Integration of Microsimulation and Optimized Autonomous Intersection Control

CTS Conference – November 1st
Jack Olsson

University of Minnesota
Driven to Discover℠

Presentation Outline

- Introduction
- Methodology
- Integration with Microsimulation
- Analysis Scenarios
- Results
- Conclusions and Future Work

University of Minnesota
Driven to Discover®

Introduction

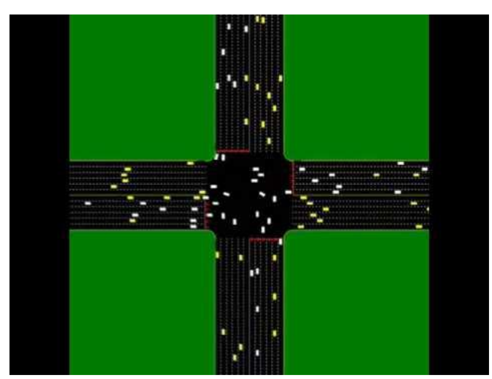
- Autonomous Intersection Management (AIM)
  - What is it?
  - Benefits
    - Improved Efficiency
    - Improved Safety
- Integration with Microsimulation
  - Purpose
  - What makes the use of microsimulation unique in this work?

https://www.sandiegocounty.gov/content/sdc/dpw/transportation/lightsout.html

UNIVERSITY OF MINNESOTA
Driven to Discover®

Introduction

- Research Contribution
  - Develop more realistic AIM optimization model
  - Integrate AIM with off-the-shelf microsimulation software (Aimsun)
  - Utilize microsimulation to analyze new AIM model against other forms of intersection control

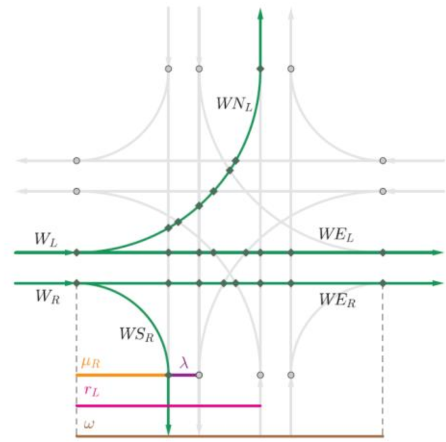


The University of Texas at Austin Computer Science Department.

UNIVERSITY OF MINNESOTA
Driven to Discover®

Methodology

- Conflict-point reservation-based AIM optimization (AIM+)
  - Based off the formulation from Levin and Rey (2017)
  - Improves formulation to give intersection manager more flexibility in optimizing vehicle paths



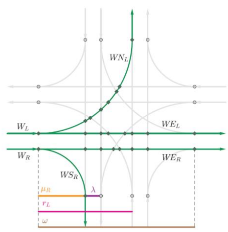
Methodology

- Model assumptions:
  - Vehicles utilize constant acceleration rate from first conflict point to last conflict point (intersection exit point)
    - Can use kinematics
  - Vehicle starts at rest at initial conflict point
    - Maintains convexity of formulation
    - Location of initial conflict point = 200 ft upstream

Methodology

- Mixed-integer linear program (MILP)
- Uses Gurobi for optimization solver

- Decision variables:
  – Time of arrival at every conflict point
  – Vehicle acceleration



$$\min \sum_{i\in\mathcal{V}} f_i\left(t_i(c_i^N) + \tau_i(c_i^N)\right)$$

subject to

$$t_i(c_i^0) \geq e_i \quad \forall\, i \in \mathcal{V}$$

$$\tau_i(c_i^0) = \frac{L_i}{w} + \frac{L_i(\lambda_i)}{\sqrt{2d_i(c_i^0, c_i^1)}} \quad \forall\, i \in \mathcal{V}$$

$$\tau_i(c_i^n) = \frac{L_i}{w} + \frac{L_i(t_i(c_i^n) - t_i(c_i^{n-1}))}{d_i(c_i^{n-1}, c_i^n)} + \frac{D_i}{2\underline{U_i}} \quad \forall\, i \in \mathcal{V}, \forall c_i^n \in \rho_i : n \in [1, \mathrm{N}_i]$$

$$t_i(c_i^n) - t_i(c_i^0) = \lambda_i \sqrt{2d_i(c_i^0, c_i^n)} \quad \forall\, i \in \mathcal{V}, \forall c_i^n \in \rho_i : n \in [1, \mathrm{N}_i]$$

$$\sqrt{\frac{1}{\overline{a_i}}} \leq \lambda_i \leq \sqrt{\frac{1}{\underline{a_i}}} \quad \forall i \in \mathcal{V}$$

$$\frac{2d_{i(}c_i^0, c_i^N)}{\overline{U_i}} \leq t_i(c_i^N) - t_i(c_i^0) \leq \frac{2d_{i(}c_i^0, c_i^N)}{\underline{U_i}} \quad \forall i \in \mathcal{V}$$

$$t_i(c_i^n) + \tau_i(c_i^n) \leq t_j(c_j^n) \quad \forall i, j \in \mathcal{V}: c_i^0 = c_j^0, e_i < e_j, \forall c_i^n \in \rho_i \cap \rho_j : n \in [1, \mathrm{N}_i]$$

$$t_i(c_i^n) + \tau_i(c_i^n) - t_j(c_j^n) \leq \left(1 - \delta_{ij}(c_i^n)\right) M_{ij} \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, \forall c_i^n \in \rho_i \cap \rho_j : n \in [1, \mathrm{N}_i]$$

$$\delta_{ij}(c_i^n) + \delta_{ji}(c_i^n) = 1 \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, i < j, \forall c_i^n \in \rho_i \cap \rho_j : n \in [1, \mathrm{N}_i]$$

$$\delta_{ij}(c_i^n) \in \{0,1\} \quad \forall i, j \in \mathcal{V}: c_i^0 \neq c_j^0, \forall c_i^n \in \rho_i \cap \rho_j : n \in [1, \mathrm{N}_i]$$

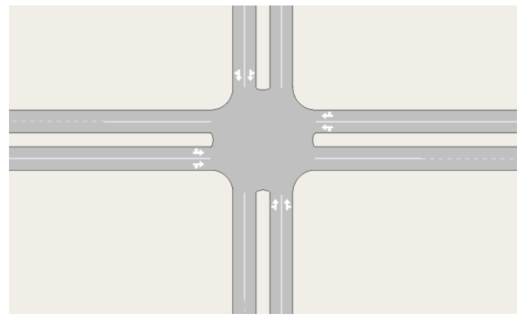
Integration with Microsimulation

- Aimsun Application Programming Interface (API)
  – Allows for creation of script that can interface with Gurobi
- Coordination between Aimsun and Gurobi during simulation
  – Output of vehicle information from Aimsun upon entering network →
- Aimsun output becomes Gurobi input

Parameter	Source
Vehicle Identification	Aimsun API
Vehicle Length, L_i	Aimsun API
Vehicle Width, D_i	Aimsun API
Vehicle Approach	Aimsun API
Vehicle Movement	Aimsun API
Earliest Arrival Time, e_i	Aimsun API (based on a calculation using location and speed)
Minimum Velocity, $\underline{U_i}$	User-Defined
Maximum Velocity, $\overline{U_i}$	Aimsun API
Minimum Acceleration, $\underline{a_i}$	User-Defined
Maximum Acceleration, $\overline{a_i}$	Aimsun API
Backwards Wave Speed, w	User-Defined

Integration with Microsimulation

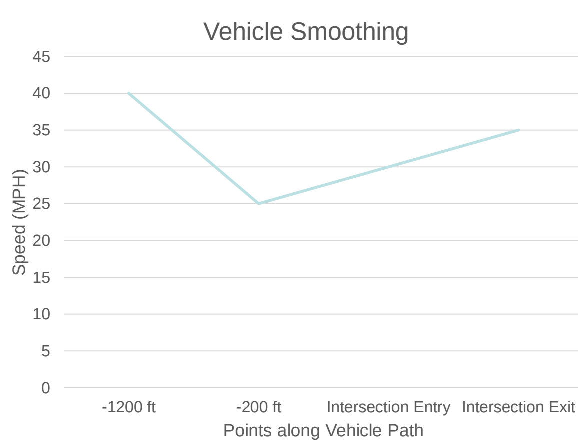
- Aimsun Modeling Choices
  - Constant initial velocity
  - Lane-changing behavior



UNIVERSITY OF MINNESOTA
Driven to Discover®

Integration with Microsimulation

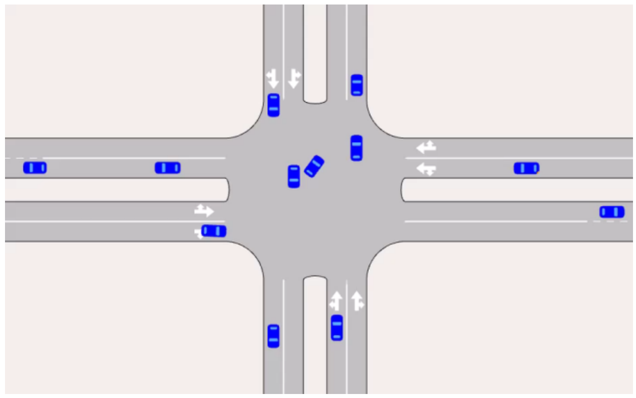
- "Velocity smoothing"*
  - Provides alternative so vehicles in Aimsun do not physically have to come to a stop at initial conflict point
  - Piecewise acceleration function
  - Ensures that vehicle arrives at intersection at the assigned time and velocity to ensure collision-free traversal through intersection



UNIVERSITY OF MINNESOTA
Driven to Discover®

Integration with Microsimulation

AIM+ in action: (intersection demand = 8,800 vph)



UNIVERSITY OF MINNESOTA
Driven to Discover®

Analysis Scenarios

- Measures of effectiveness (MOEs): delay, speed
- Parameter: Intersection demand

- AIM+ Sensitivity Analysis
- AIM+ vs Conventional Traffic Signal
  - Signal timing obtained using Synchro

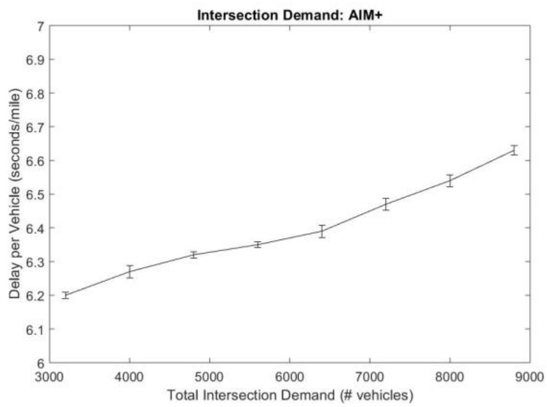
- AIM+ vs AIM “First-Come First-Served” (FCFS)

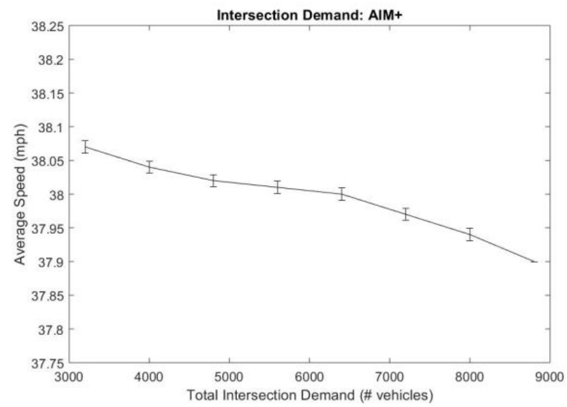
- Multiple 1-hour simulations run for each scenario

UNIVERSITY OF MINNESOTA
Driven to Discover®

Results: AIM+ Sensitivity Analysis

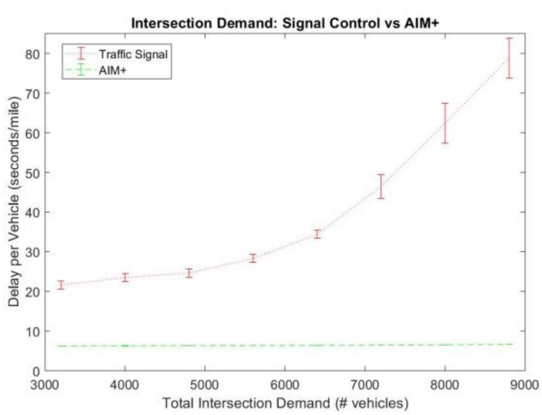
Intersection Demand
(40 mph, 0 buffer, 22%/56%/22% LTR, 1:1:1:1)

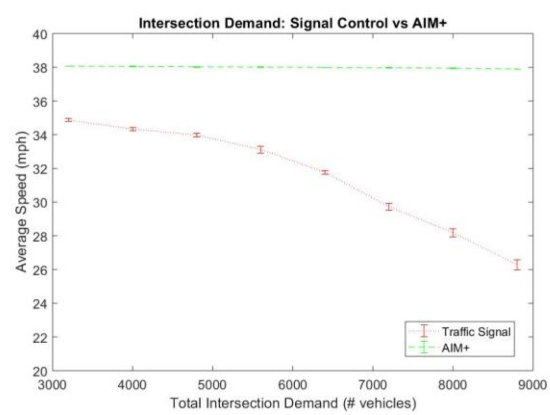




UNIVERSITY OF MINNESOTA
Driven to Discover®

Results: Traffic Signal vs AIM+

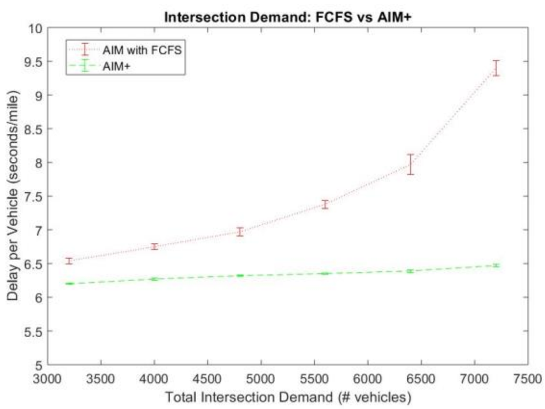


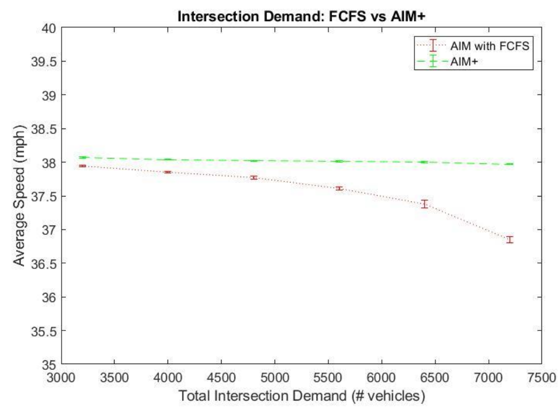


Intersection Demand
(40 mph, 0 buffer, 22%/56%/22% LTR, 1:1:1:1)

UNIVERSITY OF MINNESOTA
Driven to Discover®

Results: AIM FCFS vs AIM+





Intersection Demand
(40 mph, 0 buffer, 22%/56%/22% LTR, 1:1:1:1)

Conclusions

- AIM+ robust to changing parameters
  - Delay increases of only several seconds (sometimes only tenths of a second) across scenarios
  - Speed decreases of only tenths of a mile per hour across scenarios
- 71%-91% reduction in delay with AIM+ over signals
- 5%-31% reduction in delay with AIM+ over AIM FCFS
- Correspondingly higher speeds with AIM+

Conclusions

- Future work:
  - Improve computational efficiency of Aimsun API and Gurobi Solver
    - Can test higher demands and larger safety buffers
  - Improve formulation to allow for fluctuating vehicle accelerations
  - Extend applicability of AIM+ to multi-intersection networks