2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Adaptive Traffic Light Control for Competing Vehicle and Pedestrian Flows

Yingqing Chen and Christos G. Cassandras

***Abstract*— We study the Traffic Light Control (TLC) problem for a single intersection, considering both straight driving vehicle flows and corresponding crossing pedestrian flows with the goal of achieving a fair jointly optimal sharing policy in terms of average waiting times. Using a stochastic hybrid system model, we design a quasi-dynamic policy controlling the traffic light cycles with several threshold parameters applied to the light cycles and the partially observed contents of vehicle and pedestrian queues. Infinitesimal Perturbation Analysis (IPA) is then used to derive a data-driven gradient estimator of a cost metric with respect to the policy parameters and to iteratively adjust these parameters through an online gradient-based algorithm in order to improve overall performance on this intersection and adapt the policy to changing traffic conditions. The controller is applied to a simulated intersection in the town of Veberöd, Sweden, to illustrate the performance of this approach using real traffic data from this intersection.**

I. INTRODUCTION

The Traffic Light Control (TLC) problem entails dynamically adjusting the traffic light cycles in an intersection or a set of intersections in order to improve the overall traffic performance (normally measured through a congestion metric). Research have been conducted with different traffic models [1] [2] and different optimization methods such as model-based methods [3]–[5] and computational intelligence methods [6]–[8].

Research to date has focused on vehicle flows, while the presence of pedestrians has been largely ignored, even though it is clear that it plays an important role in TLC. There is limited work in the literature on how pedestrians influence traffic [9]–[12]. Although considering pedestrian flows, most of these methods need to train the controller with a large amount of historical data or need a prohibitive amount of computation for a single static traffic scenario (with computational complexity rapidly increasing in more crowded situations). Moreover, with recent technological developments that allow the real-time detection of vehicles and pedestrians at intersections (e.g., [13]), it has become possible to develop real-time traffic-responsive strategies instead of solely relying on historical traffic data. In this context, the key contribution of this paper is a computationally efficient data-driven framework to model the competing interactions between vehicles and pedestrians at an intersection *using online data*. We model a single intersection as a stochastic hybrid system, specifically a Stochastic Flow Model (SFM)

Y. Chen and C. G. Cassandras are with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446 {yqchenn;cgc}@bu.edu.

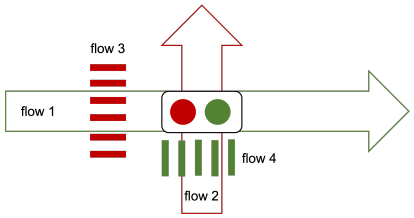


Fig. 1: Single intersection with pedestrian crossings

[14], where the traffic light switching process is event-driven, while the dynamics of the vehicle and pedestrian flows through an intersection are time-driven. Infinitesimal Perturbation Analysis (IPA) [15] is used to estimate online gradients of a performance metric with respect to several parameters of a quasi-dynamic TLC policy considering both vehicle flows and pedestrian flows. These gradient estimates are then used to iteratively seek optimal values for these system parameters. Note that IPA-based gradient estimators are entirely event-driven, therefore our framework scales with the (relatively small) number of events in each system intersection, not the (much larger) state space dimensionality, as recently shown in [16]. Moreover, IPA is independent of any modeling assumptions regarding the stochastic processes characterizing traffic demand and vehicle behavior, driven only by actual observed traffic data similar to learning-based approaches. Compared to [14], the presence of pedestrian flows requires the SFM to include additional queues and the TLC policy to incorporate conditions for enabling pedestrian flows to cross, thus creating a new trade-off to be explored between vehicle and pedestrian performance metrics.

II. PROBLEM FORMULATION

Consider a single signalized intersection as shown in Fig. 1. For simplicity, left-turn and right-turn traffic flows are not considered, and the traffic light combines yellow with red. Note that scenarios considering multiple intersections and turning flows have been explored in follow-up work such as [16], while this paper focuses specifically on the competing actions of vehicles and pedestrians. We consider only two vehicle flows (perpendicular to each other) indexed by $n = 1, 2$ and two corresponding pedestrian flows indexed by $n = 3, 4$ (see Fig. 1). A basic requirement for TLC is that the signals for vehicles and pedestrians are consistent, i.e., when vehicles of flow 1(2) face a GREEN light, pedestrians of flow 4(3) must also face a GREEN light, as indicated in Fig. 1.

Each of the two roads is modeled as a queue where vehicles stop when facing a RED light. Similarly, each

sidewalk where pedestrians wait to cross a road is modeled as a queue. Thus, we define a state vector $x(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]$, $x_n(t) \in \mathbb{R}_0^+$ of queue contents corresponding to the four flows. We assume that queue contents partially observable, meaning we only detect whether the number of queued vehicles or pedestrians is below or above a controllable threshold. This partial occupancy can be observed using sensors such as inductive loop detectors installed on each road near the intersection. Similar to [14], we model the input to each queue as an exogenous random flow process $\{\alpha_n(t)\}$ where $\alpha_n(t)$ is a stochastic instantaneous arrival rate independent of all queue states. The departure flow process is denoted by $\{\beta_n(t)\}$. Note that when $n = 1, 2$, $x_n(t)$ denotes the vehicle queue content of road n, while when $n = 3, 4$, $x_n(t)$ represents the associated pedestrian queue content intending to cross road $n - 2$.

In addition, we define a clock state variable $z_n(t)$ for $n = 1, 2$ to measure the time since the last switch from RED to GREEN for vehicle flow 1,2 respectively. Henceforth, let $\bar{n}$ denote the index of the vehicle flow perpendicular to flow n with the requirement that $z_{\bar{n}}(t) > 0$ when $z_n(t) = 0$. Similarly, for the pedestrian flows, we define $w_n(t)$, $n = 3, 4$ to be the time elapsed since the first pedestrian presence of queue n in the current RED phase; in other words, $w_n(t)$ captures the longest pedestrian waiting time in this phase.

Letting $z(t) = [z_1(t), z_2(t)]$ and $w(t) = [w_3(t), w_4(t)]$ with $z_n(t), w_n(t) \in \mathbb{R}_0^+$, we have the 8-dimensional system state vector $[x(t), z(t), w(t)]$. Before presenting the detailed state dynamics, we define the traffic light controller:

$$u(x(t), z(t), w(t)) = [u_1(t), u_2(t), u_3(t), u_4(t)] \quad (1)$$

where $u_n(t) = 1$ denotes a GREEN light faced by flow n, and $u_n(t) = 0$ denotes a RED light accordingly. We define $u_n(t)$ to be right-continuous in order to accurately represent the control policy defined in the sequel. Due to the basic safety constraint already mentioned, i.e., when vehicles of flow 1(2) face a GREEN light, pedestrians of flow 4(3) must also face a GREEN light, we can eliminate the control values that would lead to a flow conflict. As a result, the feasible control set contains only two elements: $U = \{[1, 0, 0, 1], [0, 1, 1, 0]\}$ and anytime $u_n(t)$ switches its value, the control for the other three flows switches automatically.

Since $\alpha_n(t)$ is an exogenous input process independent of the queue states, we can write the departure process as:

$$\beta_n(t) = \begin{cases} h_n(t), & \text{if } x_n(t) > 0 \text{ and } u_n(t) = 1 \\ \alpha_n(t), & \text{if } x_n(t) = 0 \text{ and } u_n(t) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

for $n = 1, 2, 3, 4$, where $h_n(t)$ is the maximum departure rate which generally depends on the road structure, vehicle specifications and pedestrian move pattern. For the pedestrian flows $n = 3, 4$, we assume $h_n(t) > \alpha_n(t)$ for all t, i.e., once pedestrians start crossing, the queue definitely gets shorter.

We can now write the state dynamics as follows:

$$\dot{x}_n(t) = \alpha_n(t) - \beta_n(t), \qquad n = 1, 2, 3, 4 \quad (3)$$

$$\dot{z}_n(t) = \begin{cases} 1, & \text{if } u_n(t) = 1 \\ 0, & \text{otherwise} \end{cases} \qquad n = 1, 2 \quad (4)$$

$$\dot{w}_n(t) = \begin{cases} 1, & \text{if } u_n(t) = 0 \text{ and } x_n(t) > 0 \\ 0, & \text{otherwise} \end{cases} \qquad n = 3, 4 \quad (5)$$

In (4), note that $\dot{z}_1(t) + \dot{z}_2(t) = 1$ always holds. In addition, we define $z_n(t)$ to be left-continuous, so that at the moment the light switches from GREEN to RED, $z_n(t) > 0$, $z_{\bar{n}}(t) = 0$, $z_n(t^+) = 0$, and $u_n(t) = 0$ (since $u_n(t)$ is right-continuous). In (5), note that $w_n(t^+) = 0$ whenever $u_n(t)$ switches from 0 to 1, and $w_3(t)w_4(t) = 0$ always holds.

Thus, the traffic light intersection in Fig. 1 can be viewed as a hybrid system in which the time-driven dynamics are given by (3), (4), (5) and (2), while event-driven dynamics are associated with light switches with events that cause the value of $x_n(t)$ to change from strictly positive to zero or vice versa. Although the dynamics involve the instantaneous flow processes $\{\alpha_n(t)\}$ and $\{\beta_n(t)\}$, we will show that the IPA-based adaptive controller we design *does not require such knowledge* and depends only on estimating such rates in the vicinity of certain critical observable events.

Controller Specification. Our TLC design is based on the ability to detect events of interest in the state dynamics above, such as a queue content becoming empty. While it may not be possible to detect the exact number of vehicles in a queue (e.g., using cameras), we assume that this number can be estimated so as to classify a queue content $x_n(t)$ as being empty and either below or above some threshold s_n, $n = 1, 2, 3, 4$, as well as the time such transitions occur. For the vehicle queue contents, the joint state space can be partitioned into the following nine regions (see Fig. 2a): $X_0 = \{(x_1, x_2) : x_1(t) = 0, x_2(t) = 0\}$; $X_1 = \{(x_1, x_2) : 0 < x_1(t) < s_1, x_2(t) = 0\}$; $X_1' = \{(x_1, x_2) : x_1(t) \geq s_1, x_2(t) = 0\}$; $X_2 = \{(x_1, x_2) : x_1(t) = 0, 0 < x_2(t) < s_2\}$; $X_2' = \{(x_1, x_2) : x_1(t) = 0, x_2(t) \geq s_2\}$; $X_3 = \{(x_1, x_2) : 0 < x_1(t) < s_1, 0 < x_2(t) < s_2\}$; $X_4 = \{(x_1, x_2) : 0 < x_1(t) < s_1, x_2(t) \geq s_2\}$; $X_5 = \{(x_1, x_2) : x_1(t) \geq s_1, 0 < x_2(t) < s_2\}$; $X_6 = \{(x_1, x_2) : x_1(t) \geq s_1, x_2(t) \geq s_2\}$

Regarding the length of a light cycle (i.e., the values that $z_n(t)$, $n = 1, 2$, can take), we assign a guaranteed minimum GREEN light cycle time θ_n^{min} and a maximum cycle time θ_n^{max}. This is to ensure that traffic light switches are not overly frequent nor can they be excessively long. Complementing θ_n^{max}, $n = 1, 2$, for vehicles, we define an upper bound θ_n to the pedestrian waiting times $w_n(t)$, $n = 3, 4$, so that their waiting never becomes excessive.

An efficient controller design also needs to address the issues of maintaining (i) a proper balance between allocating a GREEN light to competing queues and (ii) preventing the undesired phenomenon where vehicles wait at a RED light at road n while road $\bar{n}$ is empty during its GREEN phase. Such "waiting-for-nothing" instances waste the resources of vehicles that wait unnecessarily and can be eliminated through a proper controller design as detailed next. Towards these two goals, the final parameters we define for our

TLC design are the queue thresholds s_n, $n = 1, 2, 3, 4$ (see Fig. 2). To summarize, we define the following controllable parameter vector:

$$\upsilon = [\theta_1^{min}, \theta_1^{max}, \theta_2^{min}, \theta_2^{max}, \theta_3, \theta_4, s_1, s_2, s_3, s_4] \quad (6)$$

where $\theta_n^{min} \geq 0$, $\theta_n^{max} \geq \theta_n^{min}$, and $\theta_{n+2} > 0$ for $n = 1, 2$; $s_n > 0$ for $n = 1, 2, 3, 4$.

The role of these controllable parameters is to partition the 8-dimensional state space into appropriate subsets that form the basis of a *quasi-dynamic* controller: while in (1) the controller is defined as a function of the full state $[x(t), z(t), w(t)]$, a quasi-dynamic controller is a function of subsets of aggregated states defined by the partition of the queue states shown in Fig. 2 and an additional partition shown in Fig. 2b; the latter is based on defining the following auxiliary state variable for each pedestrian flow $n = 3, 4$:

$$p_{n-2}(x_n(t), w_n(t)) = \begin{cases} 1, & \text{if } x_n(t) \geq s_n \text{ OR } w_n(t) \geq \theta_n \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

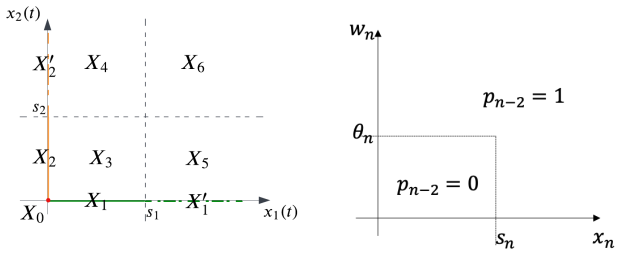
We use the simplified notation $p_{n-2}(t)$ that captures when pedestrian queue n is enabled to cross: when the queue is either long enough or the waiting time is large enough during the current RED phase (see Fig. 2b). The subscript $n-2$ indicates the target road for pedestrian queue n to cross. Thus, the aggregated pedestrian state is $p(t) = [p_1(t), p_2(t)]$.

We are now ready to specify a quasi-dynamic controller expressed as $u(X(t), p(t), z(t), w(t))$ where $X(t)$ is one of the nine subsets defined by the partition of the vehicle queue content space in Fig. 2a and $p(t)$ is the aggregated pedestrian state. Due to the coupling between vehicle and pedestrian demands, a control policy can no longer be as simple as the one in [14]. With the goal of balancing access to a GREEN light to ensure fairness and preventing the wasteful "waiting-for-nothing" phenomenon mentioned earlier, we express the TLC specification in terms of conditions for either maintaining a GREEN light or switching back to it for road 1. Recall that $U = \{[1, 0, 0, 1], [0, 1, 1, 0]\}$, hence the policy for $u_1(t)$ fully defines $u_2(t), u_3(t), u_4(t)$.

1. $(x_1(t), x_2(t)) \in \{X_0\}$: In this case there are no vehicle queues and control is only applied to serve pedestrian queues. First, if the light in road 1 is GREEN ($z_1 > 0$), it switches to RED when pedestrian demand is high for road 1 ($p_1 = 1$) and low for road 2 ($p_2 = 0$), or when demand for both roads is high ($p_1 = p_2 = 1$) when the elapsed GREEN time reaches upper bound (θ_1^{max}). Second, if the light in road 1 is RED ($z_2 > 0$), the logic for switching to GREEN is symmetric. Formally, we define

$$u_1(t) = \begin{cases} 1, & \text{if } [z_1(t) \in (0, \theta_1^{max}),\ p_1(t) = p_2(t) = 1] \text{ OR} \\ & [z_1(t) > 0,\ p_1(t) = 0] \text{ OR} \\ & [z_2(t) \geq \theta_2^{max},\ p_1(t) = p_2(t) = 1] \text{ OR} \\ & [z_2(t) > 0,\ p_1(t) = 0, p_2(t) = 1] \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Due to the space limitation, we omit the explanations for the rest cases which can be found in [17].



(a) State-space partition for vehicle queue content (b) State-space partition for pedestrian state ($n = 3, 4$)

Fig. 2: State space partitions

2. $(x_1(t), x_2(t)) \in \{X_1, X_1'\}$:

$$u_1(t) = \begin{cases} 1, & \text{if } [z_1(t) \in (0, \theta_1^{min})] \text{ OR} \\ & [z_1(t) \geq \theta_1^{min},\ p_1(t) \leq p_2(t)] \text{ OR} \\ & [z_2(t) \in (0, \theta_2^{max}),\ p_1(t) = 0] \text{ OR} \\ & [z_2(t) \geq \theta_2^{max}] \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

3. $(x_1(t), x_2(t)) \in \{X_2, X_2'\}$:

$$u_1(t) = \begin{cases} 1, & \text{if } [z_1(t) \in (0, \theta_1^{max}),\ p_2(t) = 1] \text{ OR} \\ & [z_2(t) \geq \theta_2^{min},\ p_1(t) = 0, p_2(t) = 1] \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

4. $(x_1(t), x_2(t)) \in \{X_3, X_6\}$:

$$u_1(t) = \begin{cases} 1, & \text{if } [z_1(t) \in (0, \theta_1^{min})] \text{ OR} \\ & [z_1(t) \in [\theta_1^{min}, \theta_1^{max}),\ p_1(t) \leq p_2(t)] \text{ OR} \\ & [z_2(t) \in [\theta_2^{min}, \theta_2^{max}),\ p_1(t) = 0, p_2(t) = 1] \\ & \text{OR } [z_2(t) \geq \theta_2^{max}] \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

5. $(x_1(t), x_2(t)) \in \{X_4\}$:

$$u_1(t) = \begin{cases} 1, & \text{if } [z_1(t) \in (0, \theta_1^{min})] \text{ OR} \\ & [z_2(t) \geq \theta_2^{max}] \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

6. $(x_1(t), x_2(t)) \in \{X_5\}$:

$$u_1(t) = \begin{cases} 1, & \text{if } [z_1(t) \in (0, \theta_1^{max})] \text{ OR} \\ & [z_2(t) \geq \theta_2^{min}] \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

The six cases above fully specify our TLC. Note that as the values of $x_n(t)$ change, there are associated transitions from one subset in Fig. 2a to another, in which case the controller $u_1(t)$ follows the new region specifications.

Event definitions. We define observable events associated with mode switches in the hybrid system defined through (3), (4), (5) and (2) under the controller specified above.

For $n = 1, 2, 3, 4$: (a) x_n reaches 0 from above ($x_n \downarrow 0$), (b) x_n becomes positive from 0 ($x_n \uparrow 0$), (c) x_n reaches s_n from below ($x_n \uparrow s_n$), (d) x_n reaches s_n from above

($x_n \downarrow s_n$), (e) α_n reaches 0 from above ($\alpha_n \downarrow 0$), (f) α_n becomes positive from 0 ($\alpha_n \uparrow 0$).

For $n = 1, 2$: (a) z_n reaches lower bound ($z_n \uparrow \theta_n^{min}$), (b) z_n reaches upper bound ($z_n \uparrow \theta_n^{max}$).

For $n = 3, 4$: w_n reaches threshold ($w_n \uparrow \theta_n$).

Since we have defined the auxiliary state variables $p_n(t)$, it is convenient to also define the following compound events for $n = 1, 2$: (1) p_n changes from 0 to 1 ($p_n \uparrow 1$), which happens when either $x_{n+2} \uparrow s_{n+2}$ or $w_{n+2} \uparrow \theta_{n+2}$ occurs, (2) p_n changes from 1 to 0 ($p_n \downarrow 0$), which happens only at a GREEN phase when $x_{n+2} \downarrow s_{n+2}$ occurs.

We can now define the *controllable* event $G2R_n$ $n = 1, 2, 3, 4$, which switches the light faced by queue n from GREEN to RED and triggers a state mode in the hybrid system to switch. This is the event that causes a control switch from $u_n(t) = 1$ to $u_n(t) = 0$. Similarly, $R2G_n$ indicates light switches from RED to GREEN. Note that these controllable events are coupled, i.e., when $G2R_1$ occurs, then $R2G_2$, $R2G_3$, $G2R_4$ also occur at the same time (see Fig. 1). It is important to observe that all $G2R_n$ controllable events are fully defined through the observable events above and the control logic already presented. A complete graphical representation of the hybrid system under TLC in the form of a Stochastic Hybrid Automaton (SHA) is provided in [17].

TLC Optimization Problem. With the parameterized controller defined above, our aim is to optimize a performance metric for the intersection operation with respect to these controllable parameters that comprise the vector υ defined in (6). We choose our performance metric to be the weighted mean queue lengths over a time interval $[0, T]$:

$$L(\upsilon; x(0), z(0), T) = \frac{1}{T} \sum_{n=1}^{4} \int_0^T \gamma_n x_n(\upsilon, t)\, dt \qquad (14)$$

where γ_n is a weight associated with queue n, $n = 1, 2, 3, 4$. Note that a typical sample path of the flow queue content $\{x_n(t)\}$ consists of alternating Non-empty Periods (NEPs) and Empty Periods(EPs), which correspond to time intervals when $x_n(t) > 0$ and $x_n(t) = 0$ respectively, as shown in Fig. 3. Thus, our goal is to determine υ in (6) that (at least locally) minimizes the expected weighted mean queue length $J(\upsilon; x(0), z(0), T) = E[L(\upsilon; x(0), z(0), T)]$.

Note that it is not possible to derive a closed-form expression of $J(\upsilon; x(0), z(0), T)$ even if we had full knowledge of the processes $\{\alpha_n(t)\}$ and $\{\beta_n(t)\}$. Therefore, a closed-form expression for $\nabla J(\upsilon)$ is also infeasible. The role of IPA is to obtain an unbiased estimate of $\nabla J(\upsilon)$ based on the sample function gradient $\nabla L(\upsilon)$ which can be evaluated based only on data directly observable along a single sample path such as Fig. 3, as will be shown in the next section. The unbiasedness of $\nabla L(\upsilon)$ is ensured under mild conditions on $L(\upsilon)$ (see [15]) and assuming that $\alpha_n(t)$ and $h_n(t)$ are piecewise continuously differentiable in t w.p. 1. In particular, we emphasize that *no explicit knowledge of* $\alpha_n(t)$ *and* $h_n(t)$ *is necessary* to estimate $\nabla J(\upsilon)$ through $\nabla L(\upsilon)$.

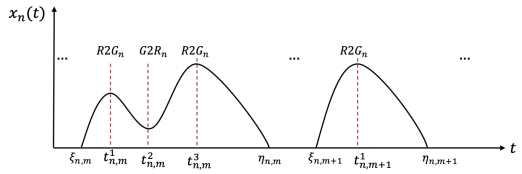


Fig. 3: Typical sample path of a traffic light queue

We can now invoke a gradient-based algorithm of the form

$$\upsilon_{i,l+1} = \upsilon_{i,l} - \rho_l \big[\frac{dJ}{d\upsilon_{i,l}}\big]_{IPA} \qquad (15)$$

where $\upsilon_{i,l}$ is the ith parameter of υ at the lth iteration, ρ_l is the stepsize at the lth iteration, and $(\frac{dJ}{d\upsilon_{i,l}})_{IPA}$ is the IPA estimator of $\frac{dJ}{d\upsilon_{i,l}}$, which will be derived in the next section.

III. Infinitesimal Perturbation Analysis

The IPA framework in [15] captures how system states change with respect to controllable parameters. Our goal is to estimate $\nabla J(\upsilon)$ through $\nabla L(\upsilon)$, and the performance metric expression is a function of event time and system state variables. Thus, we apply the IPA framework to the TLC problem and evaluate how a perturbation in υ would affect performance metrics. Consider a sample path over $[0, T]$ and denote the occurrence time of the kth event (of any type) by τ_k, the state and event time derivatives are as follows.

State Derivatives. We define the derivatives of the state variables $x_n(t)$, $z_n(t)$, $w_n(t)$ and event time τ_k with respect to parameter υ_i ($i = 1, ...10$) as follows:

$$x'_{n,i} \equiv \frac{\partial x_n(t)}{\partial \upsilon_i}, z'_{n,i} \equiv \frac{\partial z_n(t)}{\partial \upsilon_i}, w'_{n,i} \equiv \frac{\partial w_n(t)}{\partial \upsilon_i}, \tau'_{k,i} \equiv \frac{\partial \tau_k}{\partial \upsilon_i}, \qquad (16)$$

Also, we denote the state dynamics at interval time $t \in [\tau_k, \tau_{k+1})$ as follows: $\dot{x}_n(t) = f^x_{n,k}(t),\ n = 1, 2, 3, 4$; $\dot{z}_n(t) = f^z_{n,k}(t),\ n = 1, 2$; $\dot{w}_n(t) = f^w_{n,k}(t),\ n = 3, 4$.

The state derivative of any queue is unaffected within any inter-event time interval (see [15]), i.e., for $t \in [\tau_k, \tau_{k+1})$:

$$x'_{n,i}(t) = x'_{n,i}(\tau_k^+), z'_{n,i}(t) = z'_{n,i}(\tau_k^+), w'_{n,i}(t) = w'_{n,i}(\tau_k^+) \qquad (17)$$

Next, for any discrete event time τ_k, we evaluate queue content derivatives for any possible event occurring to start/end an EP/NEP or within any EP/NEP. Due to space limitations the complete derivations can be found in [17] and the results are summarized below:

$$x'_{n,i}(\tau_k^+) = \begin{cases} 0, & \text{for event inside EP} \\ x^{d}_{n,i}{}'(\tau_k^-) + (\alpha^d_n(\tau_k) & \\ -h^d_n(\tau_k))\tau'_{k,i}, & \text{for event starting EP} \\ -\alpha_n(\tau_k)\tau'_{k,i}, & \text{for } G2R_n \text{ starting NEP} \\ x'_{n,i}(\tau_k^-) - h_n(\tau_k)\tau'_{k,i}, & \text{for } G2R_n \text{ inside NEP} \\ x'_{n,i}(\tau_k^-) + h_n(\tau_k)\tau'_{k,i}, & \text{for } R2G_n \text{ inside NEP} \\ x'_{n,i}(\tau_k^-), & \text{for other exogenous event} \end{cases} \qquad (18)$$

Note that most of the queue content derivative expressions involve the event time derivatives $\tau'_{k,i}$ which we derive next.

Event Time Derivatives. The determination of $\tau'_{k,i}$ depends on each event defined by its associated guard condition. Due to space limitations the complete derivations can be found in [17] and the results are summarized below in two groups by their range of application. The notation $\mathbb{1}_{i=m}$ denotes the indicator function whose value is 1 when $i = m$.

$$\tau'_{k,i} = \begin{cases} \mathbb{1}_{i=2n} + \tau'_{k-1,i}, & \text{if } [z_n \uparrow \theta_n^{max}] \text{ occurs at } \tau_k \\ \mathbb{1}_{i=2n-1} + \tau'_{k-1,i}, & \text{if } [z_n \uparrow \theta_n^{min}] \text{ occurs at } \tau_k \\ \mathbb{1}_{i=n+4}, & \text{if } [w_{n+2} \uparrow \theta_{n+2}] \text{ occurs at } \tau_k \end{cases} \tag{19}$$

$$\tau'_{k,i} = \begin{cases} \frac{\mathbb{1}_{i=n+6} - x'_{n,i}(\tau_k^-)}{\alpha_n(\tau_k) - h_n(\tau_k)}, & \text{if } [x_n \downarrow s_n] \text{ occurs at } \tau_k \\ \frac{\mathbb{1}_{i=n+6} - x'_{n,i}(\tau_k^-)}{\alpha_n(\tau_k)}, & \text{if } [x_n \uparrow s_n] \text{ occurs at } \tau_k \\ \frac{-x'_{n,i}(\tau_k^-)}{\alpha_n(\tau_k) - h_n(\tau_k)} & \text{if } [x_n \downarrow 0] \text{ occurs at } \tau_k \\ 0, & \text{if } [\alpha_n \downarrow 0] \text{ occurs at } \tau_k \end{cases} \tag{20}$$

where all events were defined in Section II, $n = 1, 2$ in (19), $n = 1, \ldots, 4$ in (20) and $i = 1, \ldots, 10$ for both equations.

Cost Derivatives. Similar to the proof in [14], the IPA estimator, i.e., the derivative of $L(\upsilon)$ defined in (14) with respect to υ_i, $i = 1 \ldots 10$, is given by

$$\frac{dL(\upsilon)}{d\upsilon_i} = \frac{1}{T} \sum_{n=1}^{4} \sum_{m=1}^{M_n} \gamma_n \frac{dL_{n,m}(\upsilon)}{d\upsilon_i} \tag{21}$$

where

$$\frac{dL_{n,m}(\upsilon)}{d\upsilon_i} = x'_{n,i}(\xi_{n,m}{}^+)(t^1_{n,m} - \xi_{n,m}) + x'_{n,i}(t^{J_{n,m}}_{n,m}{}^+)$$
$$(\eta_{n,m} - t^{J_{n,m}}_{n,m}) + \sum_{j=2}^{J_{n,m}} x'_{n,i}(t^{j-1}_{n,m}{}^+)(t^j_{n,m} - t^{j-1}_{n,m}) \tag{22}$$

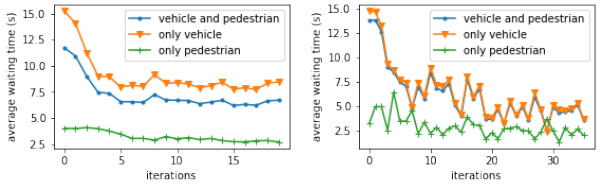
where M_n is the total number of NEPs during the sample path of queue n over $[0, T]$, $J_{n,m}$ is the (observed) total number of events related to queue n in the mth NEP, $t^j_{n,m}$ is the time of the jth event in that NEP, and $\xi_{n,m}$, $\eta_{n,m}$ are the start and end time respectively of of mth NEP.

It is clear from (22) that the IPA derivative is the sum of selected inter-event times multiplied by their corresponding state derivatives. Therefore, the information required to evaluate it consists of: event times, which are easy to observe; state derivatives at these times, which can be obtained from (18); and arrival and departure rates $\alpha_n(\tau_k)$, $h_n(\tau_k)$, needed *only* at certain event times (e.g., when event S_n is induced by light switching in (18)). The latter are easy to estimate, as detailed in the next section. Moreover, we can assume the maximum departure rate $h_n(\tau_k)$ to be a constant which can also be easily estimated offline. In summary, by simply monitoring and recording events as they are observed and very limited calculations, we can obtain the IPA gradient estimator for the expected weighted mean queue length. This is then used with any standard online gradient-based algorithm (15) so as to adjust the controllable parameters and improve the overall performance to a (generally local) optimum. Importantly, the event-driven nature enables our framework to easily generalize to multiple intersections and expanded flow settings (left and right turns). The computational complexity of the TLC is linear in the number of events, so this method inherently scales well as more intersections are added (see [16]).

IV. Simulation Results

The results in this section are based on a simulation environment for traffic through a single traffic light intersection built through Eclipse SUMO. Although the arrival processes can be arbitrary for our IPA-based method, we assume them to be Poisson processes for both vehicles and pedestrians with rates $\bar{\alpha}_n$, and estimate the maximum departure rate as a constant value $h_n = H$ through an offline analysis. Since we only need the arrival flow rate at certain event times, we can estimate an instantaneous arrival rate through $\alpha_n(\tau_k) = N_a/t_w$, where N_a denotes the number of vehicles/pedestrians joining queue n during a time window of size t_w before event time τ_k. We set $H = 1.2$ and equal weight for all flows throughout this section. With this setting, we have performed four sets of simulations.

Optimizing cost. We use the same initial parameter values $\upsilon_0 = [10, 20, 30, 50, 10, 10, 8, 8, 5, 5]$ over different traffic conditions to test how the controller performs. The direction of each update is based on the average gradient, calculated by IPA using 20 sample paths with a length of 1000s each. The results are shown in Table I where the last column shows 33.8% - 62.9% waiting time reductions. Figure 4a shows the cost trajectory for $1/\bar{\alpha} = [6, 6, 10, 20]$ as the number of parameter iterations increases. Note that since the vehicle flow is higher than the pedestrian flow, the overall weighted mean waiting time is dominated by the former. The detailed parameter trajectories can be found in [17].



(a) (b)

Fig. 4: Sample cost trajectories

TLC of an actual intersection. We have cooperated with the town of Veberöd, Sweden to study a major intersection at the town center. This is a typical 4-way intersection with a single lane for each direction. Currently, no traffic light is

TABLE I: Simulation results for different traffic intensities

$1/\bar{\alpha}$	J_{init}	J_{opt}	Cost Reduction
[5,5,20,20]	19.08	12.63	33.8%
[5,8,20,20]	12.58	4.67	62.9%
[6,6,20,20]	11.11	5.68	48.9%
[6,8,20,20]	9.45	3.58	62.1%
[6,6,10,20]	11.72	6.19	47.2%
[6,6,25,20]	11.50	5.00	56.5%

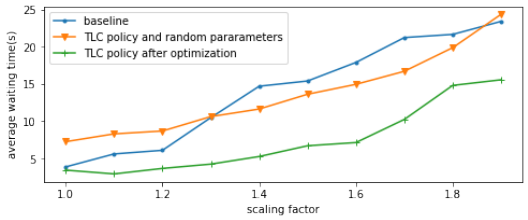


Fig. 5: Comparison of performance measure

present and vehicles and pedestrians follow the first-come-first-leave rule for crossing. During busy hours, this rule can cause long queues and congestion, while also becoming unsafe. Furthermore, traffic is expected to increase in the near future as the town expands and develops. With Veberöd traffic data available, we have modeled the arrival processes to be Poisson with rates $\bar{\alpha} = [0.11, 0.125, 0.01, 0.01]$. We first simulated the intersection operation under current conditions to establish a baseline, and then compared the result to its operation using our controller. Then, we proportionally expand the traffic applying increasing scaling factors, to capture anticipated future town development. The results are shown in Fig. 5, comparing the baseline (no control) to our adaptive quasi-dynamic controller operating with random unoptimized initial parameters, as well as with with those optimized by the IPA gradient-based method. We can see the benefits of the TLC which results in cost decreases varying from 11.13% to 64.22%.

Online TLC implementation. We consider a single long sample path with length $T = 43200s$. IPA parameters are updated every $1200s$ using the data collected during the most recent time window. We set the traffic demand at rates $\bar{\alpha} = [0.154, 0.175, 0.014, 0.014]$ (1.4 times the Veberöd traffic demand), and the same initial parameter as before. Typical sample cost trajectory is shown in Fig. 4b. Observe that cost trajectory fluctuations occur even after parameters have largely converged, indicating, as expected, that the cost is subject to noise due to the random traffic. In order to decrease such fluctuations caused by both traffic demand changes and parameter updates, several smoothing techniques can be applied. Additional results may be found in [17].

TLC adaptivity. We illustrate this property by observing how the TLC performance changes when traffic demand is perturbed. Using the same simulation setting as before, we add traffic perturbations by increasing the Poisson rate of vehicle flow 1 to 1.3 times at $21600s$, and then return to the original rate at $36000s$. The cost trajectory is shown in Fig.6 where the shaded area corresponds to the time interval over which traffic demand was increased and the ability of the TLC to adapt accordingly is seen.

V. Conclusion and Future Work

We have studied a TLC problem for a single intersection with both vehicle and pedestrian flows. We designed a parametric adaptive quasi-dynamic controller aiming at optimizing weighted mean waiting times and used IPA to estimate gradients with respect to these parameters and adjust them iteratively so that it can automatically adapt to changing traffic conditions. Next steps are to add more flows (including bicycle and pedestrian traffic flows) to a traffic network.

References

[1] W.-M. Wey, "Model formulation and solution algorithm of traffic signal control in an urban network," *Computers, environment and urban systems*, vol. 24, no. 4, pp. 355–378, 2000.

[2] M. Van den Berg, A. Hegyi, B. De Schutter, and H. Hellendoorn, "Integrated traffic control for mixed urban and freeway networks: A model predictive control approach," *European journal of transport and infrastructure research*, vol. 7, no. 3, 2007.

[3] Y. Zhang, R. Su, K. Gao, and Y. Zhang, "Traffic light scheduling for pedestrians and vehicles," in *Proc. of IEEE Conf. on Control Technology and Applications (CCTA)*, 2017, pp. 1593–1598.

[4] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transp. Research Part C: Emerging Technologies*, vol. 24, pp. 122–140, Oct. 2012.

[5] H. Wang, M. Zhu, W. Hong, C. Wang, G. Tao, and Y. Wang, "Optimizing Signal Timing Control for Large Urban Traffic Networks Using an Adaptive Linear Quadratic Regulator Control Strategy," *IEEE Trans. on Intelligent Transp. Systems*, vol. 23, no. 1, pp. 333–343, 2022.

[6] Dongbin Zhao, Yujie Dai, and Zhen Zhang, "Computational Intelligence in Urban Traffic Signal Control: A Survey," *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 485–494, Jul. 2012.

[7] Y. Bi, X. Lu, Z. Sun, D. Srinivasan, and Z. Sun, "Optimal Type-2 Fuzzy System For Arterial Traffic Signal Control," *IEEE Trans. on Intelligent Transp. Systems*, vol. 19, no. 9, pp. 3009–3027, 2018.

[8] T. Chu, J. Wang, L. Codeca, and Z. Li, "Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control," *IEEE Trans. on Intelligent Transp. Systems*, vol. 21, no. 3, pp. 1086–1095, 2020.

[9] M. M. Ishaque and R. B. Noland, "Multimodal Microsimulation of Vehicle and Pedestrian Signal Timings," *Transp. Research Record*, p. 8, 1939.

[10] W. Ma, D. Liao, Y. Liu, and H. K. Lo, "Optimization of pedestrian phase patterns and signal timings for isolated intersection," *Transp. Research Part C: Emerging Technologies*, vol. 58, pp. 502–514, 2015.

[11] L. d. C. Gomes and L. H. M. K. Costa, "Traffic Light Optimization for Vehicles and Pedestrians through Evolution Strategies," in *IEEE 95th Vehicular Technology Conf.*, Helsinki, Finland, 2022, pp. 1–7.

[12] Y. Zhang, Y. Zhang, and R. Su, "Pedestrian-Safety-Aware Traffic Light Control Strategy for Urban Traffic Congestion Alleviation," *IEEE Trans. on Intelligent Transp. Systems*, vol. 22, no. 1, pp. 178–193, 2021.

[13] U. K. Sreekumar, R. Devaraj, Q. Li, and K. Liu, "Real-Time Traffic Pattern Collection and Analysis Model for Intelligent Traffic Intersection," in *IEEE Intl. Conf. on Edge Computing*, 2018, pp. 140–143.

[14] J. L. Fleck, C. G. Cassandras, and Y. Geng, "Adaptive Quasi-Dynamic Traffic Light Control," *IEEE Trans. on Control Systems Technology*, vol. 24, no. 3, pp. 830–842, May 2016.

[15] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and C. Yao, "Perturbation Analysis and Optimization of Stochastic Hybrid Systems," *European Journal of Control*, vol. 16, no. 6, pp. 642–661, Jan. 2010.

[16] Y. Chen and C. G. Cassandras, "Scalable adaptive traffic light control over a traffic network including transit delays," in *Proc. of IEEE 26th Intelligent Transp. Systems Conf.*, 2023, pp. 1651–1656.

[17] ——, "Adaptive traffic light control for competing vehicle and pedestrian flows," *arXiv preprint arXiv:2303.08173*, 2023.

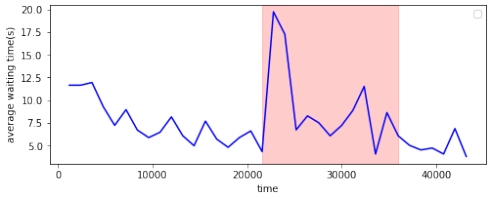


Fig. 6: Sample cost trajectory with demand perturbation