Approximating the Effects of Installed Traffic Lights: A Behaviorist Approach Based on Travel Tracks

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

***Abstract*—Decades of research have been directed towards improving the timing of existing traffic lights. In many parts of the world where this research has been conducted, detailed maps of the streets and the precise locations of the traffic lights are publicly available. Continued timing research has recently been further spurred by the increasing ubiquity of personal cell-phone based GPS systems. Through their use, an enormous amount of travel tracks have been amassed — thus providing an easy source of real traffic data. Nonetheless, one fundamental piece of information remains absent that limits the quantification of the benefits of new approaches: the existing traffic light schedules and traffic light response behaviors. Unfortunately, deployed traffic light schedules are often not known. Rarely are they kept in a central database, and even when they are, they are often not easily obtainable. The alternative, manual inspection of a system of multiple traffic lights may be prohibitively expensive and time-consuming for many experimenters. Without the existing light schedules, it is difficult to ascertain the real-improvements that new traffic light algorithms and approaches will have — especially on traffic patterns that have not yet been encountered in the collected data. To alleviate this problem, we present an approach to estimating existing traffic light schedules based on collected GPS-travel tracks. We present numerous ways to test the results and comprehensively demonstrate them on both synthetic and real data. One of the many uses, beyond studying the effects of existing lights in previously unencountered traffic flow environments, is to serve as a realistic baseline for light timing and schedule optimization studies.**

Fig. 1: Initial area to be optimized around the Mountain View, California area. A rush hour traffic flow shown (using Google Maps.)

I. Motivation and Background

One of the largest complaints of commuters in the Mountain View, California area is the amount of traffic they face during the morning and evening rush hours. One of the problem areas, controlled by seven main lights, is shown in Figure 1. The goal of our project is to evaluate the timing of the traffic lights on these intersections and also improve them through either better control algorithms or improved sensors. The preliminary results of that study are presented in [6]. One of the issues that we repeatedly encountered (not just in Mountain View) was the difficulty in obtaining the *actual, currently installed* traffic-light programs.

When the task of traffic-light optimization is undertaken, the data for the physical placement and layout of the roads and highways is often publicly available through such providers as OpenStreetMap [9]. Additionally, with the rising ubiquity of cell phone GPS and maps usage, user travel tracks are often voluntarily given by users to improve destination arrival prediction and update traffic information [3]. Even when anonymized, these travel tracks can provide the basis for extensive traffic flow estimation. One fundamental piece of information remains absent that limits the quantification of the benefits of potential new schedules or new algorithms: the existing traffic light schedules and behaviors. Unfortunately, deployed traffic light schedules and offsets are often not known. Rarely are they kept in a central database, and even when they are, they are often not always obtainable. The alternative is to manually inspect a system of multiple traffic lights; for lights on fixed schedules this may be theoretically feasible, but will likely be prohibitively expensive and time-consuming to do at scale. Nonetheless, without the existing schedule information, it is difficult to ascertain the real-improvements that new traffic light algorithms and approaches will have — especially on traffic patterns that have not yet been encountered in the collected data.

An enormous amount of research has been devoted to optimizing traffic light schedules through a variety of machine learning techniques spanning genetic algorithms [11], [14]–[16] to reinforcement learning [1], [2], [17]. Unfortunately, comparatively little work can be found on determining the current deployed light schedules. In this paper, we apply the same machine learning approaches used for light optimization to discovering the deployed schedules. We take a *behaviorist* approach: we attempt to set the light schedules for all the lights in the system we wish to model to best match known car travel timings that we have observed through travel tracks. We attempt to model the overall behavior of the system rather than modeling each light independently.

In the next section, we describe the data collected and the algorithms used. We demonstrate the procedure first on synthetic data (Section III) to get a deeper understanding of its operation, and then apply it to real data (Section IV). Section V presents numerous alternative implementations and extensions. Section VI closes the paper with a discussion of the limitations of this approach and suggestions for future research.

II. Data and Algorithms

For the real-world experiments, two sets of data are needed: the roadway information (layout, speeds, etc.), and travel track information. To gather the road information, we combined the data available from Google maps and OpenStreetMap [9]. The results provided reasonable roadways as well as traffic light locations, as shown in Figure 2. The figure shows the maps as rendered by the traffic simulator, SUMO [13]. To ensure that our results are widely reproducible, all of our experiments conducted in this paper use SUMO (Simulation of Urban MObility), which is open-source and can be freely downloaded [7].

In addition to accurate road information, we need a realistic demand profile for each road section. 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 [3]. The raw data, which itself does not include personally identifiable information (PII), is also scrubbed to further reduce identifiability risks.

From this data, we select data with tracks that intersect with the map area that we will use in our simulations. We also filter by time, limiting to looking at a given start and end date – in particular around rush hour periods. We then filter all the given times down to the weekday and time of day (e.g., Tuesday 7am local time). This provides a close-to-realistic profile of the road demand by allowing us to "overlap" several weeks worth of data. This is required to compensate for the fact that not all drivers are opted-in Android phone users.

Before working with the real data described above, we tune our algorithms on an entirely separate, independent, set of data. This is done to ensure that our algorithms are not overly tuned to only a single data set. To begin our experiments, we create a smaller, synthetic, data set. This has the advantages of being noise-free and completely within our control to modify in order to examine different aspects of matching. For the synthetic data, 3200 cars were instantiated over a period of 4000 seconds to travel along the simple grid shown in Figure 3. The speed limit for each segment was chosen independently and randomly, and seven types of car were instantiated, with differing profiles in terms of acceleration/deceleration, following distance, length, etc.

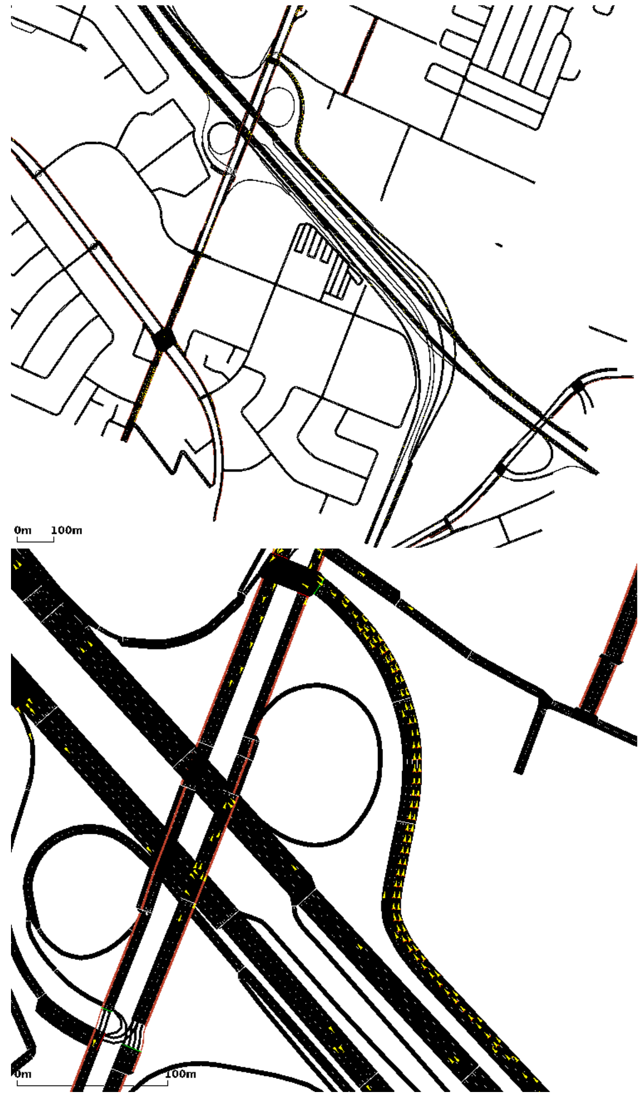

Fig. 2: Roadway data imported into SUMO simulator [13]. Top: Area we are considering. Bottom: expanded region; yellow triangles represent the cars and the traffic that is often present.

Each simulated car's path was chosen to randomly start and end at any edge node. The only constraint in the path was that no intersection could be visited twice. The launch times for the vehicles was uniformly randomly distributed over the 4000 seconds.

To obtain the analogous travel tracks within this synthetic grid, the full 3200 traffic load was simulated in SUMO; the path and start+end times of each car's journey were recorded. The light settings that were used for this initial synthetic data creation are described in the next section.

A. Matching Algorithm

Because the numerous parameters associated with traffic light schedules can have a large impact on overall performance, many studies have used automated machine learning techniques to set them. For example, even in the simplest case

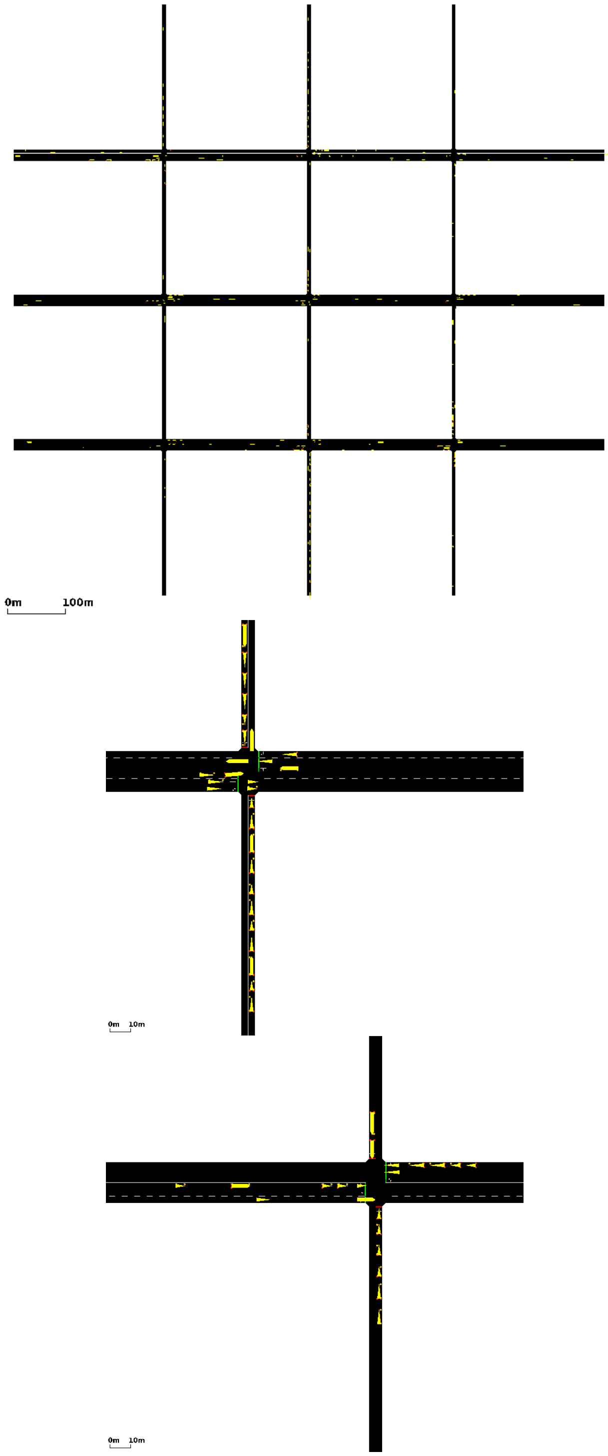

Fig. 3: Three views of the synthetic roadway. Overview (Top). The number of lanes varies between 1 and 3 in each direction. Cars can be introduced and exit at any of the twelve outer edges. Lights are at located at each of the nine intersections. Middle and Bottom: Typical traffic at two intersections shown enlarged.

of traffic lights on fixed schedules, for each light, the length of the phases and the light's offsets have a large impact on the performance of the system. Perhaps the most common approach seen in traffic light optimization literature is the use of genetic algorithms (GA) [8] to set the numeric or enumerable values associated with traffic lights [15] [14] [16] [11]. However, this has been for the purpose of changing the traffic lights' schedules, not for the task at hand: matching a pre-existing schedule.

Nonetheless, the approach can easily be modified to achieve our goal. A genetic algorithm is a type of stochastic search technique that relies on a repeated candidate-generation and evaluate methodology to guide search. The evaluation of the candidate traffic light schedules is controlled by setting an *objective function* that usually measures a characteristic about the system's traffic flow such as mean-wait time, maximum wait time, emissions, fewest stops, etc. Instead of using these traditional objective functions, we specify the following objective function (minimization):

$$\min \sum_{c \in \mathcal{C}} |\text{JourneyTime}_h(c) - \text{JourneyTime}_a(c)|, \qquad (1)$$

where h and a are the hypothesized and actual light settings, respectively, and $\mathcal{C}$ denotes the set of cars in the simulation.

Minimizing this objective allows us to determine light settings that generate a simulated traffic flow that closely mirrors the actual flow through the same road network. Note that although the constraint is not explicitly specified in Eqn. 1, the cars are all introduced into the system in the same order and at the same times as they were observed in the actual data; this is the only way for traffic congestion, etc. to be emulated correctly.

Once this new objective function is specified, the same parameter estimation or machine learning techniques that were previously used can be applied. In our first attempt, we used genetic algorithms and other evolutionary variants as conducted in published research. Despite the prevalence of genetic algorithms in this domain, we have found a much simpler mechanism, *Next-Ascent Stochastic-Hillclimbing (NASH)*, that works as effectively as GAs and is simpler to implement and faster in practice. This has also been observed by other researchers in exploring the trade-offs between genetic algorithms and hillclimbing [10]. This somewhat surprising result is especially pronounced in problems in which mutation (as opposed to crossover) is the main driver for improvement in the solution — as we have found to be the case for this domain.

As with any stochastic optimization technique, GA, NASH or other evolutionary algorithms, we start with specifying the set of parameters that can be modified.[1] For the experiments presented in the remainder of the paper, we initialize the search with a set of reasonable phases, durations and offsets (simply the ones that SUMO uses by default) and then use NASH to adjust each light's phase duration and offset to create a set of light controllers that mimics, as closely as possible, the travel times of the actual cars.

[1] For simplicity, for the studies presented in this paper, we assume that the actual traffic lights can be modeled with fixed schedules. As discussed later in the paper, feedback from induction loops also fits naturally within this procedure.

NASH operates as follows. A parameter is randomly chosen from the set of parameters that are allowed to be changed 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, a value different than the current one is randomly selected. Once the parameter modifications are made, the schedule is then "repaired," if required. The repair process ensures that the parameters are consistent with each other and are set within realistic ranges. For example, in the case of fixed-schedule light settings, we may want to ensure that the overall cycle time of the light remains constant to keep all the lights coordinated, but the individual phase lengths within the cycle 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 objective function described above. If the perturbation improved the performance on the objective function over the previous settings without the perturbation, the perturbation is accepted, and the light 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) and 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 entire process is iterated until either a satisfactory solution is found or time expires.

III. Experiments I: Synthetic Data

As mentioned in the previous section, we randomly generated 3200 travel paths and launch times for cars to travel along the grid shown in Figure 3. The 3200 cars were simulated in SUMO with a pseudo-randomized light setting, this is referred to as the *Target-Light-Setting*. This light setting was slightly optimized beyond SUMO's default light setting to ensure that it could not "accidentally" be found by NASH simply by running SUMO and making tiny perturbations to their default light settings. For the synthetic data experiments, the *Target-Light-Setting* corresponds to the settings for traffic lights that we seek to estimate using the procedures described in this paper. Given these light settings, we simulated 3200 cars in SUMO and the distribution of their travel times (target distribution) is shown in Figure 4 (Top).

The NASH-optimization procedure described in the previous section was then applied to uncalibrated lights to find a timing of the cars to approximate these travel times. The objective function was to match, for each travel path, the arrival time of the car as closely as possible to what SUMO yielded with the *Target-Light-Setting*. Recall that because we want to emulate not having **any** *a priori* information about the actual light settings, we start the NASH-optimization with random settings for all the lights in the system. The hope is

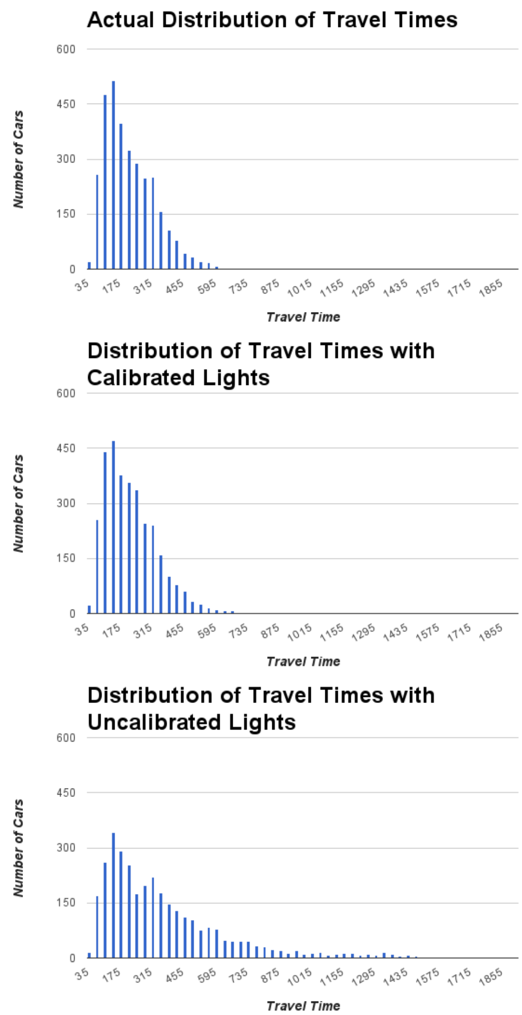

Fig. 4: Distribution of travel times for 3200 cars. (Top) Actual Distribution. We would like to model this target distribution as closely as possible. (Middle) Lights calibrated with NASH-optimization. (Bottom) uncalibrated lights. These are lights with random phases and offsets — presented for comparison.

that through the search/optimization process, the light settings that are found will behave the same as those in *Target-Light-Setting*. It is interesting to note here that there may be many possible light settings that generate similar aggregate traffic flow behaviors. As will be shown in later experiments, when multiple NASH runs are conducted, different, but equally well matching, light settings may be found.

The progress of the matching algorithm is shown in Figure 5. Two lines are shown, the bottom line (in red) shows the best light setting that was found. The top line (in blue) shows the evaluation of each candidate light setting as search progresses. From the red line, note that in the beginning of optimization, the average difference (in seconds) between when a car reaches its destination with the original *Target-Light-Setting* and the approximated light-setting was over 80 seconds. By the end of optimization, this is reduced to 31.8 seconds. Also notice that the blue line is riddled with spikes. This means that the majority of perturbations to the best-light

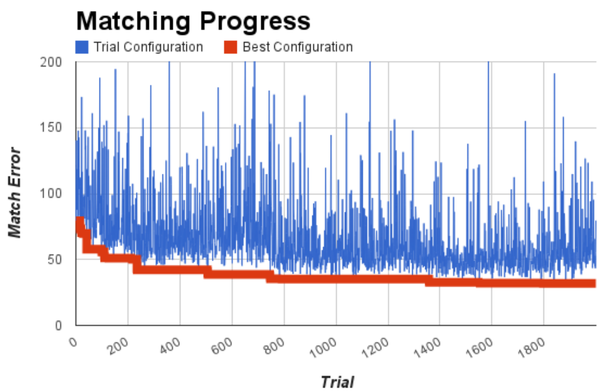

Fig. 5: Match Error over successive trials. 2000 trials shown. Y-axis the average difference in journey times for cars under the hypothesized light setting and the actual light setting (in seconds). X-axis: trial number.

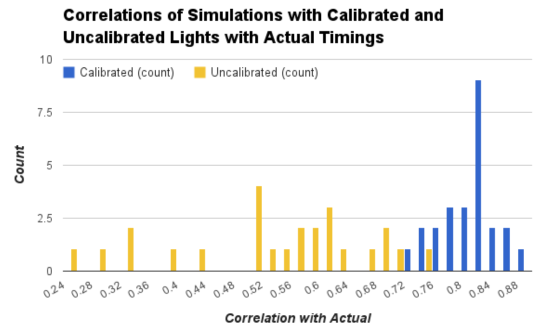

Fig. 6: Correlation of calibrated and uncalibrated lights with actual travel times. Average calibrated correlation = 0.8, average uncalibrated correlation = 0.5.

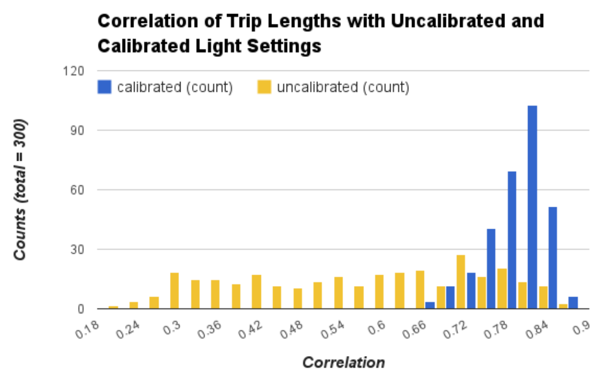

Fig. 7: All pairs correlation of 25 NASH-derived light settings (Blue). All pairs correlation of 25 random light settings (Yellow).

setting found to that point yielded matches that performed significantly worse. Only a few trials, those where the red line took a step downwards, revealed an improved performance. These are the steps in which the NASH algorithm accepted a new baseline from which the optimization then proceeded.

To compare the distributions of travel times to the actual travel times, see Figure 4 (Middle). As can be seen, the travel time appear very close. For comparison, if we look at the travel times for a randomly selected light setting, the distributions look quite different Figure 4 (Bottom).

We can also measure the correlation of journey times under the hypothesized light system and the target light system. Here, we correlate each car's travel times under both scenarios. Results are shown in Figure 6. Instead of just correlating a single trial, Figure 6 also shows the results of 24 additional tests. We reran the entire NASH-matching algorithm from scratch 25 times. Because NASH is stochastic and is initialized with random seeds, we fully expect different schedules to be found in each run; the hope is that at the end of each run, the aggregate behavior approximates the target light system — even if the light settings themselves are not the same. The correlations of the calibrated systems (found through NASH) remains high for all trials.

For comparison, 25 uncalibrated (random) light systems are also shown. It might seem surprising that for random systems there is any correlation; however, since we are measuring travel times, even with random light settings, as cars travel through the entire system, longer paths are likely to have longer travel times, regardless of the light settings. As can be seen, by matching the lights through NASH, the correlation of travel times increases dramatically.

As a final test, we can calculate the correlation of travel times between all pairs of the 25 NASH derived settings. We would expect them to be highly correlated with each other. Figure 7 provides a histogram of those correlations (there are $25 \times 24/2 = 300$ correlations in total). Also shown are the 300 correlations for the uncalibrated networks. As can be seen, very high correlation exists for the calibrated networks and there is low correlation for the uncalibrated networks.

Next we posit the question: how robust are these matches? What happens when we severely alter the underlying traffic profile? How do the travel times of the new cars compare in the NASH-calibrated light systems to the original *Target-Light-Setting*? If the NASH calibrated light systems were truly close to the *Target-Light-Setting*, we would expect a high correlation to remain under any traffic load, not just the load on which it was trained. We perform two experiments to measure this. In the first experiment, 20% of the routes are replaced with randomly selected new routes; additionally the times for the deployed cars can be shifted by up to 300 seconds. In the second experiment, 100% of the routes are replaced with randomly selected new routes. We measure the correlation of timings of the cars under the target-light-schedule, the NASH-derived-light schedules, and the uncalibrated light schedules (random). The results are shown in Figure 8.

As hoped, under both seen and unseen traffic loads, after the NASH-optimization procedure is used to estimate the settings of the lights, we observe that the correlation of travel times to the original *Target-Light-Setting* remains high. Next, we turn our attention to real data. To this point, the study was conducted in simulation with perfectly clean synthesized data. In the next section, we evaluate robustness on noisy, real, data.

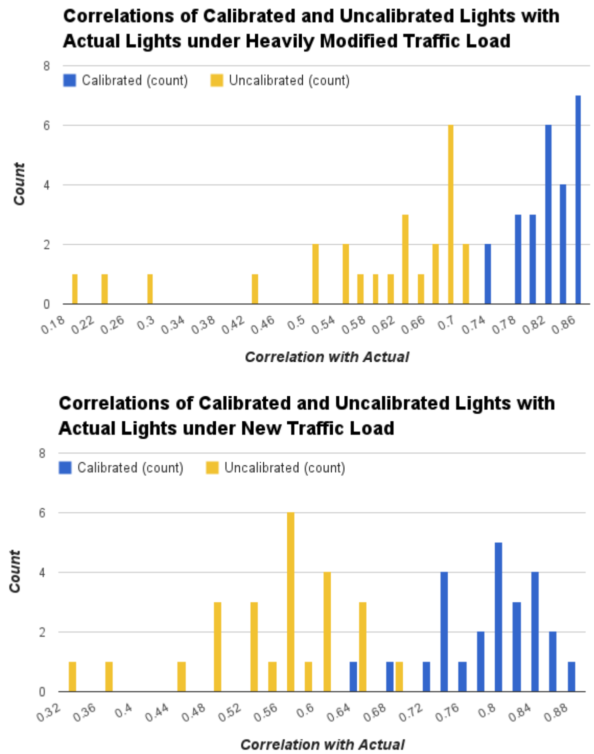

Fig. 8: (Top) Approximately 20% of routes randomly replaced. Bottom: 100% of the routes are replaced.

IV. Experiments II: Real Traffic of Mountain View, California

The results from the previous section with synthetic data were promising. In this section, we apply the same approach to the seven traffic lights in Mountain View, California. This is a much larger and more complex simulation. In the previous section, the synthetic simulations used 3200 travel tracks. For these experiments, we use approximately 67,000 tracks. Additionally, unlike in the synthetic experiments, the distribution of paths is far from uniform. Backups happen non-uniformly — only on certain streets in certain directions. Small perturbations in a single critical traffic light's schedule can lead to drastic changes in throughput while large perturbations to the schedule of a less busy traffic light may generate little observable impact.

We begin this experiment similarly to the synthetic ones — we use NASH to match the timings. The progress is shown in Figure 9. Note that the mean error in times drops quickly. In the beginning of the optimization process, the error was over over 400 seconds (this was when the default SUMO traffic light settings were used for initialization). By the end of the matching procedure, NASH dropped the error between actual and matched times to approximately 51 seconds.

Next, similar to the analysis conducted with synthetic data, we examine the distribution of travel times for the 67,000 tracks. Figure 10 shows the distributions of the tracks for the real data and the times obtained through a simulation in SUMO using the NASH-calibrated lights.

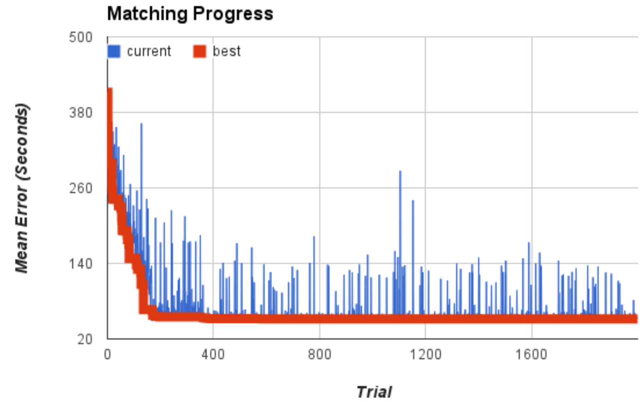

Fig. 9: Progress on matching the real traffic using NASH. Y-axis the average difference in journey times for cars under the hypothesized light setting and the real-data (in seconds). X-axis: trial number.

As can be seen the distributions are similar, but as expected, not as close a match as with the simulated data. The correlation between the predicted and real times is 0.5. For reference, when a random light setting is used (the default SUMO settings), the correlation with the real data is only 0.18. Recall that with the synthetic data, the correlation of NASH-Calibrated lights to the actual timings was 0.8 and the random light settings to the actual timings was 0.5.

There are several important reasons that explain why it was possible to correlate the synthetic data better than the real data. First, in the synthetic data, drivers did not have unnecessary delays in start/stop times due to exogenous factors such as distraction, change of plans, etc. Second, in the synthetic data, drivers behaved uniformly at yellow and red lights; this was not the case for real drivers. Even if the exact correct light settings were found, these factors would lead to lower correlations since they would not be modeled. Third, the real data is significantly more noisy; a problem we did not have with synthetic data. Recall how the data was acquired and aggregated over the period of many months (done to account for the fact that not all drivers are opted-in Android phone users). This aggregation process leads to noise in the real data which we are trying to mimic. Interestingly, as usage of cell-phone GPS/maps increases and more travel track data becomes available, these estimates will improve. Despite the above difficulties, we were able to significantly reduce the average discrepancy between actual and predict times from approximately 7 minutes to 51 seconds.

As a final test to ensure that NASH actually learned something about the lights and did not overfit the exact traffic on which it was retrained, we repeated the experiment with a more difficult setup. For the NASH optimization procedure, we only looked at the data from the first 1.5 months of collection. For testing the timings we tested on a non-overlapping, subsequent, 1.5 months of data. This also represents another common use and test scenario where a period of time is devoted to training and then the learned model is used in the future. What we found indicated that NASH captured the behavior of the lights and did not overfit the data. The results did not change at all

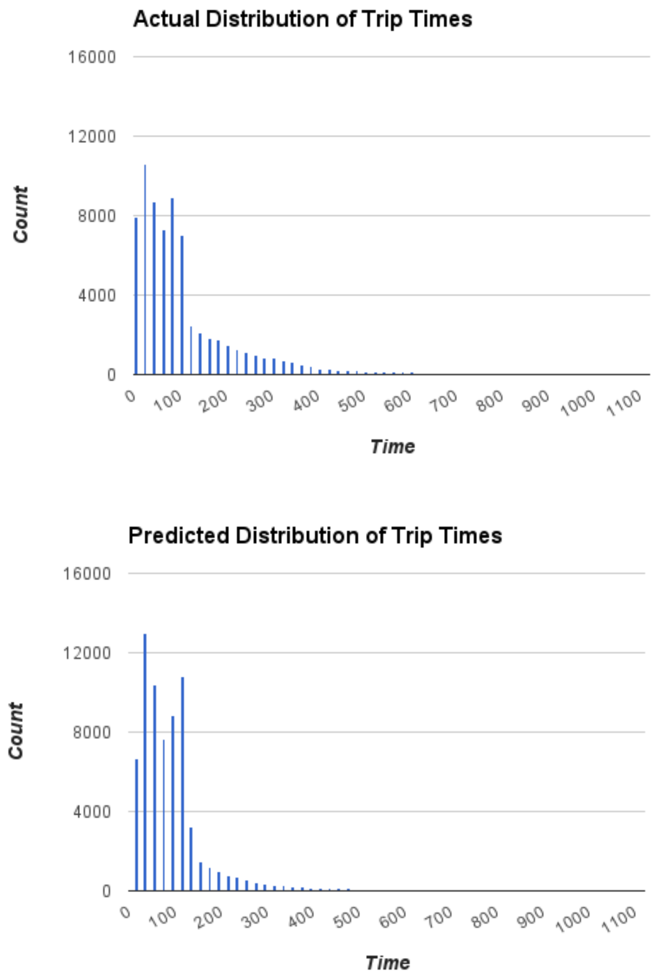

Fig. 10: Distributions of Real (Top) and NASH-calibrated light travel times (Bottom).

from using the entire 3 months of data. Both the mean error and overall correlation remained the same as when the entire 3 month period was used for modeling the lights.

V. Alternative Approaches and Future Work

In the experiments presented in this paper, we attempted to minimize the L_1 error – that is the absolute difference between the time the hypothesized system and the actual system gave for each car. An alternative is to minimize the L_2 error — the sum of squares difference between the two times. By using the L_2 metric, a stronger penalty is placed on the larger discrepancies (outliers). The hope is, therefore, that the errors may be more uniformly distributed instead of allowing some cars have small errors at the expense of a few with very large errors. To experiment with this, we returned to the synthetic data and replaced the error function with L_2 error. Similar to the graph shown earlier (Figure 6), Figure 11 shows the correlation of lights optimized with L_2 with the actual timings. Both methods, optimizing L_1 and L_2 yielded approximately the same correlations with actual.

A second alternative open for future research that may reveal more accurate timings is to use intermediate checkpoints on the path. In this paper, we attempted to match the duration of the travel time for each car as closely as possible; this was measured by the time it took to reach the traveler's **final** destination. However, with many GPS logs, more detailed information is available. For example, if the data is available, the objective function can be rewritten to match times at each intermediate intersection/road along the path to the destination. Although this will increase the computational time because of the increased matching checks required, the algorithm itself will require minimal changes.

A third alternative to the implementation presented is to match dynamic traffic light schedules, such as those that employ induction loops to trigger a change in phase. In the studies presented, the lights were modeled simply, with fixed length cycles. However, this is not a fundamental limitation of the approach. If the presence of an induction loop is suspected, the light controller's internal weighting (or any other parameter) of the induction loop can be specified and therefore optimized similarly to any other parameter in NASH. This is currently an effort in progress to potentially further refine the matches to the real data presented in this paper.

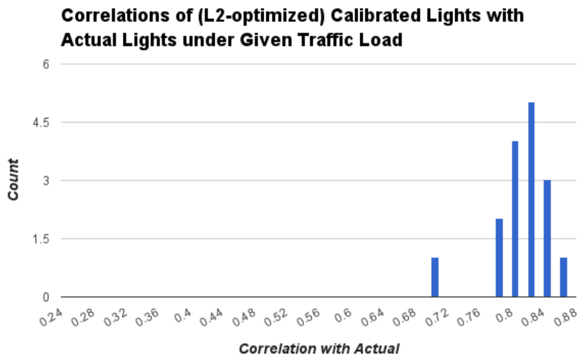

Fig. 11: L2-Correlations.

Fourth, although travel tracks from cell-phone based GPS systems are becoming more prevalent, other sources of traffic data may also be used. Research has been conducted on using cameras to track cars and estimate traffic patterns [4], [5], [12]. If precise timing information is available from such computer-vision based systems, that information could be used to replace or augment the travel tracks gathered through GPS.

VI. Conclusions

For the traffic researcher, two classes of crucial data have become increasingly available. First, detailed maps of the streets and the precise locations of the traffic lights is publicly available through a number of sources. Second, through the increased usage of personal cell-phone based GPS systems, an enormous amount of travel tracks have been amassed. What is often lacking, however, is detailed knowledge of the existing traffic light schedules and traffic light response behaviors. Sometimes this information is not recorded or may be prohibitively difficult to obtain. This paper has presented a simulation-based approach to approximate the behavior of installed lights.

Through a simple hillclimbing procedure, we modified the parameters of the traffic light schedules such that the timing of known traffic tracks matched the timings obtained with the predicted traffic light schedules. We attempted to match the behavior of the system without any knowledge of the actual internal programs.

We hope that this work can be used by practitioners in the traffic optimization field in two ways. First, by estimating traffic light schedules when they are not otherwise available, the system of traffic lights can be studied (for example in simulations) under load conditions that have not previously been seen or yet anticipated. Second, when new algorithms or schedules are devised, having at least a basic behavior-based representation of the current traffic lights will hopefully provide a better, and more realistic, baseline from which to start the evaluation.

Acknowledgments

We would like to gratefully acknowledge Kevin Wang for his help reviewing this paper and his collaboration and discussion on many related projects. All of our results were obtained with the SUMO [13] simulator. We would like to thank the entire SUMO team and the user community for their development work and support.

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] Barth, D.: The bright side of sitting in traffic: Crowdsourcing road congestion data. http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html (2009)

[4] Beymer, D., McLauchlan, P., Coifman, B., Malik, J.: A real-time computer vision system for measuring traffic parameters. In: Computer Vision and Pattern Recognition. pp. 495–501. IEEE (1997)

[5] Coifman, B., Beymer, D., McLauchlan, P., Malik, J.: A real-time computer vision system for vehicle tracking and traffic surveillance. Transportation Research Part C: Emerging Technologies 6(4), 271–288 (1998)

[6] Covell, M., Baluja, S., Sukthankar, R.: Micro-auction based traffic light control: Responsive, local decision making. In: IEEE Intelligent Transportation Systems Conference (ITSC-2015) (2015)

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

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

[10] Juels, A., Wattenberg, M.: Stochastic hillclimbing as a baseline method for evaluating genetic algorithms. In: Advances in Neural Information Processing Systems 8 (NIPS-8) (1995)

[11] 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)

[12] Kanhere, N.K., Pundlik, S.J., Birchfield, S.T.: Vehicle segmentation and tracking from a low-angle off-axis camera. In: Computer Vision and Pattern Recognition. vol. 2, pp. 1152–1157 (2005)

[13] 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)

[14] 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)

[15] Sanchez, J.J., Galan, M., Rubio, E.: Genetic algorithms and cellular automata: a new architecture for traffic light cycles optimization. In: Evolutionary Computation, 2004. CEC2004. Congress on. vol. 2, pp. 1668–1674. IEEE (2004)

[16] 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)

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