IFAC Journal of Systems and Control 12 (2020) 100087

Contents lists available at ScienceDirect

IFAC Journal of Systems and Control

journal homepage: www.elsevier.com/locate/ifacsc

Using time-inconsistent wait-time functions for cycle-free coordinated traffic intersections☆

Evan Gravelle *, Sonia Martínez

Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093, USA

ARTICLE INFO

Article history:
Received 24 September 2018
Received in revised form 31 January 2020
Accepted 16 April 2020
Available online 21 April 2020

Keywords:
Networked systems
Network stability
Traffic control

ABSTRACT

Sensing and computing capabilities of modern traffic intersections have greatly improved in recent years, but current control policies do not fully utilize these capabilities. In this paper, we present a novel intersection control algorithm based on an objective function that accounts for drivers' time preferences. In particular, the intersection places greater importance on a vehicle which has been waiting at an intersection over one that just arrived. Coordination between intersections is achieved through an added term to the objective function using the green wave idea. Under this policy given a macroscopic dynamical model, we provide a sufficient condition for the controller to maintain uniformly bounded weighted queues at an intersection given sufficiently small spawn rates. We test our algorithm and results extensively in a realistic microscopic simulation, through measuring queue stability and various performance metrics.

1. Introduction

Motivation. Modern advancement of technology combined with the rapid decrease in cost of processors has exponentially increased the capabilities of common infrastructure systems over recent decades. One such system is the traffic intersection manager, which controls the phases of a traffic light, allowing vehicles in certain lanes to pass safely through the intersection. The estimation of the state of traffic near intersections together with the use of adaptive control algorithms can lead to significant improvements in traffic conditions and quality of service. Most modern intersections use loop detectors in the road to approximate the position, velocity, and/or density of vehicles and to make control decisions. However, loop detectors are expensive to install and maintain, a significant fraction of loop detectors are not functioning on a given day (Payne & Thompson, 1997; Rajagopal & Varaiya, 2007), and they only provide data at discrete points on the road. Improved camera quality as well as detection and tracking algorithms has opened the door for traffic intersections to stop relying on loop detectors and start using cameras, offering a richer and more robust source of information. Motivated by this scenario, we design a novel intersection control policy utilizing the position and velocity information of each vehicle to enable a more intelligent policy.

Literature review. One common algorithm for intersection control is SCOOT, or Split Cycle Offset Optimization Technique (Hunt, Robertson, Bretherton, & Winton, 1981). This controller adjusts the split, cycle, and offset parameters of the cyclical sequence of phases to accommodate instantaneous traffic demands as read by loop detectors. However, there are a few problems with this formulation which many alternatives also fail to address (Keong, 1993; Lowrie, 1982). One issue is that the parameters which SCOOT optimizes perturb the cycle in a limited fashion, so SCOOT is quite constrained and efficiency is lost. Another limitation is given by the assumption that each vehicle is given equal priority, which is implicit in the objective. We claim that a better objective accounts for the time that a vehicle has been waiting so that the relative importance of a vehicle which has been waiting is more than a vehicle which has just arrived. Intuitively this idea makes sense as a mechanism to ensure that every vehicle eventually crosses the intersection. Studies have also shown that humans value time inconsistently (Laibson, 1998) and that uncertainty in travel time is costly (Noland & Small, 1995), both of these issues are mitigated by this new objective.

Some more recent works have extended these classical algorithms. Backpressure routing techniques are applied to traffic systems using fixed time slots without considering phase transition times in Wongpiromsarn, Uthaicharoenpong, Wang, Frazzoli, and Wang (2012). Linear–quadratic regulator theory is applied in Diakaki, Papageorgiou, and Aboudolas (2002) with an emphasis on heavily congested conditions, which does not consider the

☆ This work was supported by NSF-CMMI, USA 1434819.
* Corresponding author.
E-mail addresses: egravell@ucsd.edu (E. Gravelle), soniamd@ucsd.edu (S. Martínez).

https://doi.org/10.1016/j.ifacsc.2020.100087
2468-6018/© 2020 Published by Elsevier Ltd.

effect of offset of consecutive junctions. A multi-objective linear programming approach which considers light switchover dates in a time horizon is described in Dujardin, Vanderpooten, and Boillot (2015). A genetic algorithm is applied to maximize traffic flow using a graphics processing unit in Shen, Wang, and Zhu (2011). Work has also been done on bandwidth maximization using variable speed limits (De Nunzio, Gomes, de Wit, Horowitz, & Moulin, 2016) and partitioning techniques (Tian & Urbanik, 2007). However, none of these works fully utilize modern sensing technology in a flexible optimization setting, which can be used to e.g. prioritize those who have been waiting longer times.

Statement of contributions. In this paper, we devise a traffic intersection policy to minimize an objective function that accounts for time-inconsistent waiting-time preferences. These preferences have similar properties to hyperbolic discount functions, providing a simple model for human rewards, and seem to be effective to control traffic at intersections. In particular, the objective considers incoming vehicles from neighboring intersections, and places additional weight on a vehicle which has been waiting some time, this is representative of human preferences. We consider a class of policies that can be effective for controlling traffic at intersections and keep traffic bounded at intersections. We provide a sufficient condition on the intersection switching rule using a macroscopic flow model that ensures uniform bounded weighted queue lengths under a sufficiently small vehicle spawn rate. We also add a term to the objective function to enable cooperation between intersections with the goal of increasing overall efficiency. In simulation, we model vehicle agents after real drivers: a driver attempts to drive its route at the speed limit while following traffic laws and maintaining adequate distance between it and other vehicles. We simulate our intersection controller policy with the more complex vehicle agents on a system with two intersections and show the benefits of the new control algorithm. In particular, we verify the bounded weighted queue result for the complex model, and we show how the policy effectively minimizes the total squared waiting time or human impatience at the intersection. Simulation results verify how our policy is more effective in bounding weighted queues than a two alternatives and show the effect of the coordination term on various performance indices. This work is an extension of Gravelle and Martínez (2017), containing the full proof of the queue stability theorem, an extended remark, and additional simulation figures and discussion.

Organization. This paper is organized as follows. Section 2 contains the problem formulation with motivation and the vehicle behavior model. Section 3 contains the queue stability theorem and proof under a simplified macroscopic model. Section 4 contains simulation details and numerical and plotted results. Finally, Section 5 contains a summary of our work and a discussion of future work.

2. Problem formulation

In this section, we formulate the intersection control problem by defining the objective function and control variables. The goal of the intersection policy is to minimize this objective function which characterizes travel time of all vehicles while ensuring that vehicles rarely have to wait long times at a given intersection (see Table 1).

Single intersection problem

An intersection I is a junction of road segments of given length $L_I > 0$. These roads can have any number of lanes and feasible allocation of turn lanes. We denote by N the total number of vehicles that flow into the intersection system over a time horizon, $[0, H]$, which is discretized as $h \equiv k\Delta h$, for $k \in \mathbb{N}_0$ where $\mathbb{N}_0$ is the set of natural numbers including zero and $\Delta h > 0$ is a small discretization step. In what follows, we denote by $\mathcal{U}$ the set of phases u by which a conflict-free set of lanes is selected at a given time. For example, one possible phase would allow all vehicles traveling straight through the intersection northbound or southbound to pass through the intersection safely, while all other vehicles would be prohibited. For ease of notation and without loss of generality, we consider $\mathcal{U} \triangleq \{N_lS_l, N_lN_s, N_sS_l, S_lS_s, W_lE_l, W_lW_s, W_sE_l, E_lE_s\}$, where the base variable represents whether the road originates from north, south, east, or west, and the subscript denotes left-turn lanes (l) or straight/right lanes (s).

Table 1
Table of symbols.

I	Intersection
L_I	Intersection road length [m]
N	Total number of vehicles
H	Time horizon [s]
Δh	Discrete time step [s]
$\mathcal{U}$	Set of intersection phases
u	Phase given by a pair of conflict free lanes
i	Vehicle identity
$\hat{t}_i$	Extra wait time (delay) at I for vehicle i [s]
w	Wait-time function to weight extra wait times
ℓ	Lane
η	Intersection switch threshold
g_u^I	Minimum phase time [s]
n_u^I	Minimum number of vehicles through I
Q_u^I	Number of waiting vehicles
t_{gap}	Vehicle safety buffer [s]
L_u^I	Maximum path length through intersection
p	Platoon
ξ_p^I	Time for platoon to cross intersection [s]
z_p^I	Time quantifying platoon state at I [s]
E_ℓ^I	Number of vehicles approaching I
T_p^I	Time of arrival of platoon at I [s]

Let $i \in \{1, \dots, N\}$ denote a vehicle passing through the intersection within the time horizon H. As a first approximation to the intersection problem, consider the cost function:

$$f(\hat{t}_i) = \sum_{i=1}^{N} \left(\Delta t_i + w(\hat{t}_i)\right).$$

The variable Δt_i (with $\Delta t_i \approx k_i\Delta h$, for some k_i) is the time that vehicle i would spend at the junction if it were to move through the system without stopping, $\hat{t}_i$ (with $\hat{t}_i \approx \hat{k}_i\Delta h$, for some $\hat{k}_i$) is extra time spent by vehicle i due to the delay in I, and $w : [0, +\infty) \to [0, +\infty)$ is a class $\mathcal{K}_\infty$ weighting function which penalizes long delays.[1] The previous cost can be understood as a function of a sequence of phases $(u_1, \dots, u_M) \in \mathcal{U}^M$, for some integer M such that $M\Delta h \le H$. The extra time $\hat{t}_i$ is incremented when $v_i \le \gamma v_f$, where v_i is the speed of vehicle i, $\gamma \in [0, 1)$ is the wait threshold which defines when a vehicle is waiting, and v_f is the speed limit.

We assume that the intersection can track vehicles which are in lanes moving towards it within some distance L, so it knows time-of-sighting (measured in its time clock t_{sys}), position, and velocity of each vehicle. Predicting Δt_i for all vehicles is difficult because the dynamics of each vehicle must be calculated forward in time until they leave the system, and this is intractable due

[1] Recall that a continuous function $f : [0, \infty) \to [0, \infty)$ is said to belong to class $\mathcal{K}_\infty$ if it is strictly increasing, $f(0) = 0$, and $\lim_{r\to\infty} f(r) = \infty$.

to requiring full forward simulation to solve. Thus, we simplify the original cost function above by proposing a simpler objective in that does not require explicit travel time estimates. More precisely, we consider only the current time step, and focus on the current cost over $u \in \mathcal{U}$ weighting the delays of each lane to determine which directions to allow through. In other words, at each instance of time h, we consider:

$$\mathcal{J}_1(u) \triangleq \sum_{\ell \in u} w_\ell(\hat{t}_i) \triangleq \sum_{\ell \in u} \sum_{i \in \ell} w(\hat{t}_i). \tag{1}$$

Note that there is an implicit dependence of J on u_0, the previously implemented phase, and that w_ℓ is the sum of weights of vehicles in lane ℓ which are allowed through in phase u. This problem is solved at every discrete time instance $h \geq 0$, to determine a phase switch u. However, a phase switch from $u = u_0$ to $u = u_1$ can only be implemented when the set of feasible $u \neq u_0$ is non-empty at time h, or, in other words, for those u for which the following hold.

First, a substantial difference in lane weights is required to switch to a new phase due to delays associated with changing phases including yellow lights and vehicle acceleration time. Given a previous u_0, the set of feasible $u \in \mathcal{U}$ at time h are those for which

$$\sum_{\ell \in u} w_\ell(\hat{t}_i) > \eta \sum_{\ell \in u_0} w_\ell(\hat{t}_i), \tag{2}$$

holds, for some $\eta > 1$, where η is a tuning parameter called the switch threshold. In other words, this is a constraint to the cost function in (1). Intuitively, this constraint allows a phase to switch at time h only if the delays of waiting vehicles are significantly larger than the current phase.

In addition, we require a lower bound on the time spent in a specific phase u_0 to consider the time it takes for vehicles to accelerate from rest. We define this bound based on allowing a certain number of vehicles through the intersection from rest. We define n_ℓ^I as the minimum number of desired vehicles to let through at each lane ℓ at intersection I. Then, $n_u^I = \max_{\ell \in u} n_\ell^I$ is the minimum number of desired vehicles to allow through during phase u, based on which lanes will be allowed through by u. We define a queue length Q_ℓ^I at a lane ℓ entering into an intersection I as the number of vehicles waiting in ℓ to cross the intersection. With knowledge of the queue lengths at each lane Q_ℓ^I, define $Q_u^I = \max_{\ell \in u} Q_\ell^I$. If h_0 was the time at which u_0 began, and $h - h_0$ is the time spent in u_0, we consider the following constraint on u. That is, in order for u to be feasible at time h, it should satisfy

$$h - h_0 \geq g_u^I \triangleq (\min\{Q_u^I, n_u^I\} - 1)t_{\text{gap}} + \sqrt{\frac{2L_u^I}{a}}, \tag{3}$$

where t_{gap} is the time between vehicles moving safely at maximum velocity, L_u^I is the maximum length of the path through the intersection during phase u, and a is the average acceleration of vehicles. Note that $g_u^I \geq 0$. Intuitively, this constraint uses the time needed by a desired number of vehicles n_u^I to make it through the intersection safely. In all, a nonlinear optimization problem is given by the minimization of $\mathcal{J}_1(u)$, over the decision variable $u \in \mathcal{U}$, subject to the constraints (2) and (3) at time h. This problem is to be solved at every h to decide whether a phase change is implemented at h.

Multiple intersection problem

Given a system of multiple intersections, a naive extension to the previous setting would partition the roads between the intersections and assign each part to the nearest intersection, serving to decouple the intersections and localize their control. To avoid a complete decoupling, which could have detrimental effects on joint performance, we extend the one-intersection problem formulation by adding a term that can enable intersection cooperation using the idea of *green wave* (De Nunzio et al., 2016). A green wave results from the coordinated offset of green traffic lights along a road, which allows a group of adjacent vehicles to this road to pass through the intersections without stopping. We refer to a group of vehicles that can pass through multiple intersections as a platoon. By using limited communications between other intersections and vehicles, an intersection first identifies a group of vehicles or platoon that can benefit from synchronization, then it coordinates phase switching with other intersections. Current methods which attempt to maximize platoon sizes usually require a fixed cycle length. To incorporate this idea in a more flexible non-cyclic framework, we introduce a coupling term.

The problem now is extended to

$$\begin{aligned} \underset{u \in \mathcal{U}}{\text{minimize}} \quad & \mathcal{J}_2(u) \triangleq \sum_{\ell \in u} \left(w_\ell(\hat{t}_i) + B_\ell \right), \\ \text{subject to} \quad & u \in \mathcal{P}, \end{aligned} \tag{4}$$

where B_ℓ is a term that exploits the knowledge of future vehicles either from intersections or just from additional sensors to plan ahead, and $\mathcal{P}$ is the set of phases which satisfies the constraints in Eqs. (2) and (3). We describe the term B_ℓ more precisely after Eq. (5). We assume that intersections have synchronized clocks, and that an intersection I will receive some information about when vehicles will be arriving in the future either from the vehicles themselves or from another intersection. For intersection I, these information packets take the form $\{\ell_p^I, E_p^I, T_p^I\}$, where p denotes a platoon and ℓ_p^I, E_p^I, T_p^I are the lane, the number of vehicles coming, and expected time of arrival of the platoon p, respectively. Knowledge of the transition time between each phase is required, which is simply the time a light must stay in yellow before transitioning to red, we denote this as Y_p^I. We define ξ_p^I as the amount of time required for all vehicles in platoon p to cross the intersection I, and we let $z_p^I = t_{\text{sys}} + Y_p^I + S - T_p^I$ represent a time difference, where t_{sys} is the current time measured by the clocks at each intersection (which is common to all of them), and S is the average vehicle's time to stop from full speed. Thus, $t_{\text{sys}} + Y_p^I + S$ marks the time instant at which vehicles stopped at I after the traffic light for lane ℓ_p was set to red, and z_p^I measures the time difference between vehicles stopping and the new platoon of vehicles arriving at the intersection. In this way, $z_p^I = 0$ means that a platoon p just arrives at a time T_p^I that is equal to the time $t_{\text{sys}} + Y_p^I + S$ at which the light turns green. If $z_p^I < 0$ this means that the platoon arrives just before the green light, while $z_p^I > 0$ means the platoon arrives just after the green light.

With knowledge of this data, a benefit is achieved if the light is still green when the first vehicle arrives, and stays green for the whole platoon to make it through the intersection. With this, define

$$B_\ell = \alpha E_\ell \max\left\{0, \min\left\{\frac{z_p^I}{\xi_p^I} + 1, \min\left\{1, \frac{g_u^I}{\xi_p^I}\right\}, \frac{g_u^I - z_p^I}{\xi_p^I}\right\}\right\}, \tag{5}$$

where $\ell = \ell_p$, α is a control parameter determining how heavily to consider the platoons, see Fig. 1. Inside the minimum, recall that g_u^I is defined as the right-hand side of constraint (3). In addition, $\frac{z_p^I}{\xi_p^I} + 1 \leq 1$ is the fraction of the platoon which can go through the intersection when the platoon arrives early (before switching to red), $\min\{1, \frac{g_u^I}{\xi_p^I}\}$ characterizes the efficiency of the platoon crossing during the green, and $\frac{g_u^I - z_p^I}{\xi_p^I}$ is the fraction of

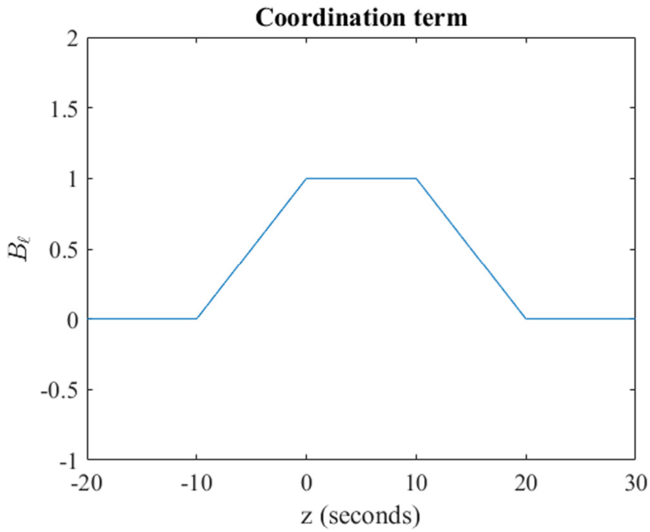

Fig. 1. Shape of B_ℓ, which is maximized when the platoon overlaps completely with a green light and is zero when there is no overlap.

the platoon which can benefit from a switch to green when the platoon arrives late. Intuitively, B_ℓ is minimized (at a value of 0) and therefore provides the most benefit when the green phase coincides with the arrival of vehicles (that is, the inside minimum is attained at $\frac{z_p^I}{\xi_p^I} + 1 = 0$ or at $g_u^I = z_p^I$), and is maximized (at a value of 1) and provides no benefit when the green phase has no overlap with vehicle arrival. In any case, the benefit of switching the light at green when the platoon arrives is limited by how much the phase is active, which is determined by the parameter g_u^I. See Fig. 1 for an illustration.

Vehicle behavior model

In our subsequent simulations, we modeled vehicle agents' behavior after real drivers, according to the following rules:

(i) A vehicle has a fixed path through the system, from origin at a boundary to destination at a boundary.
(ii) Lane changes are not allowed.
(iii) A vehicle accelerates uniformly up to the speed limit unless:
- there is risk of collision with another vehicle, with an added safety buffer,
- there is a red or yellow light and the vehicle can safely stop in front of an intersection,

in which the vehicle decelerates uniformly.

Wait-time function

The (weighted) wait-time function w models the wait-time of drivers at the intersection. The function we consider here is a type of hyperbolic, time-inconsistent discount function in reward maximization problems (Laibson, 1998). Unlike exponential discounting, these functions reflect the human preference to choose smaller-but-sooner rewards over larger-but-later rewards as the delay occurs sooner rather than later in time. These human-preference functions were validated in the Psychology and Experimental Psychology fields (Green, Fry, & Myerson, 1994; Kirby, 1997; Sheffer et al., 2016), and have since then been a

Algorithm 1: Intersection Control Algorithm for Intersection I

Initialize queue lengths $Q \leftarrow \mathbf{0}$, phase u, min green time per phase g_u, yellow time y, current phase $\bar{u}$, switch threshold s, switch time $h_s \leftarrow \infty$
Define $\mathcal{U}$ as set of phases and $\mathcal{N}_u$ as set of phases which share a direction with u
for *each time* h **do**
  if $h - h_s > g_u$ **then**
    Calculate weights in each lane w_ℓ
    Calculate weights in each phase w_u
    if $\max_{u\in\mathcal{U}} w_u - w_{\bar{u}} > \eta w_{\bar{u}}$ **then**
      $\bar{u} \leftarrow \operatorname{argmax}_{u\in\mathcal{U}} w_u$
      $Z \leftarrow 1$
      $t_s \leftarrow t$
    else if $\max_{\tilde{u}\in\mathcal{N}_u} w_{\tilde{u}} - w_{\bar{u}} > 0.5\eta w_{\bar{u}}$ **then**
      $\bar{u} \leftarrow \operatorname{argmax}_{\tilde{u}\in\mathcal{N}_u} w_{\tilde{u}}$
      $Z \leftarrow 1$
      $t_s \leftarrow t$
    end
    if $Z = 1$ **then**
      Send packet of data D containing source intersection I, Q_ℓ for all lanes ℓ leading to neighboring intersection J, and expected time of arrival T
      $Z = 0$
    end
  end
  $h \leftarrow h + \Delta h$
end

basic premise in Behavioral Economics (Frederick, Loewestein, & O'Donoghue, 2002; Laibson, 1997, 1998). Subsequently, such models have also been considered in Management and Queuing Theory (Hassin, 2016; Plambeck & Wang, 2013). Hyperbolic discount functions in reward maximization problems take the form $f(\hat{t}) = \frac{1}{\alpha+\phi\hat{t}}$, where α and ϕ are some parametric coefficients, and $\hat{t}$ is the delay. In our cost-minimization setting, this translates into wait functions of the form $w(\hat{t}) = \alpha + \phi\hat{t}$. With a similar property to these, we consider functions of the form

$$w(t) = \phi t^2 + \alpha t + \beta, \tag{6}$$

and, for simplicity in the computations and simulations that follow, $w(t) = \phi t^2$ for some constant $\phi > 0$. These functions preserve the hyperbolic discounting property, while performing better than purely hyperbolic functions in maintaining bounded weighted queues in traffic simulations. Additionally, the functions ensure that no vehicle is stuck indefinitely, and should help reduce the average travel time variance compared to a linear model.

3. Queue stability

In this section, we prove weighted queue length stability under a simplified model. Weighted queue length for road r (consisting of a set of lanes) is denoted as $w_r \triangleq \sum_{i\in r} w(\hat{t}_i)$, and recall that the queue length, Q_r is the number of cars waiting at r to cross the intersection. System stability is achieved when all lane queue lengths remain bounded below some value such that each queue does not spillover into other intersections.

The following assumptions were made to construct the new model:

Assumption 3.1 (*Traffic Model*)**.** Traffic flow follows a continuous-time macroscopic model where change in vehicle density on some part of the road is proportional to the net flow rate, or $\rho(t+\Delta t)=\rho(t)+\Delta t(q_{\rm in}-q_{\rm out})$. The traffic density of a lane into an intersection is proportional to its queue length. There is a constant flow rate $q_{{\rm in},r}=q_{\rm in}$ into each non-outbound road $r\in\{1,\dots,4\}$ from the boundary. Given a queue of vehicles with the freedom to move, the flow rate at the front of the queue is a constant $q_{\rm out}>q_{\rm in}$.

Assumption 3.2 (*Two-phase Intersections*)**.** There are two intersection phases, each allowing bi-directional flow on the two roads, one running north–south and the other east–west. Each phase has minimum duration of $g_{\rm NS}, g_{\rm EW}$, respectively. Flow begins immediately when a new phase begins.

Assumption 3.3 (*Linear Wait-time*)**.** The average time that a vehicle has been waiting at an intersection is proportional to the number of vehicles in the queue, $n_r(t)$, at time t. This implies $w_r(t)\propto n_r^2(t)$ for road r.

We formed a simpler dynamical model and applied the following Intersection Control algorithm to solve the optimization problem given by Eqs. (1)–(3) and described in detail in Algorithm 1.

At each instant of time h, one has to check whether there is a feasible phase change satisfying (1)–(3). If so, an evaluation of the cost function $\mathcal{J}_1$ over all possible feasible phase changes is performed and then the best phase is chosen. Given that the number of phase changes is small and finite, this approach is practical, and complexity does not increase with network size. The following theorem contains the stability result associated with this Intersection Control approach:

Theorem 3.4. *Given Assumption 3.1 (Traffic model) with constant flow-rate parameters $q_{in,r}$, $\forall r\in\{1,\dots,4\}$ and q_{out}, Assumption 3.2 (Two-phase intersections), Assumption 3.3 (Linear wait-time), and sufficiently small spawn rate $q_{in,r}=q_{in}$, $\forall r\in\{1,\dots,4\}$, any finite switch threshold $\eta>1$ will ensure that the Intersection Control approach to optimize the cost (1) under constraints (2) and (3) at each time maintains uniformly bounded weighted queues in all lanes for all times.*

Interpreting this result leads to the following intuitive explanation: η is directly related to T, the time elapsed between two consecutive phase switches, and as long as η is small enough so that the growing weighted queues remain bounded and large enough so that the minimum green time does not affect the dynamics, the system is stable.

Proof. Road indices 1, 2, 3, 4 indicate north, east, south, west, respectively, relative to the center of the intersection. For now assume initial conditions such that the north-south road has a green light $u(0)=[1,3]$, n_r is the number of vehicles on road r, $w_{\rm NS}(0)=\eta w_{\rm EW}(0)$, and when a switch occurs at time T, $T\ge g_{\rm NS}$. Given any initial phase $u(0)$, a cycle with duration $t_{\rm cycle}>0$ occurs when $u(t_{\rm cycle})=u(0)$ with $u(\bar t)\neq u(0)$ for some $\bar t\in(0,T)$. Queue lengths will remain bounded if, $n_r(t+t_{\rm cycle})\le n_r(t)$, $\forall r\in\{1,\dots,4\}$. Define $w_{\rm EW}(t)=w_1(t)+w_3(t)$ and $w_{\rm NS}(t)=w_2(t)+w_4(t)$. Then due to Assumption 3.3 (Linear wait-time),

$$w_{\rm EW}(t)=\frac{n_2(t)^2}{2q_{{\rm in},2}}+\frac{n_4(t)^2}{2q_{{\rm in},4}}=\frac{(n_2(0)+q_{{\rm in},2}t)^2}{2q_{{\rm in},2}}+\frac{(n_4(0)+q_{{\rm in},4}t)^2}{2q_{{\rm in},4}},$$

where

$$n_r(0)=\sqrt{2q_{{\rm in},r}w_r(0)},\ \forall r\in\{1,\dots,4\}.$$

Simplifying,

$$w_{\rm EW}(t)=w_2(0)+w_4(0)+\left(\sqrt{2q_{{\rm in},2}w_2(0)}+\sqrt{2q_{{\rm in},4}w_4(0)}\right)t+\frac{q_{{\rm in},2}+q_{{\rm in},4}}{2}t^2.$$

Now define $\Delta q_r=q_{{\rm in},r}-q_{\rm out,r}$, $\forall r\in\{1,\dots,4\}$. Computing $w_{\rm NS}(t)$,

$$w_{\rm NS}(t)=\frac{n_1(t)^2}{2q_{{\rm in},1}}+\frac{n_3(t)^2}{2q_{{\rm in},3}}=\frac{(n_1(0)+\Delta q_1 t)^2}{2q_{{\rm in},1}}+\frac{(n_3(0)+\Delta q_3 t)^2}{2q_{{\rm in},3}}.$$

Simplifying,

$$w_{\rm NS}(t)=w_1(0)+w_3(0)+\left(\Delta q_1\frac{\sqrt{2w_1(0)}}{q_{{\rm in},1}}+\Delta q_3\frac{\sqrt{2w_3(0)}}{q_{{\rm in},3}}\right)t+\left(\frac{\Delta q_1^2}{2q_{{\rm in},1}}+\frac{\Delta q_3^2}{2q_{{\rm in},3}}\right)t^2.$$

One condition guaranteeing the previous definition of bounded weighted queues is $w_{\rm EW}(\tilde T)=w_{\rm EW}(0)$ and $w_{\rm NS}(\tilde T)\le w_{\rm NS}(0)$ at some time $\tilde T$. Due to the symmetry of the two phases and the immediate flow condition after a switch of Assumption 3.2, an equivalent condition is $w_{\rm EW}(T)=w_{\rm NS}(0)$ and $w_{\rm NS}(T)\le w_{\rm EW}(0)$ for some time T. Assume the phase changes when $w_{\rm EW}(T)=\eta w_{\rm NS}(T)$, so after applying these equations, we get

$$0=w_2(0)+w_4(0)-\eta w_{\rm NS}(T)+\left(\sqrt{2q_{{\rm in},2}w_2(0)}+\sqrt{2q_{{\rm in},4}w_4(0)}\right)T+\frac{q_{{\rm in},2}+q_{{\rm in},4}}{2}T^2, \tag{7}$$

$$0\ge w_1(0)+w_3(0)-w_{\rm EW}(0)+\left(\Delta q_1\frac{\sqrt{2w_1(0)}}{q_{{\rm in},1}}+\Delta q_3\frac{\sqrt{2w_3(0)}}{q_{{\rm in},3}}\right)T+\left(\frac{\Delta q_1^2}{2q_{{\rm in},1}}+\frac{\Delta q_3^2}{2q_{{\rm in},3}}\right)T^2. \tag{8}$$

Plugging in initial conditions to Eq. (7) gives

$$0=(1-\eta^2)(w_2(0)+w_4(0))+\left(\sqrt{2q_{{\rm in},2}w_2(0)}+\sqrt{2q_{{\rm in},4}w_4(0)}\right)T+\left(\frac{q_{{\rm in},2}+q_{{\rm in},4}}{2}\right)T^2.$$

Define

$$A_1=\left(\frac{q_{{\rm in},2}+q_{{\rm in},4}}{2}\right),\quad B_1=\left(\sqrt{2q_{{\rm in},2}w_2(0)}+\sqrt{2q_{{\rm in},4}w_4(0)}\right),\quad C_1=(1-\eta^2)(w_2(0)+w_4(0)),$$

then

$$T=\frac{-B_1+\sqrt{B_1^2-4A_1C_1}}{2A_1}, \tag{9}$$

and because $\eta>1$, $A_1>0$, $B_1>0$, and $C_1<0$, a real positive solution exists for T. This quantity represents how much time passes before $w_{\rm EW}(T)=w_{\rm NS}(0)$.

Eq. (8) is satisfied when

$$\begin{aligned} T &\le \frac{-B_2 - \sqrt{B_2^2 - 4A_2C_2}}{2A_2}, \\ A_2 &= \left(\frac{\Delta q_1^2}{2q_{\text{in},1}} + \frac{\Delta q_3^2}{2q_{\text{in},3}}\right), \\ B_2 &= \left(\Delta q_1 \frac{\sqrt{2w_1(0)}}{q_{\text{in},1}} + \Delta q_3 \frac{\sqrt{2w_3(0)}}{q_{\text{in},3}}\right), \\ C_2 &= (\eta - 1)(w_2(0) + w_4(0)). \end{aligned} \tag{10}$$

Due to the fact that $\Delta q_r < 0$, $\forall r \in \{1, \dots, 4\}$ from Assumption 3.1 (Traffic model), $A_2 > 0$, $B_2 < 0$, and $C_2 > 0$, the solution to Eq. (10) is real and positive when

$$B_2^2 - 4A_2C_2 \ge 0,$$
$$B_2 + \sqrt{B_2^2 - 4A_2C_2} < 0,$$

both of which are satisfied when $q_{\text{in},r}$ is small enough, for all $r \in \{1, \dots, 4\}$. By a continuity argument, by taking an η sufficiently close to 1, the solution to Eq. (9) (which can be made arbitrarily close to 0 by taking an η sufficiently close to one) will also satisfy the bound in (10).

This argument can be repeated for each future switch by simply replacing instances of w_{EW} with w_{NS} and vice versa, due to $q_{\text{in},r} = q_{\text{in}}$, $\forall r \in \{1, \dots, 4\}$. Now relaxing the assumption on initial conditions, in finite time one of two scenarios can occur. One is, the light will switch due to the weights so either $w_{\text{NS}}(T) = \eta w_{\text{EW}}(T)$ or $w_{\text{EW}}(T) = \eta w_{\text{NS}}(T)$, this is treated as the new initial condition and the argument holds. The second scenario is that the light will switch at g_{NS} due to the green time constraint, and $w_{\text{EW}}(g_{\text{NS}}) = w_{\text{EW}}(T) + \delta$. However, from here either a switch will occur due to the weights or a switch will occur due to the constraint from Eq. (3), either case will results in queues bounded by $w_{\text{EW}}(T) + \delta$, and this argument can be propagated forward in time as well. □

Remark 3.5 (*On Theorem 3.4 Assumptions*)**.** If Assumption 3.1 (Traffic model) is relaxed and $q_{\text{in},r}(t) \in [q_{\min}, q_{\max}]$, $\forall r \in \{1, \dots, 4\}, \forall t$, then the solution to Eq. (9) is bounded (due to convexity) by the solutions to (9) with $q_{\text{in},r} = q_{\min}$ and $q_{\text{in},r} = q_{\max}$, $\forall r \in \{1, \dots, 4\}$. Similarly, if q_{out} must ramp up to the maximum value from 0, a similar argument can be made, leading to similar results and overall stability but with tighter bounds on η.

If we relax Assumption 3.2 (Two-phase intersections), the same ideas and mathematical principles can be applied to the more complex case where there are four possible phases (left–left, left–straight, left–straight, straight–straight) instead of one (straight–straight) in each direction. In the multiple intersection case, each individual intersection can be viewed as discussed previously, but now with a perturbation in the form of the cooperation term B_ℓ. This perturbation is bounded as seen in Eq. (5), and in the worst case where many lanes have long queues, the perturbation will be small compared to the quadratic wait-time costs, so we expect bounded weighted queues in these scenarios as well.

Regarding Assumption 3.3 (Linear wait-time), we note that it imposes an implicit assumption on the behavior of our Intersection Control Policy. However, we believe that it is possible to relax this assumption to a class $\mathcal{K}_\infty$ polynomial relationship between the wait-time function $w(t)$ and number of cars on a road $n_r(t)$. We provide an informal argument for this fact in the following. Similarly to the theorem proof, Eqs. (7) and (8) would be replaced by polynomial equations in T of a power k. In an analogous way to (7), we would have a polynomial with independent term given by $(1-\eta)\alpha$, for some $\alpha > 0$, and a positive leading coefficient for T^k proportional to the k^{th} power of $q_{\text{in},2}$, $q_{\text{in},4}$. By a Descartes' rule of signs on the coefficients of a polynomial, given that $\eta > 1$, we can ensure that there is at least a positive root of the polynomial. This root can be made as small as possible by taking η as close to one as possible. A more careful examination of the coefficients of an inequality similar to (8), with the condition $\Delta q_i < 0$, for $i = 1, 2$, and the application of Descartes' rule should ensure the existence of another positive root for a choice of a sufficiently small η.

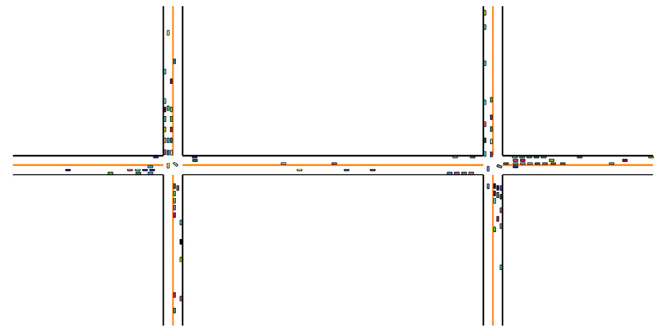

Fig. 2. Image of microscopic model.

4. Simulation results

The agent-based vehicle dynamics model is used in simulations along our Intersection Control policy described in Algorithm 1. However, we employ it to solve problem (4), which extends the formulation of problem (1)–(3). An intersection knows which lanes are associated to which phases, and has parameters fixed. At each time step, the weights are calculated in each lane, and these weights are associated to different phases. If there exists another phase with η times the weight of the current phase, a transition occurs to this phase. If there exists another phase which shares a direction with the current phase and has $\eta/2$ times the weight of the current phase, a partial transition occurs to this phase.

Simulations were run on single intersections and systems of two intersections, see Fig. 2. We define the discrete time step $\Delta h = 0.1$. The safety gap is $t_{\text{gap}} = 1$ s, so a vehicle maintains a gap with the vehicle in front such that they would not collide even with a one second delay in reaction to the vehicle ahead. The speed limit is $v_f = 20$ kph, yellow time is $y = 5$ s for each phase. The minimum time in green was initially 10 s, however during simulation we found this to be inefficient, so an adaptive minimum green time was used, where $g_p^I = \min\{10, 3.5 + 1.5 \max_{\ell \in L} Q_\ell^I\}$, where L is the set of lanes which are allowed through intersection I in phase p, Q_ℓ^I is the length of the queue in lane ℓ, and the coefficients were chosen to give enough time for each vehicle to pass through. Travel time across intersections $T_p = 17.5$ s. The scaling factor in Eq. (6) is $\phi = .05$, the switch threshold is $\eta = 2$, and the wait threshold is $\gamma = 0.1$ as defined in Section 2. Vehicles spawn at each time step according to a Poisson distribution with parameter λ, and vehicles are distributed into available spawn points with uniform probability unless stated otherwise. The spawn rate was chosen by selecting the highest value in which queues remained bounded for all time, this value was around $\lambda \approx 0.8$ veh/sec, see Fig. 3.

In the single intersection case, our intersection algorithm significantly outperforms a simple intersection with constant phase cycles, where an entire cycle takes 1 min. We use three performance metrics to compare algorithms, the total travel time of all vehicles tt, the total wait time of all vehicles twt, and the total

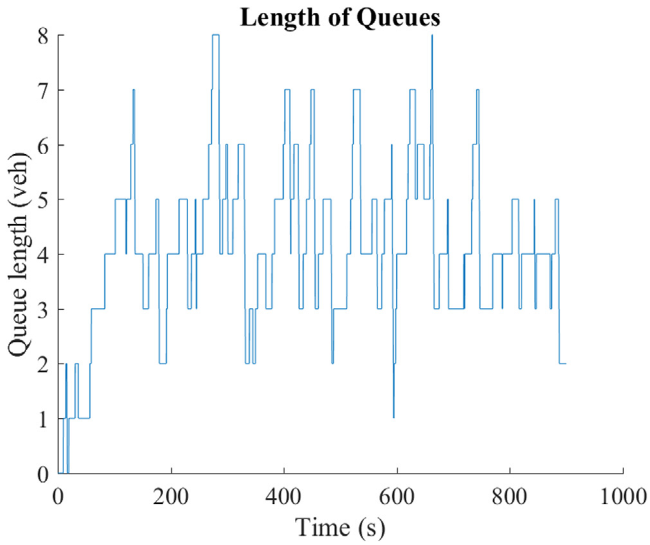

Fig. 3. Max queue length is plotted over time.

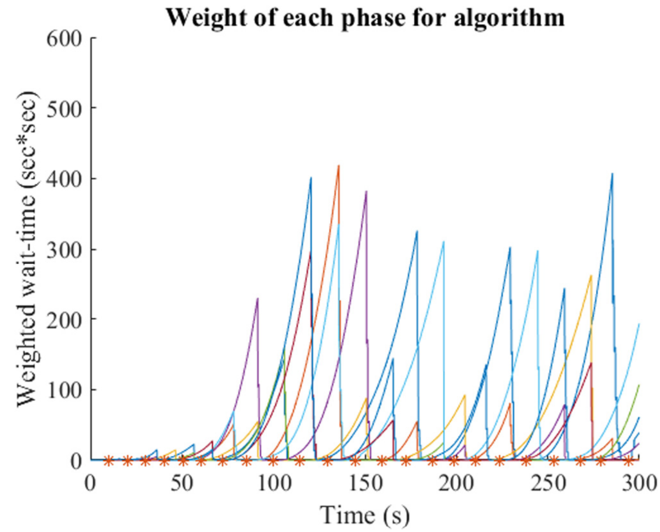

Fig. 4. The evolution of the weight of each phase under the algorithm, where each color represents a different phase.

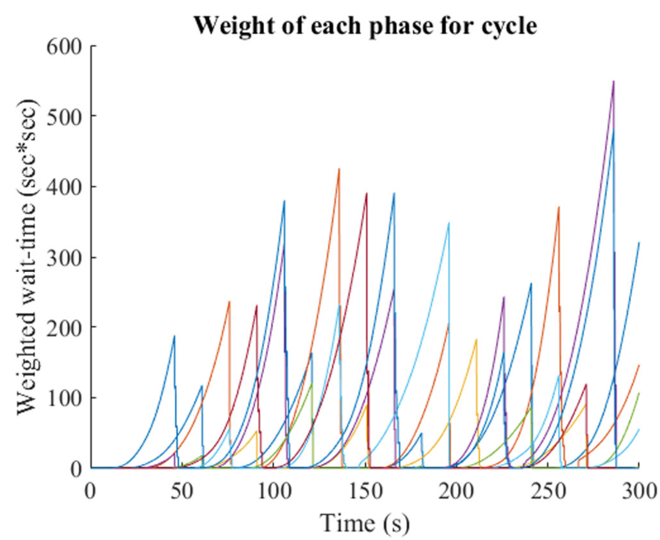

Fig. 5. The evolution of the weight of each phase under a cyclical policy, where each color represents a different phase.

weighted wait time of all vehicles $twwt$. It can be seen in Figs. 4 and 5 that the wait time is kept lower using our algorithm.

In the two intersection case, we compared our algorithm with both the constant cyclic policy and with TAPIOCA (Faye, Chaudet, & Demeure, 2012). TAPIOCA is an algorithm optimizing a cost which tries to balance short-term delay reductions with coordination to reduce long-term delays. It considers linear functions of vehicle density to do so, and makes a few simplifying assumptions about the intersections. TAPIOCA was run with default cost function weights. Compared with the cyclic policy, TAPIOCA performed worse for each metric, and scaled worse with vehicle spawn rate. This could likely have be improved using adaptive weights, but this was not implemented in the paper. In our opinion, without adaptive weights, the cost functions of Faye et al. (2012) does not consider existing large queues with high enough priority. Our policy outperforms the other two policies in all categories, with significant improvements in $twwt$ in congested conditions. We chose coordination coefficient $\alpha = 1$. We also tested varying the spawn distribution or vehicles, essentially creating a main road crossing two minor roads. By doubling the allocation of vehicle spawns to the main road, total travel time increases by 5%, total wait time decreases by 5%, and total weighted wait time decreases by 7%. Overall, the algorithm performs well under both scenarios. Results of the simulations are found in Table 2, units are thousands of seconds.

We also compared each performance given different values of α using Algorithm 1, where α is the coefficient on the cooperation term B_ℓ, see Fig. 6. This simulation used a non-uniform Poisson spawn, where vehicles are twice as likely to spawn on the road which passes through both intersections. These simulations were run for two minutes each. We can see a trade-off where increasing α reduces tt but increases twt and $twwt$, which is expected because B_ℓ is perturbing an algorithm which is minimizing $twwt$ to allow more vehicles through immediately. In Fig. 7, five minute simulations were run to better characterize performance as a function of α when α is small.

Table 2
Performance of the custom algorithm vs cyclical algorithm on one intersection over 5 min, in thousands of seconds.

Policy	λ	tt	twt	$twwt$
Cycle	0.60	7.34	3.34	5.03
	0.80	10.7	4.80	7.50
	1.00	15.4	7.76	14.9
TAPIOCA	0.60	8.03	3.75	7.71
	0.80	12.1	6.41	1.83
	1.00	18.7	11.6	39.3
Custom	0.60	7.18	2.81	3.37
	0.80	10.1	3.94	4.71
	1.00	13.7	5.83	8.17

5. Conclusion and future work

In summary, we presented a novel intersection control algorithm based on an objective function which attempts to accurately characterize driver preferences. A realistic vehicle model is constructed and a simplified version is used to guarantee bounded weighted queues given small spawn rates. The algorithm is tested thoroughly in simulation and compared with other methods.

We are considering several possible extensions to this work. One potential problem with this formulation is that vehicles must be delayed before they are considered important, perhaps a more

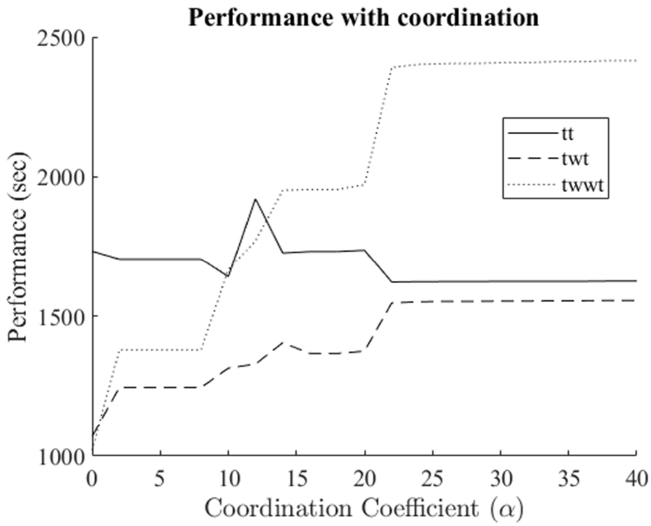

Fig. 6. Two minute two intersection simulation with performance plotted as a function of α.

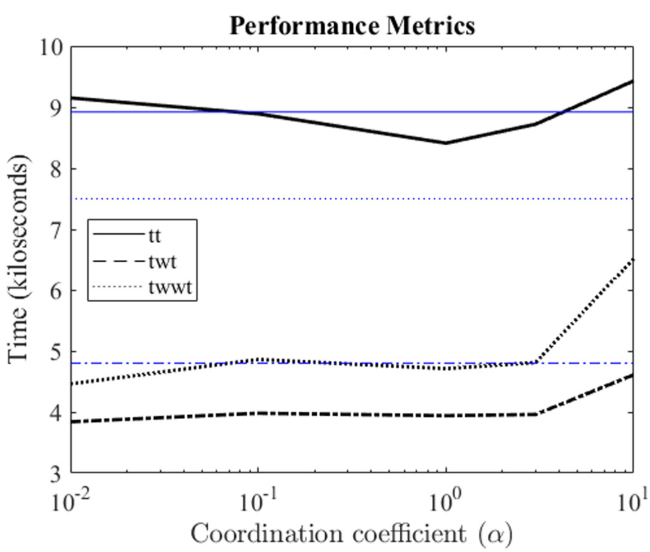

Fig. 7. Five minute simulation of two intersections with performance plotted as a function of α. The thin lines represent the performance under the default cyclic policy under the same conditions. Notice the large decrease in twwt but small increase in tt by using our algorithm.

sophisticated model not based on a threshold value will perform better. We plan to compare performance to more established algorithms such as SCOOT or SCATS. We are interested in addressing the spillover problem, when the queue from one intersection spills into another, we believe the current formulation would allow this to be addressed through an additional term in the objective function. We are also considering a learning-based method for tuning parameters such as wait threshold γ, switch threshold η, time spent in each phase g_u^I, and coordination coefficient α.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Evan Gravelle: Conceptualization, Software, Writing - original draft. **Sonia Martínez:** Formal analysis, Writing - review & editing, Supervision.

References

De Nunzio, G., Gomes, G., de Wit, C. C., Horowitz, R., & Moulin, P. (2016). Speed advisory and signal offsets control for arterial bandwidth maximization and energy consumption reduction. *IEEE Transactions on Control Systems Technology*, *25*(3), 875–887.

Diakaki, C., Papageorgiou, M., & Aboudolas, K. (2002). A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, *10*(2), 183–195.

Dujardin, Y., Vanderpooten, D., & Boillot, F. (2015). A multi-objective interactive system for adaptive traffic control. *European Journal of Operational Research*, *244*(2), 601–610.

Faye, S., Chaudet, C., & Demeure, I. (2012). A distributed algorithm for multiple intersections adaptive traffic lights control using a wireless sensor networks. In *Proceedings of the first workshop on urban networking* (pp. 13–18). ACM.

Frederick, S., Loewestein, G., & O'Donoghue, T. (2002). Time discounting and time preference: A critical review. *Journal of Economic Literature*, *40*, 351–401.

Gravelle, E., & Martínez, S. (2017). A cycle-free coordinated traffic intersection policy using time-inconsistent wait-time functions. In *IEEE conference on control technology and applications*, Kohala Coast, Hawaii, USA (pp. 363–368).

Green, L., Fry, A. F., & Myerson, J. (1994). Discounting of delayed rewards: A life span comparison. *Psychological Science*, *5*(1), 33–36.

Hassin, R. (2016). *Rational queuing*. CRC Press.

Hunt, P., Robertson, D., Bretherton, R., & Winton, R. (1981). *SCOOT - a traffic responsive method of coordinating signals*: *Technical report*.

Keong, C. K. (1993). The glide system - singapore's urban traffic control system. *Transport Review*, *13*(4), 295–305.

Kirby, K. N. (1997). Bidding on the future: Evidence against normative discounting of delayed rewards. *Journal of Experimental Psychology*, *126*(1), 54–70.

Laibson, D. (1997). Golden eggs and hyperbolic discounting. *Quarterly Journal of Economics*, *112*, 443–477.

Laibson, D. (1998). Life-cycle consumption and hyperbolic discount functions. *European Economic Review*, *42*(3), 861–871.

Lowrie, P. (1982). The sydney coordinated adaptive traffic system-principles, methodology, algorithms. In *Int. conference on road traffic signalling, number 207*.

Noland, R., & Small, K. (1995). *Travel-time uncertainty, departure time choice, and the cost of the morning commute*. Irvine: Institute of Transportation Studies, University of California.

Payne, H., & Thompson, S. (1997). Malfunction detection and data repair for induction-loop sensors using i-880 data base. *Transportation Research Record: Journal of the Transportation Research Board*, (1570), 191–201.

Plambeck, E. L., & Wang, Q. (2013). Implications of hyperbolic discounting for optimal pricing and scheduling of unpleasant services that generate future benefits. *Management Science*, *59*(8), iv–1946.

Rajagopal, R., & Varaiya, P. (2007). *Evaluating the health of california's loop sensor network* (p. 94720). Berkeley: University of California.

Sheffer, C. E., Mackillop, J., Fernandez, A., Christensen, D., Bickel, W. K., Johnson, M. W., et al. (2016). Initial examination of priming tasks to decrease delay discounting. *Behavioral Processes*, *128*, 144–152.

Shen, Z., Wang, K., & Zhu, F. (2011). Agent-based traffic simulation and traffic signal timing optimization with gpu. In *IEEE Int. Conf. on intelligent transportation systems* (pp. 145–150). IEEE.

Tian, Z., & Urbanik, T. (2007). System partition technique to improve signal coordination and traffic progression. *Journal of Transportation Engineering*, *133*(2), 119–128.

Wongpiromsarn, T., Uthaicharoenpong, T., Wang, Y., Frazzoli, E., & Wang, D. (2012). Distributed traffic signal control for maximum network throughput. In *IEEE Int. Conf. on intelligent transportation systems* (pp. 588–595). IEEE.