Generalizing Assured AI for Traffic Light Control

Daniel Stambler and Evan Leung
Assuring AI Systems

- AI systems becoming ubiquitous
- Cannot be used in critical systems
  - Need to guarantee the worst case
  - AI fails on edge cases
- Assuring AI
  - Switch to safe algorithm to handle situations when AI fails
AI Traffic Light Controllers

- Potential for more efficient travel through intersections
- Certain errors considered unacceptable
  - Unreasonably long wait times
- Blackbox Monitor
- Whitebox Monitor
Outline

● Motivation
● Background
● Previous Work
● Problem Definition
● Approaches and Results
● Future Work
Definitions

● Model:
  ○ Differentiable mathematical formula for fitting input data

● Reinforcement Learning:
  ○ Process of training an agent to act in an environment, where it receives rewards for its actions. With enough time, the agent should learn to pick the best action

● Evaluation:
  ○ Process of using a trained model. Pass in inputs, get results

● Monolithic Model:
  ○ Inflexible model. To evaluate an m x n grid, the model needs to be trained on an m x n grid

● General Model:
  ○ Flexible model that can be evaluated on any m x n grid
Definitions: Defining our Environment

- Four way intersection, bidirectional roads
- Four incoming edges
- Four outgoing edges
- Straight + right turn on green
- Separate green light for left turns
- All lights (straight + left) must switch to yellow lights
Outline

- Motivation
- Background
- **Previous Work**
- Problem Definition
- Approaches and Results
- Future Work
Previous Work

- **Monolithic 2x2 model using SUMO/FLOW**
  - Can’t scale up

- **Generalized model in Gym CityFlow**
  - Solves scaling problem
  - Couldn’t replicate results
Early Challenges

- Onboarded to Gym CityFlow
  - Didn’t see any learning

- Switched back to SUMO/FLOW
  - Replicated previous monolithic model success
  - Spent a while learning very large codebase
  - Understood Jerry’s approach and its possible flaws, brainstormed new approaches
Outline

● Motivation
● Background
● Previous Work
● **Problem Definition**
● Approaches and Results
● Future Work
Problem Definition

- Monolithic model takes too long to train for any topology larger than 2x2

- Goal - generalized model that:
  - outperforms Safe Controller, similar to Monolithic
  - can be applied to any $n \times m$ topology

- Performance measured by average speed of all cars in system
Previous Attempts at Generalized Model

● 2x2 grid where each intersection employs the same model
  ○ AI learns at each intersection
  ○ Keeps feature vector small

● Training was unsuccessful
  ○ Every intersection is a corner case
Outline

● Motivation
● Background
● Previous Work
● Problem Definition
● Approaches and Results
● Future Work
Key Terms

- **NxM training environment**: AI controller placed in the center of \( NxM \) grid of intersections, all other intersections safe or random
- **NxM evaluation environment**: trained AI controller placed at every intersection of \( NxM \) grid
- **Pertinent avg speed**: average speed over cars that enter edges connected to AI-controlled intersections
- **Grid padding**: add an extra layer of safe-controlled intersections around an \( NxM \) grid
Creating our own Tools

- Scripts to evenly distribute and limit training jobs across machines
- Scripts to manage jobs across machines
- Scripts for logging and plotting training metrics, finding best models
Approach 1

- 3x3 training environment
- Fixed controllers on outer 8 intersections
- AI controller in the center
Inputs to the Model

- **Modifying feature vector**

- **Changes to the system**
  - Change number of traffic lights for AI to update
  - Update RL actions function to manually update traffic lights that aren’t at center node

<table>
<thead>
<tr>
<th>Variable/Size</th>
<th>3 x 3 Monolithic Vals</th>
<th>Our Implementation Vals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speeds</td>
<td>216</td>
<td>24</td>
</tr>
<tr>
<td>Distance to Intersection</td>
<td>216</td>
<td>24</td>
</tr>
<tr>
<td>Edge Number</td>
<td>216</td>
<td>24</td>
</tr>
<tr>
<td>Density</td>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>Velocity Average</td>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>Last Change</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Direction</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Currently Yellow</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>723</td>
<td>91</td>
</tr>
</tbody>
</table>
Experiments

- Three runs with 3x3 grid all safe controllers to establish a baseline
- Three runs with 3x3 grid all safe controllers except for center node
- Three runs with 3x3 grid all random controllers except for center node
Safe Controller Baselines

● Need to compare our 3x3 results with 3x3 safe controller

● Three Safe Controller runs under different seeds
  ○ Seed A: 5.59 m/s
  ○ Seed B: 5.57 m/s
  ○ Seed C: 5.55 m/s

● Average across runs: 5.57 m/s

● Note: 5.39 m/s on 5x5 with all safe controllers
Safe Controller Baselines
Monolithic 2x2 Baseline Results

Best Average Speed: 6.429 m/s (49,700,000 steps)
Safe Controller w/ One AI 3x3 Results

Best Pertinent Average Speed: 6.484 m/s (47,050,000 steps) from seed B
Training Environment Results
Random Controller 3x3 Results

Best Pertinent Average Speed: 1.837 m/s (45,500,000 steps) from seed C
Evaluation Results

- Applied models to 3x3 and 5x5 evaluation environments
- Discovered models did not learn to generalize
- AIs trained on different environments and with different reported training speeds all yield ~4.3 m/s for 3x3 and 4.22 m/s for 5x5
Generalized Environment Results
Generalized Environment Results
Approach 2

- Increase AI’s observation space to “look ahead” 1 intersection
  - Work in tandem with other AIs on evaluation environment
- 5x5 training environment to avoid edge cases
Look Ahead Controller Results

Best Pertinent Average Speed: $5.51 \text{ m/s} (44,350,000 \text{ steps})$ from seed B
Training Environment Results
Look Ahead Evaluation Results

- Same results, failing to generalize/work with other AI controllers
- Average speed on 3x3 eval environment is 4.3 m/s
Evaluation Environment Results
Outline

● Motivation
● Background
● Previous Work
● Problem Definition
● Approaches and Results
● Future Work
Future Work

- Training on multiple environments in parallel
  - Allows AI to learn more scenarios
- Replace SUMO with more efficient environment
  - Connect our environments with Gym CityFlow
  - Build a new environment that can take advantage of GPU resources

Image Source: https://github.com/wu6u3/async_ppo
Questions
Thank You