EINDHOVEN UNIVERSITY OF TECHNOLOGY

Platoon Forming Algorithms For Intelligent Street Intersections

Rik Timmerman, 02-07-2019
Joint work with Marko Boon

Applied^2 Probability Workshop

Outline

- Introduction
- Model description
  - Platoon forming algorithms
  - Speed regulation of vehicles
- Performance analysis
  - Mean delay
  - Fairness
  - Case study
- Conclusions

Self-driving vehicles introduction

Waymo is gearing up to put a lot more self-driving cars on the road

Kirsten Korosec @kirstenkorosec / 3 months ago

Comment

Self-driving vehicles introduction

Model assumptions

We assume the following

Model assumptions

We assume the following

- All vehicles are self-driving.

Model assumptions

We assume the following

- All vehicles are self-driving.
- There is a central controller at the intersection.

Model assumptions

We assume the following

- All vehicles are self-driving.
- There is a central controller at the intersection.
- There is a control region around the intersection.

Model assumptions

We assume the following

- All vehicles are self-driving.
- There is a central controller at the intersection.
- There is a control region around the intersection.
- Speed regulation of vehicles is possible.

Platoon forming algorithms

Platoon forming algorithms

The exhaustive discipline:

Platoon forming algorithms

The exhaustive discipline:
As long as no vehicles are delayed we do not form platoons bigger than size 1.

Platoon forming algorithms

The exhaustive discipline:
As long as no vehicles are delayed we do not form platoons bigger than size 1.
If a vehicle is delayed, we keep adding vehicles

Platoon forming algorithms

The exhaustive discipline:
As long as no vehicles are delayed we do not form platoons bigger than size 1.
If a vehicle is delayed, we keep adding vehicles

Platoon forming algorithms

The exhaustive discipline:
As long as no vehicles are delayed we do not form platoons bigger than size 1.
If a vehicle is delayed, we keep adding vehicles

until a gap arises

Platoon forming algorithms

The exhaustive discipline:
As long as no vehicles are delayed we do not form platoons bigger than size 1.
If a vehicle is delayed, we keep adding vehicles

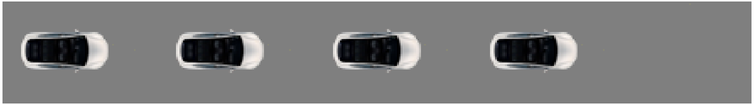
until a gap arises

Platoon forming algorithms

The gated discipline

Platoon forming algorithms

The gated discipline



Platoon forming algorithms

The gated discipline

Platoon forming algorithms

The gated discipline

Speed regulation of arriving vehicles

Speed regulation of arriving vehicles

$$\texttt{MotionSynthesize}(z_{i,k}(t'_0), t'_0, t'_f, y) :=$$

$$\begin{aligned}
\underset{x:\,[t'_0,t'_f]\to\mathbb{R}}{\arg\min} \quad & \int_{t_0}^{t_f} |u(t)|dt \\
\text{subject to} \quad & \ddot{x}(t) = u(t), \text{ for all } t \in [t'_0, t'_f]; \\
& 0 \le \dot{x}(t) \le v_m, \text{ for all } t \in [t'_0, t'_f]; \\
& |u(t)| \le a_m, \text{ for all } t \in [t'_0, t'_f]; \\
& |x(t) - y(t)| \ge l, \text{ for all } t \in [t'_0, t_f]; \\
& x(t'_0) = x_{i,k}(t'_0); \quad \dot{x}(t'_0) = \dot{x}_{i,k}(t'_0); \\
& x(t'_f) = 0; \quad \dot{x}(t'_f) = v_m,
\end{aligned}$$

(After Miculescu and Karaman, 2016)

Speed regulation of arriving vehicles

The algorithm controls the applied acceleration to make sure:

Speed regulation of arriving vehicles

The algorithm controls the applied acceleration to make sure:

- a vehicle is at the right time at the right place and at the right speed;

Speed regulation of arriving vehicles

The algorithm controls the applied acceleration to make sure:

- a vehicle is at the right time at the right place and at the right speed;
- the amount of acceleration is minimized;

Speed regulation of arriving vehicles

The algorithm controls the applied acceleration to make sure:

- a vehicle is at the right time at the right place and at the right speed;
- the amount of acceleration is minimized;
- no collisions occur.

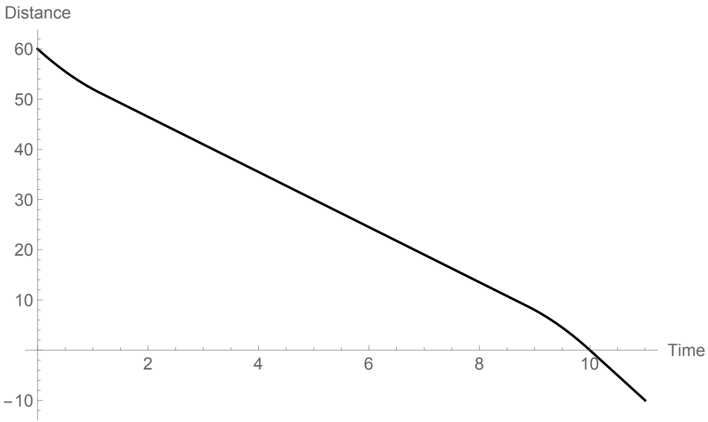
Speed regulation of arriving vehicles

This results in the following control:

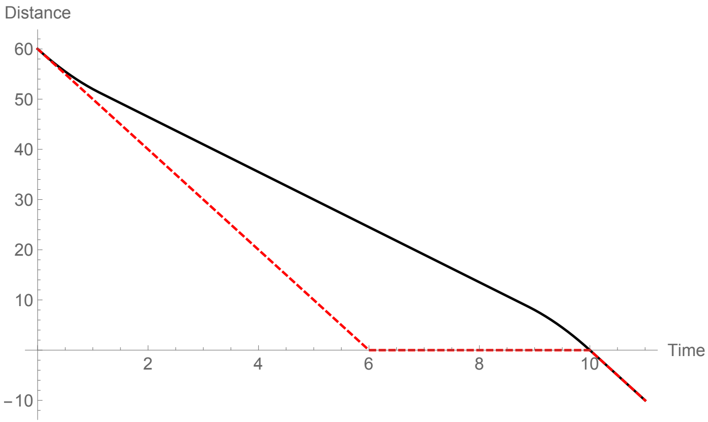
$$a(t) = x''(t) = \begin{cases} -a_m & \text{if } t_0 \leq t < t_1, \\ 0 & \text{if } t_1 \leq t < t_2, \\ a_m & \text{if } t_2 \leq t < t_3, \\ 0 & \text{if } t_3 \leq t < t_f, \end{cases}$$

for certain known and computable t_1, t_2 and t_3 and fixed t_f.

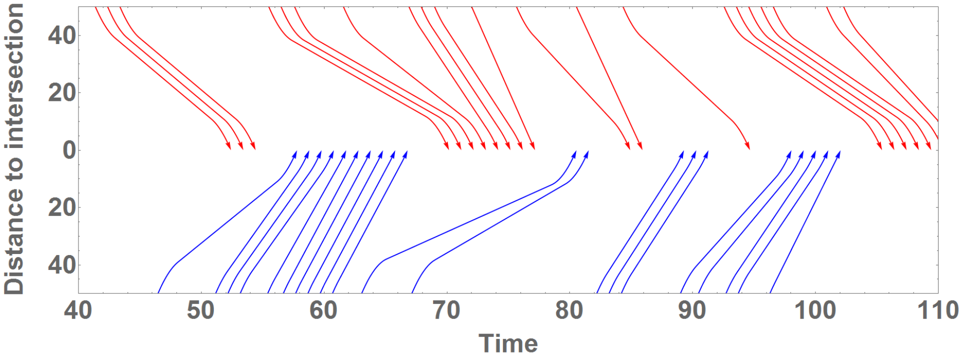
Speed regulation and a link to queueing



Speed regulation and a link to queueing



Example with two streams of vehicles



Approximations – mean delay

Our intersection is a polling model, but slightly different than the standard version of polling models.

Approximations – mean delay

Our intersection is a polling model, but slightly different than the standard version of polling models.
We derive approximations for the mean delay.

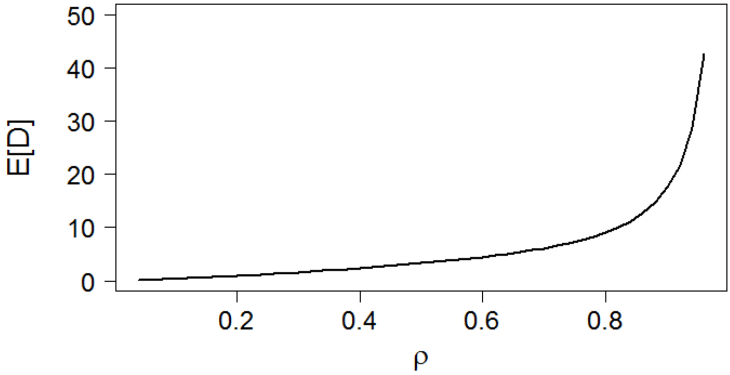
Approximations – mean delay

Our intersection is a polling model, but slightly different than the standard version of polling models.
We derive approximations for the mean delay. We use the following interpolation formula (Boon et al, 2011).

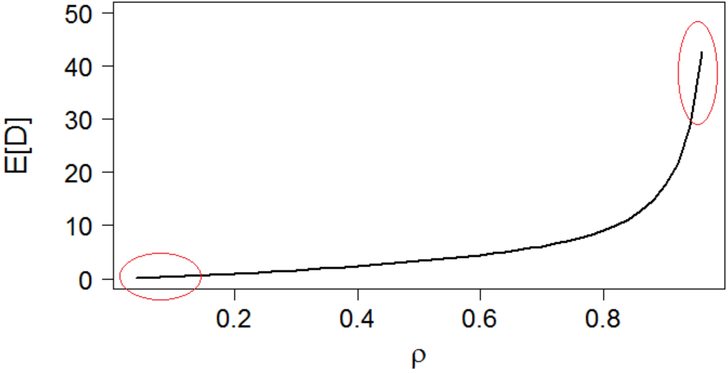
$$\mathbb{E}[D_i] = \frac{K_{0,i} + K_{i,1}\rho_i + K_{2,i}\rho_i^2}{1 - \rho_i},$$

where ρ_i is the load on the intersection and with carefully chosen constants $K_{0,i}$, $K_{1,i}$ and $K_{2,i}$.

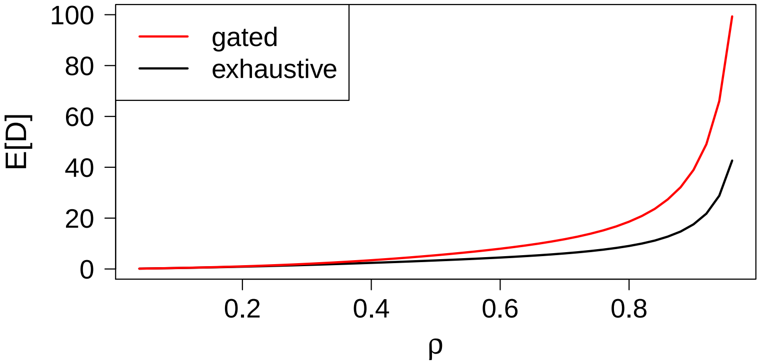
Approximations – mean delay



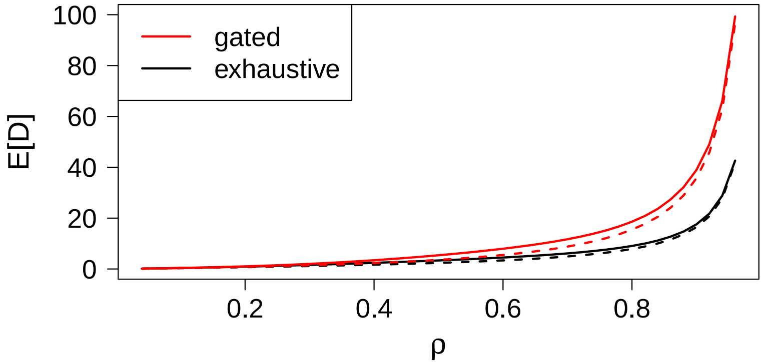
Approximations – mean delay



Approximations – mean delay



Approximations – mean delay



Simulations – fairness

Fairness is an important property as well.

Simulations – fairness

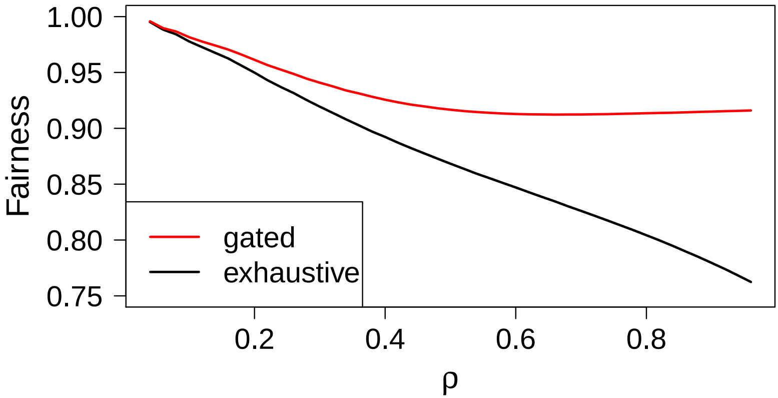
Fairness is an important property as well.

We define fairness as:

$$F = \frac{\mathbb{E}[Y]}{\mathbb{E}[X]},$$

where X is the number of cars seen by an arbitrary arriving vehicle and Y is the number of cars from those X that leave the system before that arbitrary car does (Shapira and Levy, 2016)

Simulations – fairness

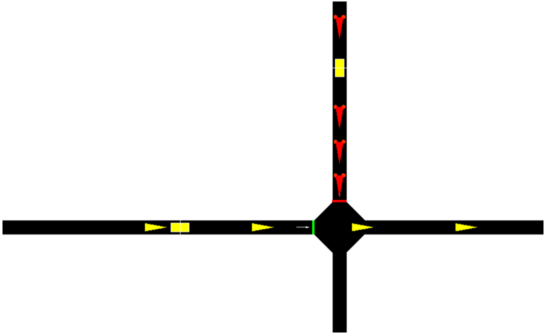


Case study – SUMO

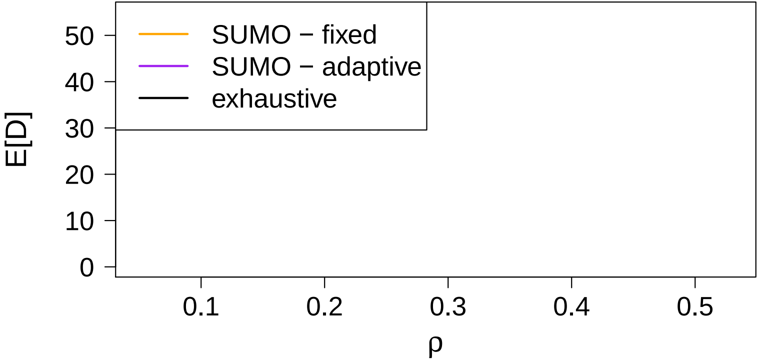
To represent the current traffic situation with conventional cars, we perform a simulation in SUMO

Case study – SUMO

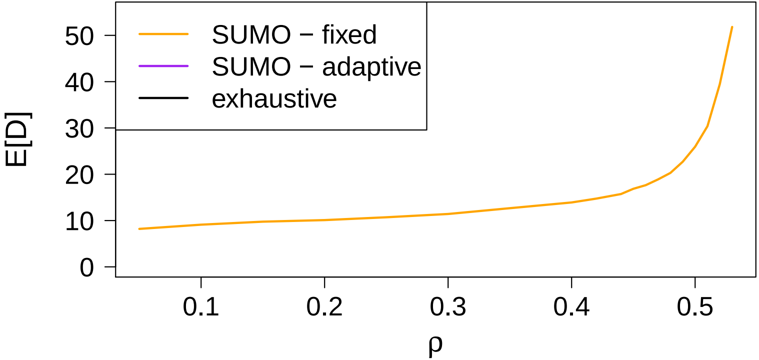
To represent the current traffic situation with conventional cars, we perform a simulation in SUMO



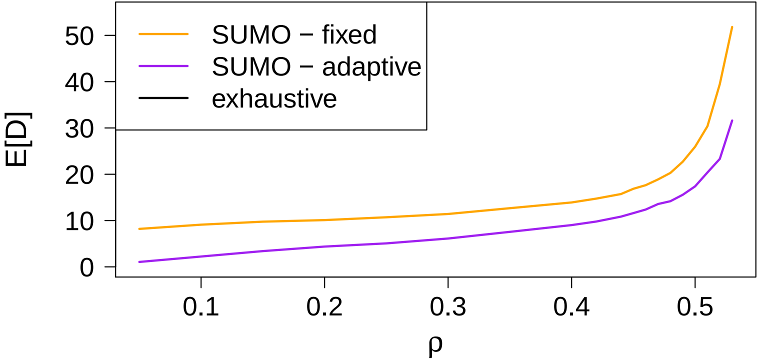
Case study – SUMO



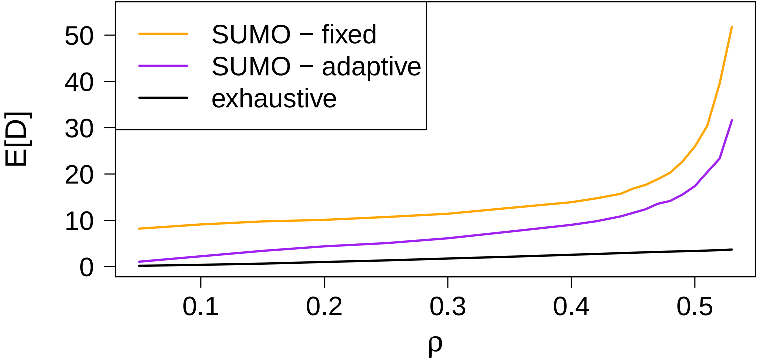
Case study – SUMO



Case study – SUMO



Case study – SUMO



Conclusions

- PFAs are closely connected to polling models.

Conclusions

- PFAs are closely connected to polling models.

- The exhaustive PFA is performing very close to optimality with respect to mean delay.

Conclusions

- PFAs are closely connected to polling models.
- The exhaustive PFA is performing very close to optimality with respect to mean delay.
- There is a lot to be gained, using platoon forming and speed regulation.

Conclusions

- PFAs are closely connected to polling models.
- The exhaustive PFA is performing very close to optimality with respect to mean delay.
- There is a lot to be gained, using platoon forming and speed regulation.
- We have been looking at an *isolated* intersection. What about networks?

Thank you for your attention!