002)

[13] Kalganova, T., Russell, G., Cumming, A.: Multiple traffic signal control using a genetic algorithm. In: Artificial Neural Nets and Genetic Algorithms. pp. 220–228. Springer (1999)

[14] Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of sumo - simulation of urban mobility. International Journal On Advances in Systems and Measurements 5 (2012)

[15] Park, B..B., Messer, C.J., Urbanik II, T.: Enhanced genetic algorithm for signal-timing optimization of oversaturated intersections. Transportation Research Record: Journal of the Transportation Research Board 1727(1), 32–41 (2000)

[16] Sanchez, J.J., Galan, M., Rubio, E.: Genetic algorithms and cellular automata: a new architecture for traffic light cycles optimization. In: Congress on Evolutionary Computation. vol. 2, pp. 1668–1674. IEEE (2004)

[17] Smith, S.F., Barlow, G.J., Xie, X.F., Rubinstein, Z.B.: Surtrac: Scalable urban traffic control. In: Transportation Research Board Annual Meeting (2013)

[18] Thankuriah, P.V., Geers, D.G.: Transportation and Information: Trends in Technology and Policy. Springer Science & Business Media (2013)

[19] Tielert, T., Killat, M., Hartenstein, H., Luz, R., Hausberger, S., Benz, T.: The impact of traffic-light-to-vehicle communication on fuel consumption and emissions. In: Internet of Things (IOT), 2010. pp. 1–8. IEEE (2010)

[20] Tonguz, O.K., Viriyasitavat, W., Bai, F.: Modeling urban traffic: A cellular automata approach. IEEE Communications Magazine (2009)

[21] Turky, A.M., Ahmad, M.S., Yusoff, M.Z.M.: The use of genetic algorithm for traffic light and pedestrian crossing control. Int. J. Comput. Sci. Netw. Security 9(2), 88–96 (2009)

[22] U.S. Department of Transportation Federal Highway Administration: Traffic signal timing manual: Chapter 5 basic signal timing procedure and controller parameters. http://ops.fhwa.dot.gov/publications/-fhwahop08024/chapter5.htm (2013)

[23] Vanderbilt, T.: Traffic: Why We Drive the Way We Do (and what it Says about Us). Vintage Books (2009)

[24] Wiering, M.: Multi-agent reinforcement learning for traffic light control. In: ICML. pp. 1151–1158 (2000)

[25] Xie, X.F., Smith, S.F., Barlow, G.J.: Schedule-driven coordination for real-time traffic network control. In: International Conference on Automated Planning and Scheduling (2012)

[26] Xie, X.F., Smith, S.F., Lu, L., Barlow, G.J.: Schedule-driven intersection control. Transportation Research Part C 24 (2012)