Australian
National
University

Intelligent Intersection Control for Platoons of Autonomous Vehicles

William Shen, u609655@anu.edu.au

Supervisor: Jochen Renz

Motivation

- Traditional Intersection Control
  - Traffic lights, give way signs, stop signs, etc.
  - Can lead to congestion, large delays
  - Does not account for true demand
- Congestion is bad for society
  - Travel time, energy consumption, pollution
  - Social cost $16.5 billion AUD in 2015 (BITRE)
  - These social costs are avoidable

Motivation

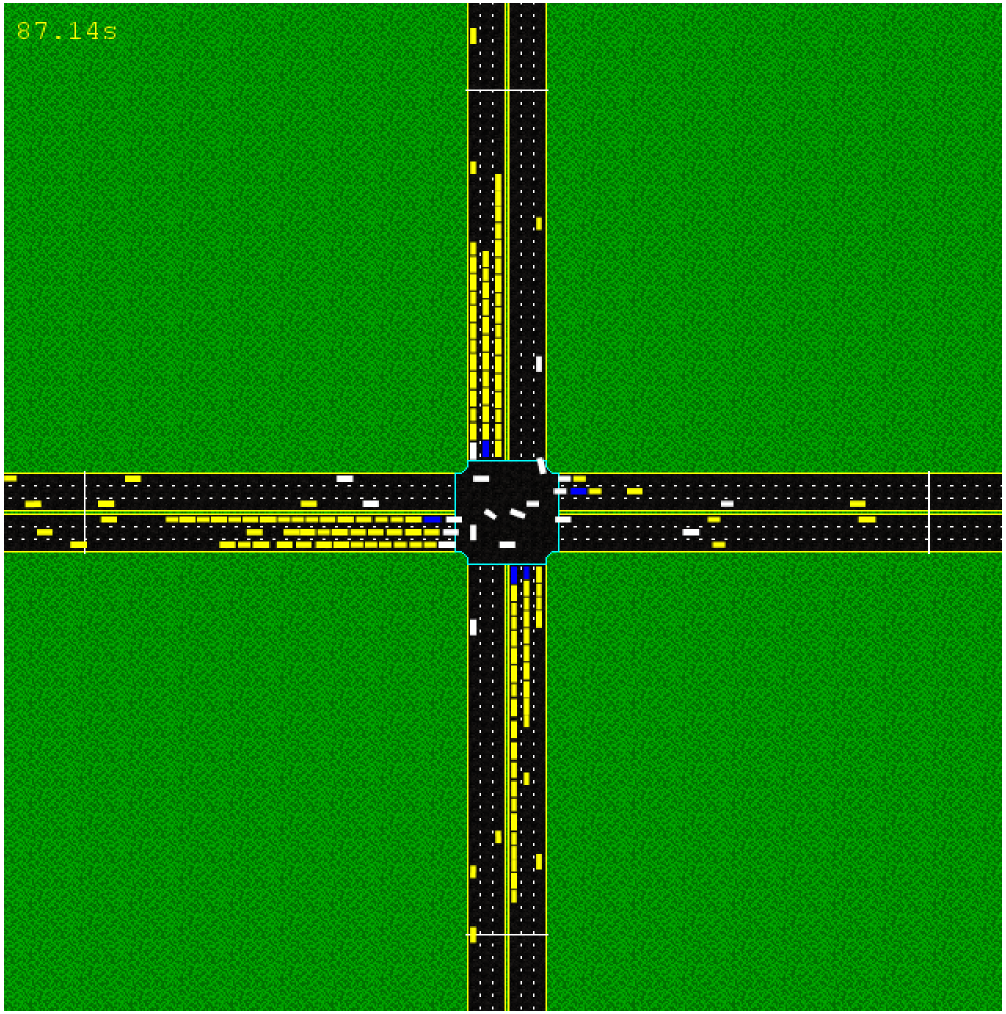
- Autonomous Vehicles
  - Self-driving, gaining more prominence
  - Co-ordinated intersection control, ‘intelligent’
- Traffic accidents at intersections
  - 50% of urban crashes (NCHRP)
  - 30% of rural crashes (NCHRP)

Related Work

- Centralized Solutions
  - Vehicle-to-Infrastructure (V2I) communication
  - Reservation-based (first come first served)
- Decentralized Solutions
  - Vehicle-to-Vehicle (V2V) communication
  - Cost-saving, no need for infrastructure
  - Reservation-based (FCFS, auction)

Autonomous Intersection Manager (AIM)

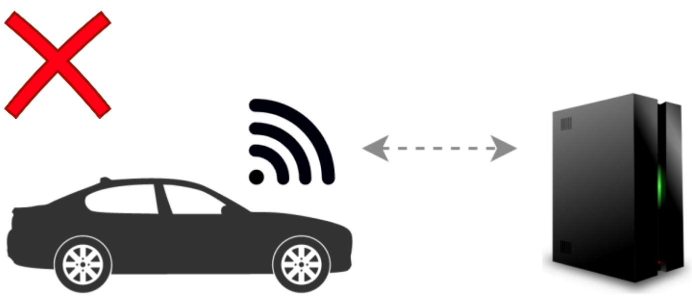
- University of Texas at Austin
- V2I Communication
- Central server at each intersection
- Vehicles send crossing requests to this server
- Can result in huge blockades
- http://www.cs.utexas.edu/~aim/



Simulation in AIM

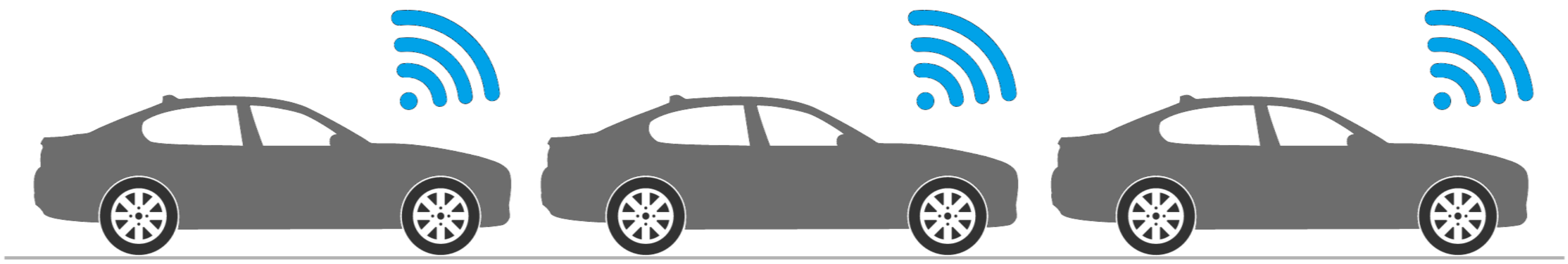
Our Approach

- Decentralized Communication
  - Vehicle-to-Vehicle (V2V) Communication
  - Reduced cost of implementation
  - Network-wide Communication



Our Approach

- Platooning
  - Grouping vehicles into ‘platoons’
  - Increased capacity on roads, less congestion
  - Reduced energy consumption, drag reduction



Problem Definitions

- Define a vehicle or a platoon (group of vehicles) as a **vehicular agent** (VA)
- A plan is the sequence of operations that is applied to the VAs approaching an intersection.
- A plan is valid if there are no collisions.
- A plan is optimal if it is valid and minimizes our objective function.

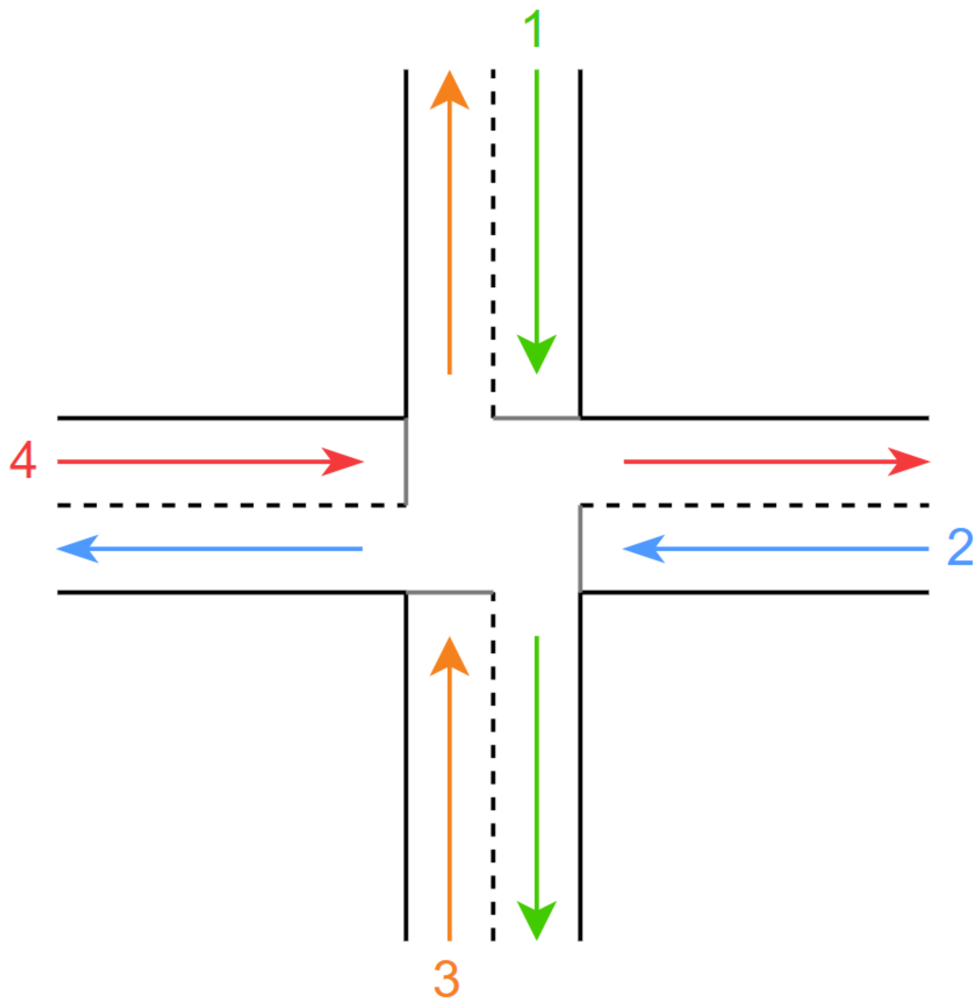
Problem Formulation

- Each vehicle has a:
  - $route$, a sequence of intersections
  - $next$, the next intersection in the route
  - t_a, estimated time of arrival to $next$
  - t_d, estimated time of departure from $next$
  - d, distance away from $next$
  - length, velocity, acceleration

Problem Formulation

- Each intersection has a ‘central’ area where the crossing of VAs take place
- We need to check for collisions here!
- Each VA has a trajectory (α, β) with respect to the intersection:
  - α is the unique lane ID
  - β is the turning intent at the intersection

Intersection Example



$$\alpha \in \{1, .., 4\}$$
$$\beta \in \{l, s, r\}$$

We only need to check VAs with **crossing** trajectories for collisions.

- $(3, r)$ and $(4, s)$
- $(1, r)$ and $(3, r)$

Non-crossing:

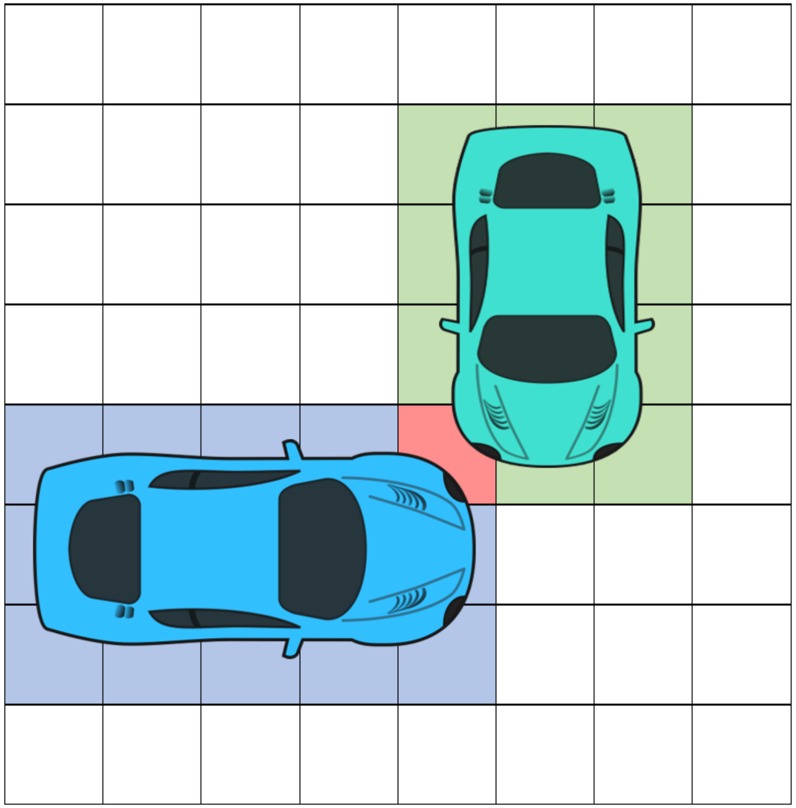
- $(1, s)$ and $(3, l)$

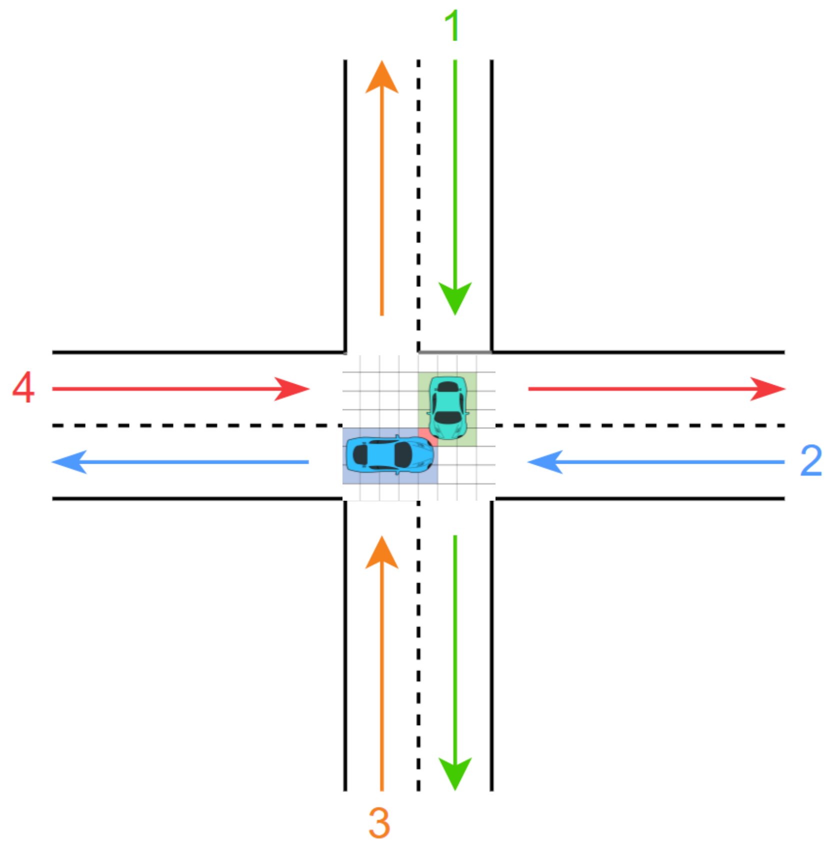
Collision Detection – within lanes

```
boolean roadCollision(VA x, VA y) {
    if (x.d == y.d) {
        return true;
    } else if (x.d < y.d) {
        return x.t_a + x.d / x.v < y.t_a;
    } else {
        return y.t_a + y.d / y.v < x.t_a;
    }
}
```

Australian National University

Collision Detection – intersections





Collision Detection – intersections

Given two VAs x and y:

1. Check if the trajectories of x and y are crossing, if non-crossing no collision and return.
2. Define a collision table for the intersection crossing area.
3. Step through time until one VA has cleared the intersection ($\min(x.t_d, y.t_d)$), updating and checking the collision table as we go.

Constraint-Based Scheduling

For all vehicles approaching the intersection:

- Vehicular agents travelling on the same lane must not collide in the lane and the intersection
  - Collision detection algorithm, and
  - if $x.t_a < y.t_a$ then $x.t_d < y.t_d$ (and vice-versa)
- Vehicular agents with crossing trajectories must not collide within the intersection
  - Collision detection algorithm

Accounting for Platoons

Let p represent a platoon, and $V = \{v_1, \dots, v_n\}$ represent the VAs that form or have been split to p

- $p.size = \sum_{v \in V}(v.size \text{ if } v \text{ is a platoon, else } 1)$
- $\forall x \in p.vehicles, p.v = x.v \wedge p.a = x.a$
- $\forall x \in \{1, \dots, p.size - 1\},\ p.vehicles[x].d + p.vehicles[x].l + \gamma = p.vehicles[x+1].d$

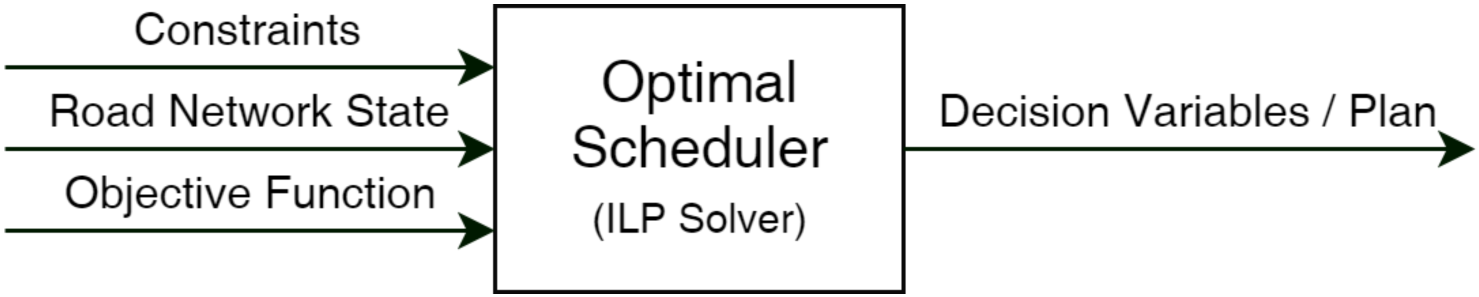
Where $\gamma > 0$ is the 'safe' distance we must keep between each vehicle in the platoon.

Optimal Scheduling

Adopt integer linear programming (ILP) approach:

- Decision variables t_a and t_d for each VA
- Let n be the total number of VAs approaching the intersection, denote each VA by v_i where $$i \in \{1, \dots, n\}$$
- Minimize travel time through the intersection
- Objective function: $\min \sum_{i=1}^{n} (v_i.t_d - v_i.t_a)$

Optimal Scheduling



Now, given the t_a and t_d for each VA, calculate the necessary changes in velocity (and acceleration) to satisfy this.

Therefore, we have our valid, optimal plan.

Communication Protocol

Once we have found a valid, optimal plan for a single intersection:

1. Each VA transmits its resulting state (V2V) after applying the plan (relative to next intersection)
2. This generates a ‘snapshot’ of all vehicles heading towards the other relevant sections
3. Carry out the optimal plan calculation and repeat.

Ideally, we agree on a plan early on to minimize potential disruptions.

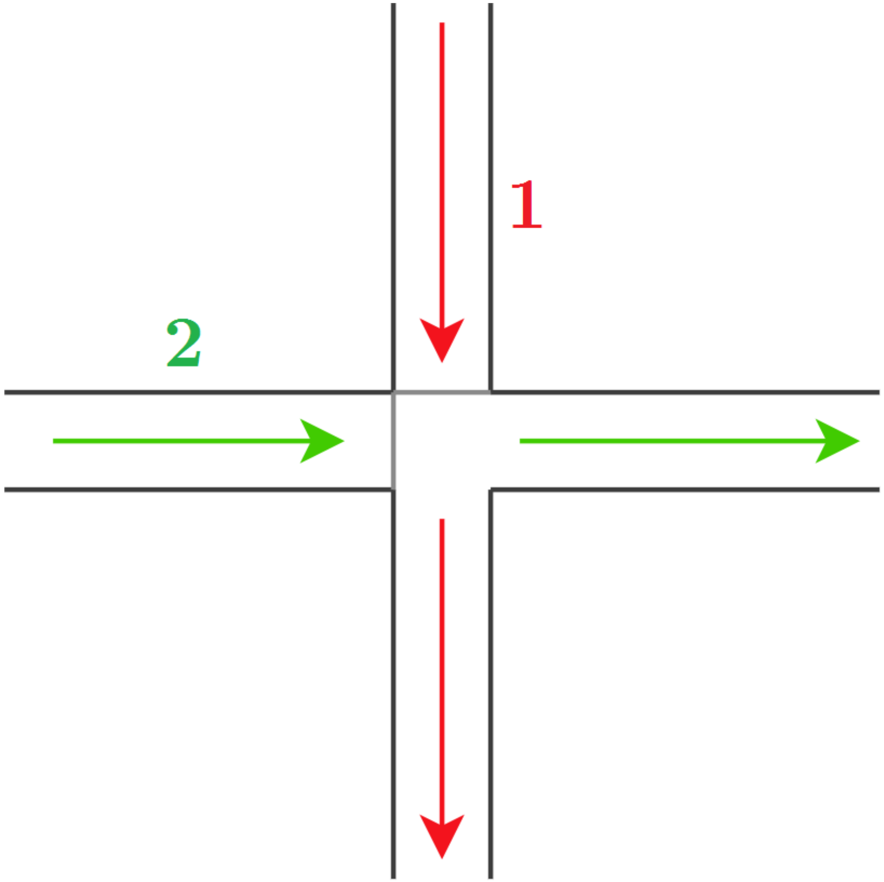
A Simple Example Scenario

Trajectories (α, β):

- $\alpha \in \{1, 2\}$
- $\beta \in \{s\}$

Assumptions:

- Speed Limit = 15ms^{-1}
- VA length = 25m
- Lane width = 5m
- Time to cross = 2s (at speed limit)



Scenario – Optimal Plan

Let vehicles heading South-bound be $S = \{s_1, \dots, s_n\}$, and East-bound be $E = \{e_1, \dots, e_n\}$

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

Scenario - Comparisons

Let n be the number of vehicles in each lane.

Total travel time for our scenario (in seconds):

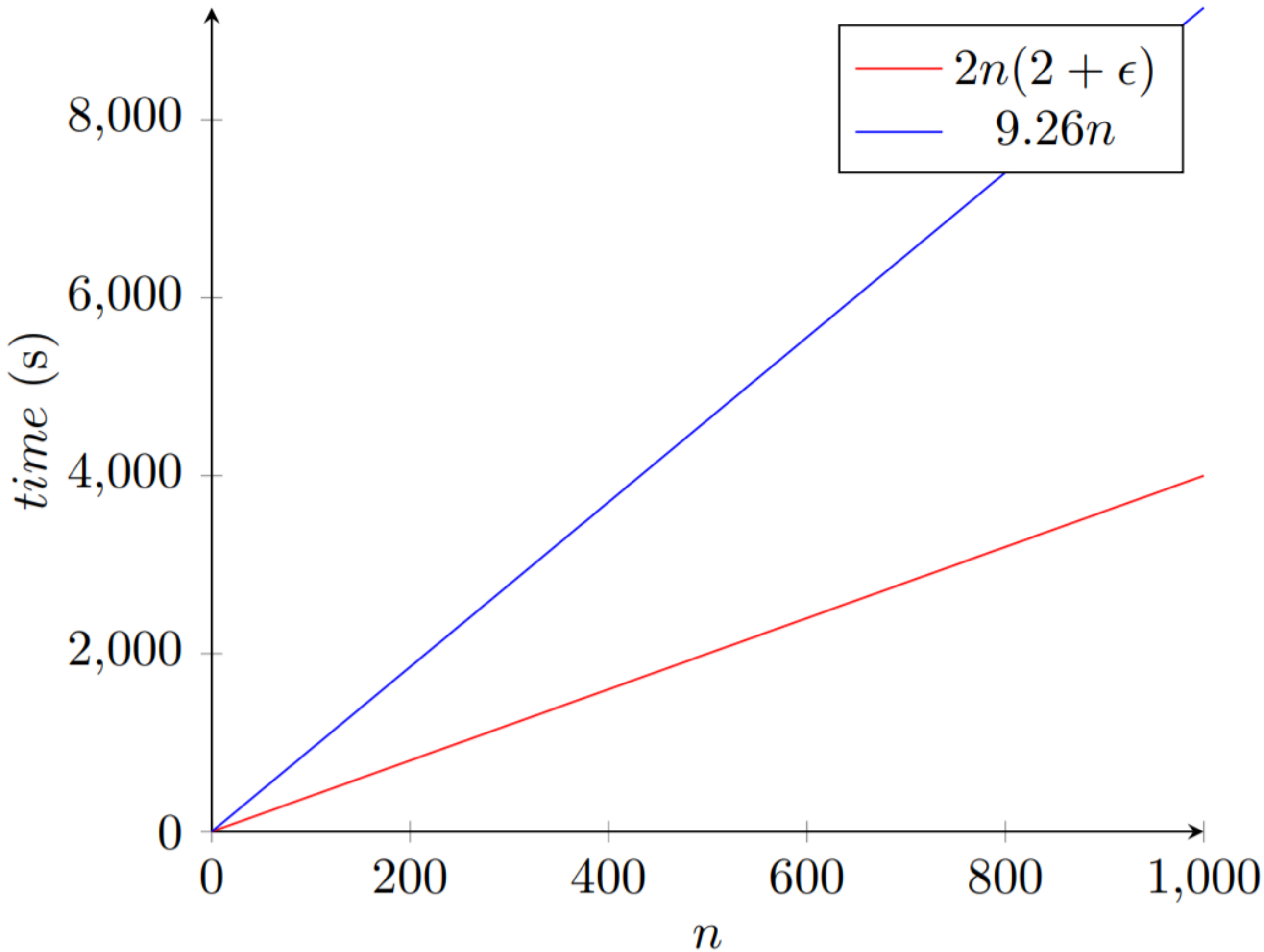
$$2n(2+\epsilon) = 4n \text{ (when } \epsilon = 0)$$

Total travel time for modified stop sign system (alternating 'approvals')

$$9.26n$$

In the stop sign system, vehicles have to stop.
Our algorithm does ≈ 2.3 times better.

Scenario - Comparisons



Results

- Our optimal scheduling algorithm can have significant benefits over traditional intersection control methods.
  – Of course, we could have shown this with more scenarios, traffic control methods, etc.
- We theorize that our algorithm will result in better plans than reservation algorithms
  – First come first served, auction-based, etc.

Future Work

- Simulations
  - Test against traditional control methods
  - Test against existing solutions
  - Measure different factors (pollution, total travel time, energy consumption, etc.)
- Consider different objective functions
- Improve platooning capabilities, consider demand functions, communication protocols

Thank you!

Questions?

Australian National University

References

- K. Dresner and P. Stone, "*A multiagent approach to autonomous intersection management*," Journal of Artificial Intelligence Research, vol. 31, pp. 591–656, 2008.
- D. Cosgove, *Traffic and congestion cost trends for Australian capital cities. Department of Infrastructure and Regional Development, Bureau of Infrastructure*, Transport and Regional Economics, 2015.
- T. R. Neuman, *Guidance for implementation of the AASHTO strategic highway safety plan. A guide for reducing collisions at signalized intersections*, 2004.