Learning Policy Selection for Autonomous Intersection Management

Pierre-Luc Grégoire

Département d'Informatique et de Génie Logiciel
gregoire@damas.ift.ulaval.ca
http://damas.ift.ulaval.ca

19 octobre 2007
Problem of Congestion

Congestion is one of the leading causes of lost productivity and decreased standard of living in urban settings.

**Congestion Statistics**

- Time spent waiting in traffic has increased from 16 hours per capita to 46 hours per capita since 1982.
- The annual financial cost of traffic congestion was of $63 billions dollars in 2002.
- Each year, Americans burn approximately 5.6 billion gallons of fuel while idling in heavy traffic.
Motivation

- Traffic lights often require that cars remain stopped even when no cars are present on the intersection road.
- In future, cars will be controlled by agents.
- Goal: put more vehicle in the intersection at same time.

Approach

- MAS-based approach to alleviating traffic congestion at intersection.
- Minimize centralized infrastructure.
- Two types of agents: vehicles and intersections.
- Communication between vehicles and intersections.

Solution: A Reservation-Based Intersection Control Mechanism
1 Introduction
   - Problem of Congestion
   - Motivation, approach and solution

2 Reservation-Based Intersection Control Mechanism
   - Original System
   - Improved System with Communication Protocol and Algorithm
   - Human-Usable Control Policies

3 Learning Policy Selection
   - Goal: adapt to traffic condition
   - Smoothly Switching Between Two Policies
   - Classifier
   - Experimental Results

4 Conclusion
Reservation-Based Intersection Control

Intersection Control Mechanism

- Communicate directly to *driver agents* piloting autonomous vehicles.
- Driver agent call ahead to an agent stationed at the intersection.
- It tries to reserve a region of space-time in the intersection.
- In its message, it includes:
  - Physical qualities and capabilities of the vehicle.
  - Predicted arrival time.
  - Velocity and desired direction of travel.
- The *Intersection Manager*, using an *Intersection Control Policy*, decides whether or not to grant the driver agent’s request.

Vehicle Rules

- Once a reservation is made, the driver agent must only enter the intersection in accordance with the parameters of the reservation.
- If the driver agent determines that this is not possible, it must either cancel the reservation or change the reservation.
Simulator

Environment

- Timebased simulator which model each car individually.
- The simulator (not including the intersection manager) runs with two main parameters:
  - Number of lanes traveling in each of the four directions.
  - Probabilities of attempting to spawn a vehicle in each direction (independently) at the beginning of each time step.

Driver Agent Behavior

- Safe gap of one second between each vehicle.
- Three possible actions: Accelerate, Decelerate and Coast.
- Each vehicle computes its own delay.
Grid Granularity

- The intersection is divided into an $n \times n$ grid of reservation tiles, where $n$ is called the *granularity* of the reservation system.
- Each tile can be reserved by one car per time step.

Processing of requests

(a) Successful reservation

(b) Failed reservation
Performance Metric

Delay

Delay is the primary metric to consider: what effect does the presence of the intersection have on the overall journey of a vehicle?

We want to insure that:

1. Average case is not bad.
2. Worst case is not too bad.

Average Delay

\[ \frac{1}{|C|} \sum_{v_i \in C} t(i) - t_0(i) \]

Maximum Delay

\[ \max_{v_i \in C} t(i) - t_0(i) \]
Results: Traffic Light vs Reservation-Based System

Scenario
- Ran a million steps of the simulator for an increasing car spawning probability.
- Reservation-based system has 1 tile.
- Each direction has 1 lane.

Results
- For lighter traffic, a shorter period is strictly better.
- The delay levels off because at this point the cars are so backed up that the simulation cannot create any more of them.
Lack in the original model

Problem

- No turns allowed.
- No detailed communication protocol.
- Vehicles were not allowed to accelerate in intersection.

Desirable Properties

- Vehicle agents must have more flexibility.
- The agents should only have access to information that can be reliably obtained with current technology.
- Communication failure (dropped messages) should not violate the system’s safety properties.
- The system should incorporate a simple communication protocol that allows agents to know only a minimal amount about each other.
- Every vehicle should eventually make it through the intersection (i.e. no deadlocks or starvation).
Detailed Protocol: Message Types

Vehicle → Intersection

- **REQUEST**: Contains properties of vehicle and of proposed reservation.
- **CHANGE-REQUEST**: Same as REQUEST but with updated parameters.
- **CANCEL**: Cancel its reservation.
- **RESERVATION-COMPLETED**: Advise it completed its traversal of the intersection.

Intersection → Vehicle

- **CONFIRMATION**: Is a response to a REQUEST or CHANGE-REQUEST message. Can contains a counter-offer. The reservation parameters in this message are implicitly accepted by the vehicle, and must be explicitly cancelled if the driver agent of the vehicle does not approve.
- **REJECTION**: Reject the request or change request and informs if the vehicle has to stop at the front of the intersection.
- **ACKNOWLEDGMENT**
**Vehicle Actions**

1. Can’t enter without reservation.
2. Must absolutely try to satisfy reservation parameters.
3. Only send one message at time.
4. Respect protocol (meaning of message types).

**Intersection Actions**

1. Must respond to a REQUEST or CHANGE-REQUEST message by a CONFIRMATION or REJECTION message.
2. A CONFIRMATION message guarantee that the vehicle will be safe while crossing the intersection.
First Come, First Serve (FCFS) Intersection Algorithm

```
function FCFS-RECEIVEMsg(msg)

    Inputs : msg : A vehicle message.
    Statics : reservations : A map of vehicles ID with reservations.

    if msg.type = REQUEST || msg.type = CHANGE-REQUEST then
        reservation = SIMULATEJOURNEYACCELARATION(msg.vehicle_parameters)
        if reservation = CONFLICT then
            reservation = SIMULATEJOURNEYCONSTVELOCITY(msg.vehicle_parameters)
        end if
        if reservation ≠ CONFLICT then
            reservations[msg.vehicle_ID] = reservation
            SENDMESSAGE(CONFIRMATION, reservation)
        else
            SENDMESSAGE(REJECTION)
        end if
    else
        if msg.type = CANCEL || msg.type = RESERVATION-COMPLETED then
            reservations.REMOVE(msg.vehicle_ID)
            SENDMESSAGE(ACKNOWLEDGMENT)
        end if
    end if

end function

```
Optimism and Pessimism

- Original System: Always use the current velocity.
- Improvement: the notion of an optimistic or pessimistic driver agent.

- An optimistic agent makes a reservation assuming it will immediately get to accelerate to full speed.
- An agent which no longer finds itself stuck behind a slower vehicle will become optimistic and attempt to make a new, earlier reservation.
- A pessimistic agent assumes it will be stuck at its current velocity until it reaches the intersection.
- If an agent has to cancel its reservation because there is no way for it to arrive on time, it becomes pessimistic.
Driver Agent Communication

Cancellation Policy

- **Original System**: Determines if reservation is valid by using its current velocity.
- **Improvement**: Agent only cancels a reservation if there is absolutely no physical way it could reach the intersection on time.

Impact of improvements on original system

The improvements to the driver agent drastically reduced both the average number of reservations made as well as the average number of messages transmitted.

<table>
<thead>
<tr>
<th></th>
<th>Messages</th>
<th>Reservations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>560.85</td>
<td>165.89</td>
</tr>
<tr>
<td>After</td>
<td>5.97</td>
<td>1.02</td>
</tr>
</tbody>
</table>
**Notion of others control policy**

**Differents Intersection Control Policy**

- **First Come, First Serve (FCFS)**
  - As seen in previous slide.

- **Traffic Light**
  - When the traffic light receives a REQUEST message, it examines the arrival time in the message. It then calculates the next time after this that the light for the direction, turn, and lane of the sending vehicle will be green and responds with a CONFIRMATION message that reflects this information.

- **Stop sign**
  - The stop sign is exactly like the reservation system, except that it only accepts reservations from vehicles that are stopped at the intersection. Any other reservation requests are rejected with a message indicating the vehicle must stop at the intersection.

- **Overpass**
  - The overpass accepts all REQUEST and CHANGE-REQUEST messages exactly as they are, sending corresponding CONFIRMATION messages.
Results of Reservation-Based System

Trip times for varying amounts of traffic

![Graph showing trip times for varying amounts of traffic with different systems and probabilities.]

Intersection Management  Pierre-Luc Grégoire
Integrate Human Drivers

**FCFS-Light**

- Adds standard traffic lights at intersection.
- Determines which tiles are in use by the light system at any given time and rejects all requests that cross these tiles.

**All-Lanes Off-tiles**

- Diagrams showing traffic flow with off-tiles.

**One-Lane Off-tiles**

- Diagrams showing traffic flow with one-lane off-tiles.
Results of Human Drivers integration

Average Delay

- 100% Human
- 5% Human
- 10% Human
- 1% Human
- Fully Autonomous

Graph shows the impact of human drivers on average delay with increasing traffic load.
Sommaire

1 Introduction
   - Problem of Congestion
   - Motivation, approach and solution

2 Reservation-Based Intersection Control Mechanism
   - Original System
   - Improved System with Communication Protocol and Algorithm
   - Human-Usable Control Policies

3 Learning Policy Selection
   - Goal : adapt to traffic condition
   - Smoothly Switching Between Two Policies
   - Classifier
   - Experimental Results

4 Conclusion
Goal

Problem

- The current system accommodates human drivers, but only with a large constant efficiency penalty.
- If the proportion of human drivers decreases, the system cannot exploit the more favorable conditions.
- Measuring the current proportion of human drivers at an intersection would necessitate expensive infrastructure.

Solution

- Collect information about humans indirectly, via the parameters of the requests made by autonomous vehicles.
- Use a classifier on these data to select the best policy.
Smoothly Switching Between Two Policies

Greedy Solution: turn all lights red and refuse all reservation until the intersection is empty.

More efficient solution

- Each policy $P$ must keep track of the latest time $lasp_p$ for which any vehicle could be in the intersection.
  - Initialize $lasp_p$ at the current time.
  - Each time $P$ grants a reservation, it updates $lasp_p$.
- When the intersection manager wants to switch from $P$ to $P'$, it freezes $P$.
- When it receives a new request, it uses $P'$ to determine its response if the arrival time is after $lasp_p$.
- Else, it uses $P$ and $P$ rejects all requests that could modify $lasp_p$. 
The Cost of Switching

Wall in Time

- During a switch between $P$ and $P'$, the mechanism insists that no vehicle can enter the intersection before $lasp_p$ unless it also exits before $lasp_p$.
- For a brief instant at time $lasp_p$, there can be no vehicles in the intersection; there is a wall (in time) that cannot be crossed.

Cost of Switching

The policy switching mechanism has no effect on delay until the time between switches approaches the time it takes to traverse the intersection.

<table>
<thead>
<tr>
<th>Period</th>
<th>Delay(s)</th>
<th>CI(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\infty$</td>
<td>2.03</td>
<td>$\pm0.01$</td>
</tr>
<tr>
<td>1h</td>
<td>2.03</td>
<td>$\pm0.01$</td>
</tr>
<tr>
<td>10m</td>
<td>2.03</td>
<td>$\pm0.01$</td>
</tr>
<tr>
<td>1m</td>
<td>2.13</td>
<td>$\pm0.01$</td>
</tr>
<tr>
<td>30s</td>
<td>2.20</td>
<td>$\pm0.01$</td>
</tr>
<tr>
<td>10s</td>
<td>4.25</td>
<td>$\pm0.1$</td>
</tr>
<tr>
<td>5s</td>
<td>5.14</td>
<td>$\pm0.07$</td>
</tr>
</tbody>
</table>
Collect information about humans indirectly, via the parameters of the requests made by autonomous vehicles.

<table>
<thead>
<tr>
<th>Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>The current policy</td>
</tr>
<tr>
<td>The rate of requests/second</td>
</tr>
<tr>
<td>The rate of cancellations/second</td>
</tr>
<tr>
<td>The rate of changes/second</td>
</tr>
<tr>
<td>The average time before the start of a reservation that requests are made</td>
</tr>
<tr>
<td>The average velocity at which autonomous vehicles expect to arrive at the intersection</td>
</tr>
<tr>
<td>The ratio of accepted reservations to total requests</td>
</tr>
</tbody>
</table>
Training Data

Method

- Generate training data by simulating over 800 one-hour episodes, half using ALL-LANES and half using SINGLELANE.
- Recorder classifier input data in sliding windows from 2.5 to 30 minutes long.
- For each episode, set the target policy for each generated instance to the policy that had the lowest average delay at the end of the episode.

Data

- Spawning rate chosen uniformly from $[0.001, 0.025]$
- Proportion of human drivers chosen uniformly from $[0, 0.25]$. 
  - Above 25% humans, all but the lightest traffic scenarios favor ALL-LANES.
Choosing Classifier

Method

- Test many different classifiers using the WEKA machine learning software.
- Evaluate each classifier on each sliding window size with 10-fold cross-validation.

Results

<table>
<thead>
<tr>
<th>Window</th>
<th>Const.</th>
<th>AdaB.</th>
<th>J48</th>
<th>JRip</th>
<th>N.N.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5 min.</td>
<td>66.94</td>
<td>69.03</td>
<td>78.94</td>
<td>79.21</td>
<td>80.19</td>
</tr>
<tr>
<td>5 min.</td>
<td>66.95</td>
<td>71.15</td>
<td>80.33</td>
<td>81.23</td>
<td>82.19</td>
</tr>
<tr>
<td>10 min.</td>
<td>66.84</td>
<td>70.05</td>
<td>83.23</td>
<td>82.18</td>
<td>83.16</td>
</tr>
<tr>
<td>20 min.</td>
<td>65.64</td>
<td>74.10</td>
<td>81.66</td>
<td>84.44</td>
<td>84.88</td>
</tr>
<tr>
<td>30 min.</td>
<td>67.82</td>
<td>73.27</td>
<td>85.89</td>
<td>83.16</td>
<td>88.48</td>
</tr>
</tbody>
</table>

Choice: Neural network trained on the data from the 10-minute sliding windows.
3 lanes in each direction.
Maximum speed of $25m/s$.
72 randomly generated simulator traffic settings.
Spawning probability and human driver proportion were chosen from intervals $[0.001, 0.025]$ and $[0, 0.25]$ respectively.
Each trial last 72 hours and the same traffic configuration sequence was used for each trial.
## Average Delay during 72-hour simulated period

<table>
<thead>
<tr>
<th>Policy</th>
<th>Delay(s)</th>
<th>CI(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALL-LANES</strong></td>
<td>57.70</td>
<td>±0.43</td>
</tr>
<tr>
<td><strong>SINGLE-LANE</strong></td>
<td>48.30</td>
<td>±0.40</td>
</tr>
<tr>
<td><strong>Switching</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20m</td>
<td>43.28</td>
<td>±0.51</td>
</tr>
<tr>
<td>10m</td>
<td>41.77</td>
<td>±0.46</td>
</tr>
<tr>
<td>5m</td>
<td>41.53</td>
<td>±0.33</td>
</tr>
<tr>
<td>1m</td>
<td>41.45</td>
<td>±0.66</td>
</tr>
<tr>
<td>30s</td>
<td>41.05</td>
<td>±0.42</td>
</tr>
<tr>
<td><strong>Omniscient</strong></td>
<td>37.50</td>
<td>±0.45</td>
</tr>
</tbody>
</table>
Results

Classifier reaction on traffic changes

10-minute Sliding Window Delays for the 10-minute Switching Policy

- All-Lanes
- Single-Lane
- Switching (10m)

switch 1

switch 2

time (h)

delay (s)
Sommaire

1. Introduction
   - Problem of Congestion
   - Motivation, approach and solution

2. Reservation-Based Intersection Control Mechanism
   - Original System
   - Improved System with Communication Protocol and Algorithm
   - Human-Usable Control Policies

3. Learning Policy Selection
   - Goal: adapt to traffic condition
   - Smoothly Switching Between Two Policies
   - Classifier
   - Experimental Results

4. Conclusion
You are not jam in traffic, you ARE the traffic.