

# Expressive Real-Time Intersection Scheduling

**Rick Goldstein**

The Robotics Institute  
Carnegie Mellon University  
rgoldste@cs.cmu.edu

## Introduction

Traffic congestion is a widespread annoyance throughout global metropolitan areas. It causes increases in travel time, increases in emissions, inefficient usage of gasoline, and driver frustration. Inefficient signal patterns at traffic lights are one major cause of such congestion. Intersection scheduling strategies that make real-time decisions to extend or end a green signal based on real-time traffic data offer one opportunity reduce congestion and its negative impacts.

My research proposes Expressive Real-time Intersection Scheduling (ERIS). ERIS is a decentralized, schedule-driven control method which makes a decision every second based on current traffic conditions to reduce congestion.

## Problem Overview

In the intersection control problem, we consider an intersection (or network of intersections), such as the one presented in Figure 1 where traffic lights have been installed to improve mobility or increase safety. The intersection has inbound lanes from various directions. The intersection may associate a subset of lanes, referred to as movements, with a green signal to indicate to vehicles in the respective lanes that it is safe to traverse the intersection. Movements must satisfy pre-specified minimum and maximum timing limits and we require yellow and all-red clearances with fixed lengths to occur between conflicting green movements for safety. In the intersection control problem, we are interested in implementing a legal control strategy that will minimize a measure of dis-utility such as average vehicle delay.

## Related Work

High-level scheduling strategies were first applied to intersection scheduling. These strategies set timing parameters, typically *cycle length* for an entire network of intersections, *splits* between green time for each intersection within a network, and *offsets* which specify how to stagger the start times of green phases across a network. TRANSYT computed these parameters offline, while more recent methods such as SCATS and SCOOT update such parameters on-line (every 10 minutes) (Robertson 1969; Lowrie 1990; Robertson and Bretherton 1991). These methods optimize

for average expected flows, yet realized flows across a few seconds are often quite variable from the averages. When realized flows on any side are much lower than expected, these methods provide unnecessary green time that could be better allocated to busier sides.

Real-time intersection scheduling methods improve upon high-level strategies by making decisions on real-time data. Real-time methods are not constrained by fixed timing requirements and can immediately adapt when one side of the intersection is unexpectedly much busier. The majority of these methods (including ERIS) are schedule-driven methods. They use real-time vehicle information to generate a schedule of phases to minimize a cost function. The first step of this schedule is then sent to the traffic controller and executed. This process repeats every time-step.

Our work most closely resembles SURTRAC, which is a decentralized schedule-driven method which runs a forward dynamic program to calculate an optimal schedule for known vehicle clusters in real-time by treating each intersection as a single-machine scheduling problem (Xie et al. 2012). SURTRAC makes several modelling simplifications to ensure that the forward dynamic program can run in real time, such as simpler traffic light patterns and combining compatible flows. ERIS does not make these simplifications and thus maintains a more accurate representation of the underlying traffic model.

## Expressive Real-time Intersection Scheduling

Expressive Real-time Intersection Scheduling (ERIS) is my overall framework that performs all necessary steps to make a scheduling decision for the intersection during every time-step. ERIS' Executor first calculates a prediction of vehicle clusters in each lane using nearby vehicle broadcasts of position, speed, and heading as well as information shared from neighboring intersections about predicted future out-

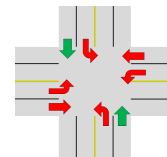


Figure 1: Intersection

flows. Historical turning proportions and a model of expected travel times are learned from past vehicle flows to estimate intersection arrival times and desired lane.

Once clusters are obtained, they are passed to the Scheduling Module to obtain an optimal schedule of phases that allows all vehicle clusters to pass through the intersection while minimizing total delay. ERIS maintains separate clusters for each lane approaching the intersection; this makes the underlying traffic model more accurate, but also makes the search more difficult. We employ an A\* search where search states keep track of vehicles served on each lane, the traffic light state, world time, and incurred delay. With use of an admissible heuristic function that breaks the problem into additive sub-problems, ERIS efficiently calculates optimal schedules.

Once a schedule is obtained, the first step of the schedule is formatted into a command to stay in the same phase or to switch to another phase and sent to the traffic light controller to execute. Similarly, scheduled cluster outflows are broadcast to and received from neighboring intersections for use during the next time-step when building vehicle clusters.

ERIS has been shown to outperform SURTRAC and other real-time strategies by as much as 20% when compared on an open-source traffic simulator (Goldstein and Smith 2018).

### Research Progress & Thesis Contribution

I have two primary goals for my thesis research. First, I hope to develop an advanced connected vehicle scheduling framework that can handle a variety of complicated situations, including varying intersection geometries, multiple vehicle types, uncertainty, and multi-intersection scheduling strategies. Secondly, I intend to implement my scheduling framework on actual intersections throughout the city of Pittsburgh, Pennsylvania.

I have implemented the core modules of ERIS and run experiments on traffic networks in simulation. Our paper explaining these core modules was presented at AAAI-18 (Goldstein and Smith 2018). I have begun to examine varying intersection geometries, multiple vehicle types, and thinking about multi-intersection scheduling strategies.

My work uses a decentralized scheduling framework because fully centralized models, such as the PDDL+ model proposed by McCluskey and Vallati (2017), tend to be too slow to schedule multiple intersections in real-time. I am currently developing a multi-intersection framework that discourages myopic intersection-level decisions that increase network-level delay. This framework is robust to communication failures; intersections enact their no-communication optimum if communication fails. I intend to submit this work to IJCAI-19.

In my work, I assume that all traffic movements are protected movements. In many metropolitan intersections, vehicles desiring to turn left are provided with a green ball, indicating they may turn if there are no oncoming vehicles. By incorporating probabilistic left turns that may occur during a break in oncoming traffic, I will complicate the model, yet make the model applicable to a larger set of intersections. Additionally, I assume that vehicles and intersections

communicate every second, which is a little more frequently than what I observed while interning at Google Maps this past summer. I am considering methods to reduce the required amounts of communication.

After completing these model additions, I intend to spend the final year of my PhD running field studies of ERIS on the 50 traffic lights throughout Pittsburgh that our research lab controls. This will include writing a back-end that allows vehicles to communicate their position, speed, and heading with nearby intersections.

I have written all of the code for ERIS' Modules. I expect field testing to be a joint effort. A proposed timeline is presented below.

### Proposed Timeline

- 11/18-1/19: Finalize multi-intersection scheduling work and prepare IJCAI paper.
- 2/19-4/19: Develop thesis proposal. Work on multi-modal scheduling.
- 5/19-8/19: Continue multi-modal work. Target a paper for AAAI or ICAPS. Develop field testing goals and write necessary communication code.
- 9/19-11/19: Begin field testing.
- 12/19-3/20: Incorporate uncertainty & unprotected left turns into model, adapt search dynamics and heuristic function, write paper. Continue field testing.
- 4/20-8/20: Write thesis explaining core model, multi-intersection scheduling, multi-modal scheduling, uncertainty, and field testing.

### Conclusion

This extended abstract presented the intersection control problem and explained ERIS, my proposed solution to reduce congestion. ERIS outperforms less expressive scheduling models by as much as 20%. I designed the A\* search and other necessary modules and have begun working on developing advanced features. The next two years will be spent incorporating additional features and performing field tests of ERIS.

### References

- Goldstein, R., and Smith, S. F. 2018. Expressive Real-time Intersection Scheduling. In *32rd Conference on Artificial Intelligence (AAAI)*.
- Lowrie, P. 1990. SCATS, Sydney Co-ordinated Adaptive Traffic System: A Traffic Responsive Method of Controlling Urban Traffic. *Roads and Traffic Authority of New South Wales- Traffic Control Section*.
- McCluskey, T. L., and Vallati, M. 2017. Embedding Automated Planning within Urban Traffic Management Operations. In *ICAPS*.
- Robertson, D. I., and Bretherton, R. D. 1991. Optimizing Networks of Traffic Signals in Real Time - The SCOOT Method. *IEEE Transactions on vehicular technology* 40(1):11-15.
- Robertson, D. I. 1969. TRANSYT: A Traffic Network Study Tool. *Road Research Laboratory Report, LR 253*.
- Xie, X.-F.; Smith, S. F.; Lu, L.; and Barlow, G. J. 2012. Schedule-Driven Intersection Control. *Transportation Research Part C: Emerging Technologies* 24:168-189.