Intelligent Intersection Control for Platoons of Autonomous Vehicles

By William Shen*, ANU CECS, u6096655@anu.edu.au. Supervisor: Jochen Renz†.

1 Introduction

In an environment made up entirely of autonomous vehicles, intersections should not rely on traditional control methods including traffic lights and stop signs. These traditional control methods obstruct the flow of traffic, as vehicles must come to a standstill and wait, even if there will be no other vehicles passing through the intersection in the near future.

Instead, intersections should be controlled intelligently by determining the best set of actions when autonomous vehicles approach that intersection to minimize any potential delays and maximize throughput for that intersection.

In platoons, vehicles are grouped together such that each platoon can be considered and behave as a single vehicle itself. Adopting platooning can lead to several potential benefits, including reduced energy consumption through drag reduction, and decreased travel times.

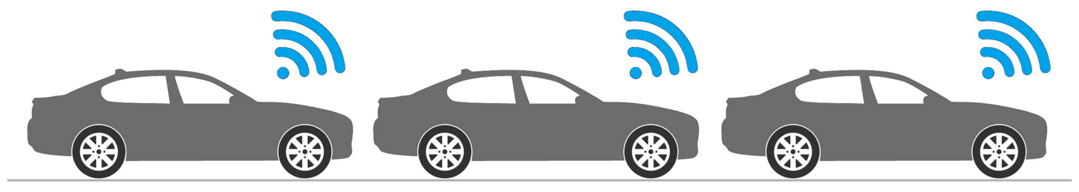


Figure 1. Platooning and V2V Communication

Existing solutions are mainly concerned with 'reservation'-based systems, where each vehicle sends a reservation request to a central coordinator (vehicle-to-infrastructure communication, V2I). We believe a decentralized (vehicle-to-vehicle communication, V2C), calculation-based scheduler could yield far better results as it would consider vehicles as a whole rather than on a need-to-need basis.

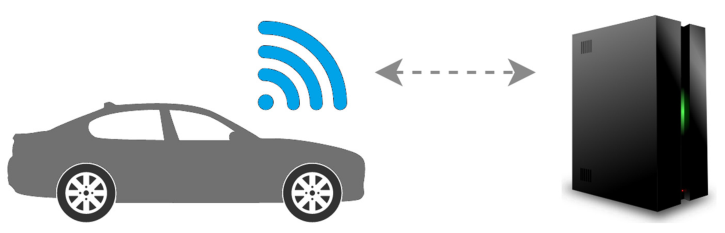


Figure 2. V2I Communication

1.1 Motivation

The Australian Bureau of Infrastructure, Transport and Regional Economics (BITRE) estimated in 2015 that the 'avoidable' social costs of traffic congestion was $16.5 billion AUD. Ultimately, an intelligent intersection controller could lead to a stronger economy through a rise in efficiency.

Studies by the National Cooperative Highway Research Program (NCHRP) in the United States show that around 50% of urban crashes and 30% of rural crashes occur at intersections. Thus, our intersection controller would provide all the economic benefits as aforementioned, as well as improving road safety and overall human health.

2 Problem Description

We define a plan to be the sequence of operations that must be applied to an arbitrary number of vehicles or platoons approaching an intersection.

A plan is 'valid' if no vehicle or platoon overlaps with another during the execution of the plan (i.e. no collisions). We define a plan to be optimal if it is a valid plan, and if it minimizes our objective function (total travel time).

We define a platoon to be a collection of vehicles, and let the term 'vehicular agent' (VA) refer to either platoons or vehicles.

* College of Engineering & Computer Science, Australian National University

† Research School of Computer Science, College of Engineering & Computer Science, Australian National University

2.1 Intersections

Each intersection has a number of lanes, each with a width and direction in which traffic flows. More importantly, an intersection has a central area where the actual crossing of vehicular agents takes place.

2.2 Important Terminology

Each vehicular agent has a t_a (time of arrival) and t_d (time of departure) with respect to its intersection.

Each vehicular agent also has a trajectory which is a tuple (α, β) where α is the lane the VA is travelling in, and β is the turning intent. We define a pair of trajectories to be crossing if they share at least one crossing point in the intersection.

2.3 Collision Detection

To detect collisions on each lane of our road network, we simply calculate whether any vehicular agent will 'catch-up' with another vehicular agent ahead. We call this constraint `roadCollision`.

To detect collisions within each intersection, we divide the area of our intersection into little grids as observed in figure 3. If any of the grid squares is occupied by another VA, then our plan is invalid. We call this constraint `intersectionCollision`.

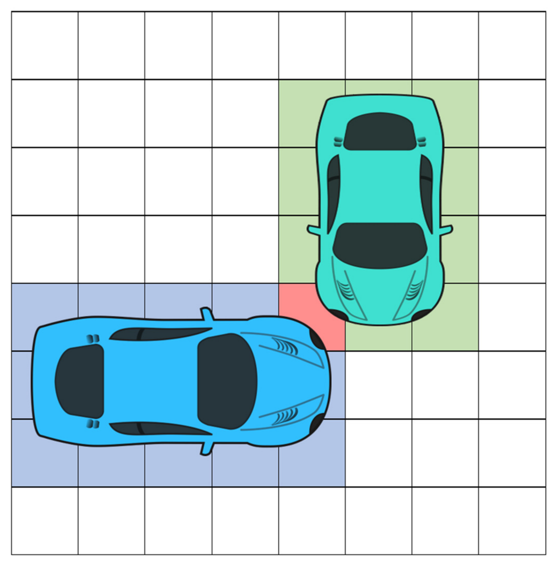


Figure 3. Intersection collision detection

3 Constraint-Based Scheduling

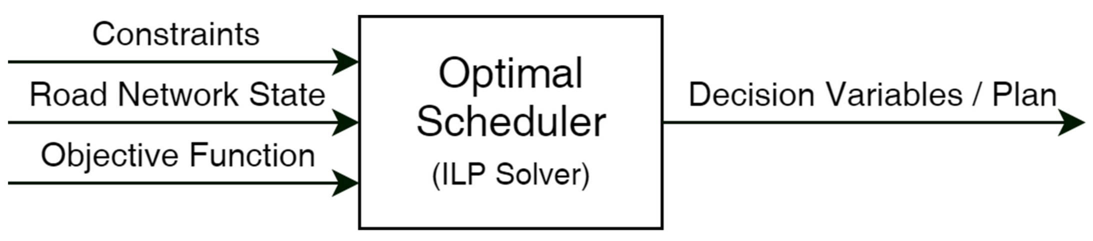
We outline the constraints that must be satisfied for a plan to be valid:

- VAs travelling in the same lane must not collide. i.e. for any two VAs x and y with $\alpha_x = \alpha_y$
  - ¬ `roadCollision(x, y)`
  - if $x.t_a < y.t_a$ then $x.t_d < y.t_d$
- VAs with crossing trajectories must not collide. i.e. for any two VAs x and y with crossing trajectories
  - ¬ `intersectionCollision(x, y)`

We define the decision variables to be t_a and t_d for each VA. With the resulting plan, we calculate the changes in velocity and acceleration required to satisfy these decision variables for each VA.

3.1 Optimal Scheduling

We adopt an integer linear programming model.



We let our objective function be the total travel time through the intersection, which we aim to minimize.

$$\min \sum_{i=1}^{n} (v_i.t_d - v_i.t_a)$$

Evidently, since our decision variables can be broken down into integers, we can use ILP. As previously, t_a and t_d are our decision variables.

Once the ILP solver has returned the optimal t_a and t_d for each VA, we calculate and apply the changes in velocity and acceleration required to satisfy the plan.

4 Example Scenario and Results

Consider a cross intersection with only two directions of travel, East and South-bound. Assume there is only one lane for each direction of traffic flow as seen in figure 4.

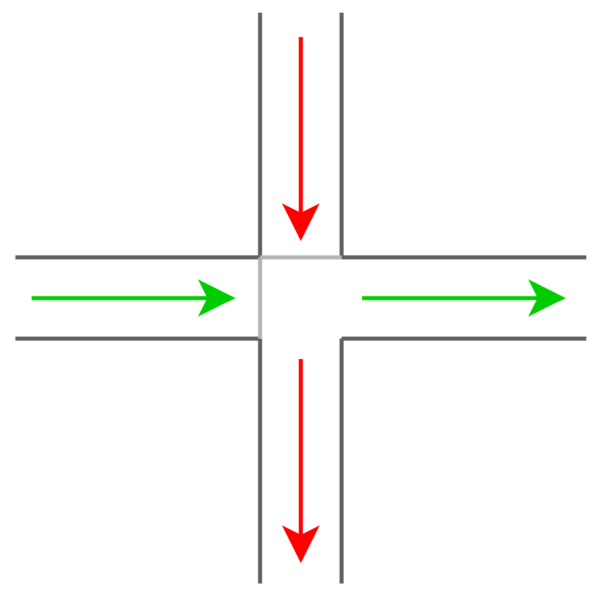


Figure 4. A simple scenario

4.1 Optimal Schedule

We make the following assumptions:

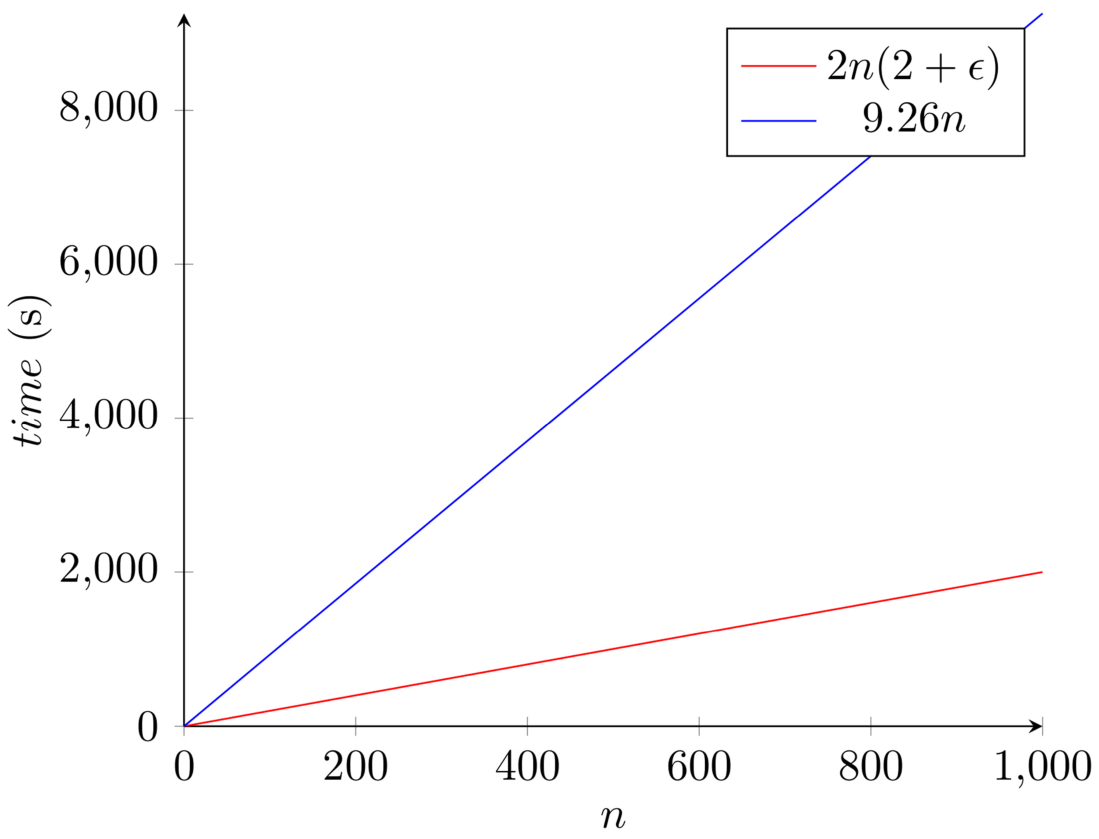
- Speed limit is 15ms^{-1}
- Each platoon has length 25m
- Width of each lane is 5m
- Time required to travel entirely through intersection is then (5 + 25)/15 = 2s
- Vehicles heading South-bound, $S = \{s_1, \ldots, s_n\}$
- Vehicles heading East-bound, $E = \{e_1, \ldots, e_n\}$

The optimal schedule can then be given by:

$$s_1.t_a = 0$$
$$s_1.t_d = s_1.t_a + 2 = 2$$
$$\forall\, i \in \{2, \ldots, n\}, s_i.t_a = s_{i-1}.t_d + (2 + \epsilon)$$
$$e_1.t_a = 2 + \epsilon$$
$$e_1.t_d = e_1.t_a + 2 = 4 + \epsilon$$
$$\forall i \in \{2, \ldots, n\}, e_i.t_a = e_{i-1}.t_a + (2 + \epsilon)$$
$$\forall v \in S \cup E, v.t_d = v.t_a + 2$$

Where ε is a safe clearing time between VAs. Clearly, this schedule minimizes the travel time.

Then, the total travel time is given by $2n(2 + \varepsilon)$ where n is the number of vehicles in each lane. We calculated that a modified stop-sign control mechanism gives $9.26n$ total travel time.



5 Conclusion and Future Work

We have shown that our scheduling algorithm can have significant time benefits over traditional intersection control methods.

We identified constraints in our problem and then provided a linear-programming based scheduling algorithm that minimizes the total time required for VAs to cross an intersection.

To understand the true effectiveness of our scheduling algorithms, simulations on a larger scale should be carried out and results compared with existing solutions. With this, we investigate different objective functions with factors such as congestion, energy consumption, and pollution.