

# Bi-Directional Information Exchange in Decentralized Schedule-Driven Traffic Control

Extended Abstract

Hsu-Chieh Hu, Stephen F. Smith  
Carnegie Mellon University, Pittsburgh, PA, USA  
hsuchieh@andrew.cmu.edu, sfs@cs.cmu.edu

## ABSTRACT

Recent work in decentralized, schedule-driven traffic control has demonstrated the ability to improve the efficiency of traffic flow in complex urban road networks. In this approach, each time an agent generates a new intersection schedule it communicates its expected outflows to its downstream neighbors as a prediction of future demand and these outflows are appended to the downstream agent’s locally perceived demand. In this paper, we extend this basic coordination protocol to additionally incorporate the complementary flow of information reflective of an intersection’s current congestion level to its upstream neighbors. We present an asynchronous decentralized algorithm for updating intersection schedules and congestion level estimates based on these bi-directional information flows. By relating this algorithm to the self-optimized decision making of the basic protocol, we are able to approach network-wide optimality and reduce inefficiency due to myopic intersection control decisions.

## KEYWORDS

Distributed artificial intelligence, Coherence and coordination, Multi-agent planning and scheduling

### ACM Reference Format:

Hsu-Chieh Hu, Stephen F. Smith. 2018. Bi-Directional Information Exchange in Decentralized Schedule-Driven Traffic Control. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)*, Stockholm, Sweden, July 10–15, 2018, IFAAMAS, 3 pages.

## 1 INTRODUCTION

A recent approach to traffic signal control has achieved significant traffic flow efficiency improvements through real-time, distributed generation of long-horizon, signal timing plans. [2, 4, 5] The key idea behind this approach is to formulate the intersection scheduling problem as a single machine scheduling problem, where input jobs are represented by as sequences of spatially proximate vehicle clusters (approaching platoons, queues). One potential limitation of this approach, however, is its reliance on one-way flow of demand information from upstream intersections to downstream intersections. In this abstract, we consider the possibility of improving network-level performance by augmenting the information exchanged between neighboring intersections to include complementary upstream flow of congestion information.

We propose an expanded bi-directional information exchange protocol between intersections that combines forward communication of projected vehicle outflows to downstream intersections with backward communication of the estimated delay for each vehicle to upstream intersections as a prediction of next-hop costs. This additional information is incorporated by redefining the local intersection scheduling objective to include these costs. In situations where traffic is light, the feedback delay will be small and local intersection scheduling will proceed as before. However as the network becomes saturated and the cumulative delay of downstream neighbors becomes larger, the feedback cost will reflect this and lessen the number of vehicles that are sent downstream in this direction. To ensure scalability, messages continue to be exchanged only between direct neighbors and the asynchronous nature of local intersection scheduling is preserved.

## 2 PROBLEM DEFINITION

As in [4], we assume that streams of stopped and approaching vehicles detected along each intersection approach (or phase) are aggregated into sequences of vehicle clusters  $c = (|c|, arr, dep)$ , and these input flows are integrated into a single sequence (called a phase schedule) that minimizes cumulative weight time through a forward recursion dynamic programming (DP) search. Formally, the resulting control flow can be represented as a tuple  $(S, C)$ , where  $S$  is a sequence of phase indices, i.e.,  $(s_1, \dots, s_{|S|})$ ,  $C$  contains the corresponding sequence of clusters  $(c_1, \dots, c_{|S|})$ . The delay that each cluster contributes to the cumulative delay  $\sum_{k=1}^{|S|} d(c_k)$  is defined as  $d(c_k) = |c_k| \cdot (ast - arr(c_k))$ , where  $ast$  is the actual (scheduled) start time derived by the DP search. Once  $(S, C)$  is generated, the first step of the plan is executed and projected outflows are communicated to each downstream neighbor and appended to each’s detected input flows. Intersections re-optimize their local schedules asynchronously at each time step  $t$ .

Our hypothesis is that the effectiveness of this schedule-driven process is restricted by the fact that each intersection’s scheduling agent optimizes its local cumulative delay without regard to the cost it imposes on downstream intersections. To formulate the network level problem, we model a transportation network by a graph  $G = \{V, E\}$ , where the vertex  $v \in V$  is the intersection and  $e \in E$  is the road segment connecting the intersections. Since schedule-driven traffic control is an online planning approach, overall performance can be formulated as the sum of the following coupled objective:

$$\min_{\{C_i(t), i \in V\}} \sum_{i \in V} f_i(C_i(t), C_{-i}(t)), \quad (1)$$

where  $f_i(C_i(t), C_{-i}(t)) = \sum_{k=1}^{|S|} d(c_k)$  is the cumulative delay of intersection  $i \in V$  given the schedules  $C_{-i}(t) = (C_1(t), \dots, C_{i-1}(t), C_{i+1}(t), \dots, C_{|V|}(t))$  of all intersections except  $i$ .

### 3 BI-DIRECTIONAL INFORMATION EXCHANGE

Our approach to incorporating next-hop delay through reciprocal exchange of downstream congestion cost information in addition to exchanged schedule outflow information approximates the network level optimization problem by modifying the local intersection scheduling objective. In addition, we assume that intersections only communicate with a set of their direct neighbors  $\mathcal{N}_i$ . To illustrate, we rewrite the overall approximated performance of intersection  $i$  as:

$$\min_{C_i(t)} f_i(C_i(t), C_{\mathcal{N}_i}(t-1)) + \sum_{j \in \mathcal{N}_i} f_j(C_j(t-1), \{C_{\mathcal{N}_j \setminus i}(t-1), C_i(t)\}). \quad (2)$$

The second term of (2) considers the number of vehicles that will be sent to neighbors according to  $C_i(t)$ . In other words, the possible delay of vehicles being sent forward from intersection  $i$  incurred at intersections other than  $i$  is taken into account when computing  $i$ 's schedule  $C_i(t)$ . We introduce a *congestion feedback* metric, denoted by  $\hat{d}[C_{p,j}(t-1)]$  for phase  $p$ , to quantify this contribution. Through the cluster representation of schedule-driven traffic control, we have an intuitive way to estimate  $\hat{d}[C_{p,j}(t-1)]$

*Definition 3.1 (Congestion Feedback).* Intersection  $j$  computes its average delay of each phase  $p$  for communication to its corresponding upstream neighbors as follows:

$$\hat{d}[C_{p,j}(t-1)] = \frac{\sum_{c_{p,k} \in C_{p,j}(t-1)} d(c_{p,k})}{\sum_{c_{p,k} \in C_{p,j}(t-1)} |c_{p,k}|}. \quad (3)$$

Specifically, (3) is the estimated next-hop delay of  $c_{p,k}$  at intersection  $j$  according to control flow  $C_{p,j}(t-1)$  at the previous time step. Using the notion, we can define a new version of delay for each cluster at the intersection  $i$  that incorporates the cost it imposes on others:

*Definition 3.2 (Augmented Delay).* Assume the next hop of  $c_{p,k}$  is intersection  $j$ . Then, the delay associated with two hops is represented as

$$d(c_{p,k}) = |c_{p,k}| \cdot \left[ (ast - arr(c_{p,k})) + \hat{d}[C_{p,j}(t-1)] \right], \quad (4)$$

where  $c_{p,k} \in C_{p,i}(t)$ .

### 4 DISTRIBUTED CONGESTION COMPENSATION (DCC) PROTOCOL

To incorporate congestion feedback information, each local intersection  $i$  optimizes the following revised objective:

$$C_i(t) = \arg \max_{\hat{C}_i = (\hat{C}_{1,i}, \dots, \hat{C}_{|P|,i})} \sum_{p=1}^{|P|} \sum_{c_{p,k} \in \hat{C}_{p,i}} d(c_{p,k}), \quad (5)$$

where the  $d(c_{p,k})$  is the augmented delay defined in (4).

The DCC protocol used to compute network-level schedules is given below.

**The DCC Protocol** Steps defining how intersection  $s$  communicates to its upstream neighbors to achieve "social welfare" of the network

- 1: **Initialization:** For intersection  $i \in V$  generate a initial schedule  $C_i(0)$  and set the congestion feedback to 0.
- 2: **Receive congestion feedback and outflow information:** At each time  $t$ , intersection  $i$  receives congestion feedback from downstream of  $j \in \mathcal{N}_i$ , which is  $\hat{d}[C_{p,j}(t-1)]$ , and schedule (outflow information) from upstream of  $j \in \mathcal{N}_i$ .
- 3: **Forward-recursion dynamic programming search:** Intersection  $i$  computes its schedule  $C_i(t)$  according to equation (2).
- 4: **Feedback congestion feedback and outflow information:** According to equation (3), intersection  $i$  calculates  $\hat{d}[C_{p,i}(t)]$  and schedule and shares them with upstream and downstream neighbors.

### 5 PERFORMANCE EVALUATION

To assess performance, a real world network with 2-way, multiple lane, and multi-directional traffic flow is considered. The simulation model was developed in VISSIM. The network model is based on the Baum-Centre neighborhood of Pittsburgh, Pennsylvania as shown in Figure 1. The network consists mainly of 2-phased intersections. It can be seen as a two-way grid network. All simulation runs were carried out according to a realistic traffic pattern from late afternoon through "PM rush" (4-6 PM). To serve as a second practical benchmark, we implemented a version of the SCATS system according to [1, 3].

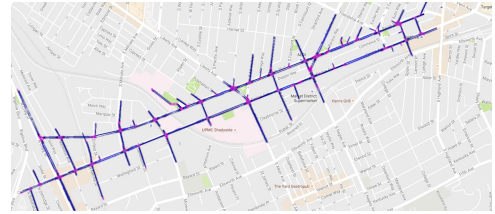


Figure 1: Map of the 24 intersections in the Baum-Centre neighborhood of Pittsburgh, Pennsylvania

Table 1: Summary of Baum Centre Model Results

	Delay Mean (s)	Delay SD.	Avg no. of stops
Benchmark	147.00	177.94	8.27
SCATS	169.23	265.91	10.81
DCC	121.56	100.65	5.33
DCC w/ BC	116.01	93.22	5.32

Table 1 shows the results of DCC under PM rush, compared to SCATS approach and the baseline schedule-driven approach. In addition to DCC, we also compare DCC with DCC-BC, an extension with the additional bottleneck condition that the original objective be used for intersection  $i$  if  $\hat{d}[C_{p,i}(t)] \geq \hat{d}[C_{p,j}(t-1)]$ , for each downstream neighbor  $j$ . As can be seen, delay is reduced by 17.7% and 28.4%, compared to the schedule-driven and SCATS approaches respectively.

## REFERENCES

- [1] LIU DAI ZONG. 2003. *Comparative evaluation of dynamic TRANSYT and SCATS-based signal control systems using Paramics simulation*. Ph.D. Dissertation.
- [2] Stephen F Smith, Gregory J Barlow, Xiao-Feng Xie, and Zachary B Rubinstein. 2013. Smart Urban Signal Networks: Initial Application of the SURTRAC Adaptive Traffic Signal Control System.. In *ICAPS*. Citeseer.
- [3] Tichakorn Wongpiromsarn, Tawit Uthaicharoenpong, Yu Wang, Emilio Frazzoli, and Danwei Wang. 2012. Distributed traffic signal control for maximum network throughput. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 588–595.
- [4] Xiao-Feng Xie, Stephen F Smith, and Gregory J Barlow. 2012. Schedule-Driven Coordination for Real-Time Traffic Network Control.. In *ICAPS*.
- [5] Xiao-Feng Xie, Stephen F Smith, Liang Lu, and Gregory J Barlow. 2012. Schedule-driven intersection control. *Transportation Research Part C: Emerging Technologies* 24 (2012), 168–189.