Coordinated Look-Ahead Scheduling for Real-Time Traffic Signal Control

(Extended Abstract)

Xiao-Feng Xie, Stephen F. Smith, Gregory J. Barlow
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue,Pittsburgh, PA 15213
{xfxie,sfs,gjb}@cs.cmu.edu

ABSTRACT

We take an agent-based approach to real-time traffic signal control based on coordinated look-ahead scheduling. At each decision point, each agent constructs a schedule that optimizes movement of the currently approaching traffic through its intersection. For strengthening its local view, each agent queries the scheduled outflows from its direct upstream neighbors to obtain an optimistic observation, which is capable of incorporating non-local impacts from indirect neighbors. We summarize results on a road network of tightly-coupled intersections that demonstrate the ability of our approach.[1]

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems, Coherence and coordination*

General Terms

Algorithms

Keywords

Distributed Scheduling, Multi-Agent Coordination, Intelligent Transportation Systems, Real-Time Systems

1. INTRODUCTION

Intelligent traffic signal control presents the potential to substantially reduce congestion in road networks. However, how to achieve effective real-time control remains challenging [2]. Not only are the number of joint signal control sequences and local observations huge for just one intersection, but efficient flow of traffic through a road network also requires coordination among neighboring intersections.

Given the complexity and inherently distributed nature of real-time traffic signal control, we take an agent-based approach to solving this problem. We assume that each intersection is controlled by an agent using a schedule-driven intersection control strategy (SchIC) [4]. To strengthen the local views of individual agents and avoid myopic decisions, each agent asynchronously requests a projection of output flows from its direct upstream neighbors at each decision point to obtain an optimistic observation, which is capable of incorporating non-local impacts from indirect neighbors.

2. PROBLEM DEFINITION

To keep the following description of our coordinated look-ahead scheduling as simple as possible, we focus on an one-way road network of signalized intersections. At each intersection, the traffic light cycles through a fixed sequence of phases I, where each phase $i \in I$ governs the right of way for a set of non-conflicting movements from entry to exit roads.

Each intersection is controlled by an agent that proceeds according to a rolling horizon [2–4], by holding a finite signal sequence SS_{TL}, and continually appending it with a short sequence (SS_{ext}) at each successive decision point. Each *signal sequence* contains a sequence of green phases and associated durations. Furthermore, SS_{TL} always satisfies the timing constraints for fairness and safety: each phase i has a variable duration (g_i) that can range between a minimum (G_i^{min}) and maximum (G_i^{max}), while the yellow light after each phase i runs for a fixed duration (Y_i).

For traffic signal control, the objective is to minimize the average delay of vehicles traveling through the road network.

3. INTERSECTION CONTROL

We adopt a schedule-driven intersection control (SchIC) strategy [4]. The basic idea is to form a *scheduling problem* using the current *observation* (o), particularly the *inflows* (IF) in the *prediction horizon* (H), and to generate a *schedule* that obtains a near optimal *control flow* (CF^*).

To achieve efficiency, we exploit an aggregate flow representation. Vehicles in a given non-uniform flow are organized using an ordered *cluster sequence* $C = (c_1, \cdots, c_{|C|})$, where $|C|$ is the number of clusters in C. Each *cluster* c is defined as $(|c|, arr, dep)$, where $|c|$ is the number of vehicles in c, and arr (dep) gives the expected arrival (departure) time at the intersection respectively for the first (last) vehicle in c.

An observation o contains the current decision time cdt, the current phase index cpi and duration cpd of SS_{TL}, and the inflows IF containing the currently sensed vehicles.

Formally, $IF = (C_{IF,1}, \cdots, C_{IF,|I|})$, where $C_{IF,i}$ is a cluster sequence containing the vehicles with the right of way during phase i. Clusters in each $C_{IF,i}$ are further aggregated into an anticipated queue and arriving clusters.

A control flow CF contains the results of applying a signal sequence that clears all clusters in an observation o. Formally, $CF = (S, C_{CF})$, where S is a sequence of phase indices, i.e., $(s_1, \cdots, s_{|S|})$, and C_{CF} contains a sequence of clusters $(c_{CF,1}, \cdots, c_{CF,|S|})$ that are reorganized from IF.

[1]This research was supported in part by the Traffic21 Initiative at Carnegie Mellon University, with support from the Hillman Foundation and the Heinz Endowments.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Algorithm 1 Obtain an optimistic non-local observation

1: $m = GetEntryRoadByPhase(i)$ {For each phase i}
2: $UpAgent = GetUpstreamAgent(m)$
3: Request C_{OF} from $UpAgent$ using (cdt, m, H_{ext})
4: $Shift(C_{OF}$, the travel time on $m)$
5: Append C_{OF} into $C_{IF,i}$

Algorithm 2 Return C_{OF} for a message (cdt, n, H_{ext})

1: $(C_{OF}, S_{OF}) = (C^*_{CF}, S^*) \cap [cdt, cdt + H_{ext}]$
2: **for** $k = |C_{OF}|$ to 1 **do**
3: $\quad |c_{OF,k}| = |c_{OF,k}| \cdot tp(s_{OF,k}, n)$ {turning proportion}
4: **end for**

For any k, all vehicles in $c_{CF,k}$ belong to C_{IF,s_k}.

The *scheduling search space* is formed by viewing each cluster as a non-divisible job. The jobs in $C_{IF,i}$ can only leave the intersection when the phase index is i, and the jth job can only leave after the $(j-1)$th one has left. Each S is a *schedule* with $|S| = \sum_{i=1}^{|I|} |C_{IF,i}|$. For a partial schedule S_k (the first k elements of S), its *schedule status* is defined as $X{=}(x_1, \cdots, x_{|I|})$, where $x_i \in [0, |C_{IF,i}|]$. In the state update that adds s_k to S_{k-1}, we have $x_{s_k} = x_{s_k} + 1$, $c_{CF,k}$ comes from the x_{s_k}th cluster in C_{IF,s_k}, and the actual arrival time and cumulative delay of $c_{CF,k}$ are determined according to a greedy construction of the corresponding signal sequence [4].

The cumulative delay of CF^* is minimized by a dynamic programming process [4], which has $|I|^2 \cdot \prod_{i=1}^{|I|}(|C_{IF,i}| + 1)$ state updates in the worse case, where $|C_{IF,i}| \leq H$, and each state update can be executed in constant time. It is polynomial in H since $|I|$ is limited in the real world.

The first job in CF^*, if available, is used to determine SS_{ext}. There are two possible extension choices: 1) terminate the current green phase and move to the next (if $|S^*| \equiv 0$, or $s^*_1 \neq cpi$, or $arr(c^*_{CF,1}) \geq SwitchBack(cpi)$); or otherwise 2) extend the current phase, in which case $ext = \min(dep(c^*_{CF,1}) - cdt, th_{ext})$, where th_{ext} is the upper limit. A *repair rule* is applied lastly to ensure that SS_{TL} does not violate any time constraints after appending SS_{ext}.

4. BASIC COORDINATION MECHANISM

In a road network, an agent is susceptible to myopic decisions if its local prediction horizon is not sufficiently long. To counteract this possibility we extend each agent's local view with an optimistic non-local observation from its upstream agents, as shown in Algorithm 1. For each phase index i, the corresponding entry road m is identified, and the corresponding upstream agent $UpAgent$ is obtained. The agent then sends $UpAgent$ a request message (cdt, m, H_{ext}), where H_{ext} is the maximum *horizon extension*, for the planned output flow C_{OF} of $UpAgent$. Upon receipt of C_{OF}, the downstream agent adds an offset time — the average travel time between the two agents (intersections) — to all the clusters in C_{OF} and appends the clusters to the end of $C_{IF,i}$.

$UpAgent$ executes Algorithm 2 to obtain the output flow C_{OF} at the current time cdt, based the previously planned control flow (S^*, C^*_{CF}). The entry road m of the requesting agent is the exit road n of $UpAgent$. In Line 1, (C_{OF}, S_{OF}) is obtained as the subsequence of (C^*_{CF}, S^*) that belongs to the time period $[cdt, cdt + H_{ext}]$. In Line 3, $tp(i, n)$ is the portion of traffic turning onto exit road n during phase i.

An essential property of this protocol is that non-local impacts from indirect neighbors can be included if H_{ext} is sufficiently long, since the control flow of direct neighbors contains flow information from their upstream neighbors.

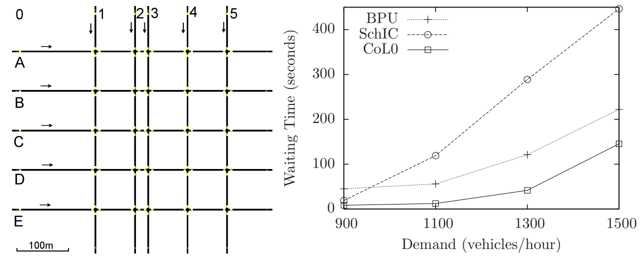

Figure 1: (a) 5X5 grid network; (b) Average Results.

The optimistic assumption that is made is that direct and indirect neighbors are trying to follow their schedules. The optimization capability of SchIC makes schedules quite stable. Minor schedule changes in neighbors can be absorbed by exploiting the temporal flexibility in their control flows.

5. RESULTS

We simulate performance using SUMO[2] on a 5X5 one-way grid network as shown in Figure 1 (a). In this network, all road lengths are 75 meters, except for the horizontal roads **2** $\rightarrow$ **3** and **0** $\rightarrow$ **1**, which are respectively 25 and 150 meters.

On each road, the free-flow speed is 10 meters per second. For each intersection, Y, G^{min} and G^{max} are respectively 5, 5, and 55 seconds. Because the minimal switchback time ($Y + G^{min} + Y = 15$ seconds) is longer than the travel time on one road (2.5 or 7.5 seconds), non-local impacts from indirect neighbors might be nontrivial and cannot be ignored.

Only through traffic movements are considered. For background traffic, each minor route generates a flow of 1/20 of the total traffic. There are two major flows on **C** and **3** that generate 3/5 of the total traffic. The total simulation time is one hour, and for each twenty minute period, the demand ratios between **C** and **3** are 35:25, 40:20, and 45:15.

Figure 1 (b) shows the average results of three control strategies, i.e., BPU, SchIC, and CoL0, for different demands. BPU (balanced phase utilization) [1] is an adaptive coordination strategy using offset calculation, SchIC is the isolated control strategy [4], and CoL0 applies the optimistic non-local observation ($H_{ext} = 15$ seconds) to SchIC.

CoL0 produced lower waiting times than both other strategies. Comparison to SchIC demonstrates the added benefit of optimistic non-local observation. Furthermore, CoL0 outperforms BPU without requiring explicit offset calculation; coordination between neighbors is instead accomplished implicitly by looking ahead to upstream output flows. Future work will explore the use of additional coordination mechanisms to address specific situations (e.g., queue spillbacks).

6. REFERENCES

[1] G. J. Barlow. *Improving Memory for Optimization and Learning in Dynamic Environments*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2011.

[2] M. Papageorgiou, C. Diakaki, V. Dinopoulou, and et al. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

[3] S. Sen and K. Head. Controlled optimization of phases at an intersection. *Transport. Sci.*, 31(1):5–17, 1997.

[4] X.-F. Xie, S. F. Smith, L. Lu, and G. J. Barlow. Schedule-driven intersection control. Technical Report CMU-RI-TR-11-34, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2011.

[2] Simulation of urban mobility: http://sumo.sourceforge.net