Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)

Evaluating Multi-Agent Traffic Controllers

Crystal Redman **Adele Howe**
Computer Science Dept.
Colorado State University
Fort Collins, CO 80526 U.S.A.
emails: redman,howe@cs.colostate.edu

Abstract

Many multi-agent traffic control designs have been proposed; each makes different assumptions about the environment, which makes it difficult to compare their performance. In this paper, we present a testbed for flexible and consistent evaluation of multi-agent approaches to urban traffic control. The testbed, an extension of the MASON simulator, varies parameters such as traffic load, existence of arterial street and grid size. We instrument the testbed to collect a set of metrics defined in the literature: delay, normal travel time, number of stops, percent stopped, and wait time ratio. We implement three distinct, well known multi-agent traffic controllers and use the testbed to assess the impact of different scenarios by comparing agent performance. The different metrics highlight clear trade-offs between time and flow metrics, but the more challenging scenarios dilute the distinctions. The testbed supports evaluation, comparative analyses and hybridization of the approaches. We use our analyses to suggest modifications to the agents and show that the agent designs can be improved.

Introduction

With increasing congestion in many countries and global warming concerns, improving traffic efficiency through automated control is becoming a valuable application for multi-agent systems. Many approaches have been proposed; each making unique assumptions about the environment and the goals of traffic control. For example, Dresner and Stone (Dresner and Stone 2008) anticipate future automated vehicle/road systems and explore safety and efficiency issues; Balan and Luke (Balan and Luke 2006) consider whether fairness and efficiency can coexist; Bazzan et al. (Bazzan 2008; Bazzan and Klügl 2008) consider traffic control game theory and driver route planning.

The differences in the agents translate naturally to differences in performance. Yet because each agent has been implemented within different simulations, it is difficult to compare their performance quantitatively. We present a unified testbed for evaluating traffic control agent trade-offs. The testbed provides a parameterizable simulator instrumented with a set of metrics derived from the literature and supports the addition of alternative traffic control agents. Thus, one contribution of this paper is the testbed itself, which is publiclyilable at *http://www.cs.colostate.edu/ redman/traffic*.

Our testbed extends MASON (Luke et al. 2005) (**M**ulti-**A**gent **S**imulation of **N**eighborhoods (or **N**etworks)). Our current implementation includes five metrics from the traffic control and the agents literature; they are: delay (Lieberman and Rathi 1990; Balan and Luke 2006), normal travel time (Lieberman and Rathi 1990; Bazzan 2005), number of stops (Bazzan 2005), percent stopped (Wiering et al. 14 17 June 2004), and wait time ratio (Lieberman and Rathi 1990; Balan and Luke 2006). The current agents represent four diverse designs: *Evolutionary* game theory (Bazzan 2005), *History* thresholding (Balan and Luke 2006), *Reservation* allocation (Dresner and Stone 2005) and a static baseline agent. The simulation uses a simple, but flexible model of traffic that is straightforward to interpret and can accommodate different capabilities required by the different agents.

A second contribution of this paper is that we used the testbed to explore three questions about agent performance in traffic simulations: Do the evaluation metrics differentially distinguish performance? Do the environment settings differentially influence performance on the metrics? Can new designs be motivated by comparisons of performance metrics? We find some overlap between the metrics currently defined and a trade-off between two categories of metrics (timing and flow). We also find an interaction effect with environment scenarios, especially in the more demanding scenarios. Finally, although interactions between the agent, the environment and the metrics are complex, we discover that it is possible to improve performance on one metric (although not necessarily the one being targeted), and not always to a commensurate detriment of other metrics.

Traffic Control Testbed

To design a realistic traffic simulator for agent evaluation, we relied on (Lieberman and Rathi 1990) which summarizes the dynamics of traffic simulators. The minimum environment features for traffic control simulation are:

Arterial if and where there is a street with higher traffic
Block size distance between intersections
Grid size number of intersections
Load number of vehicles
Speed average speed of vehicles

Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Speed variation variance of vehicle speed

Time to measure cycle lengths, travel times, total time, etc.

Our testbed parametrizes these features to support a microscopic simulation like in (Bazzan 2005; Burmeister, Doormann, and Matylis 1997). Arterial streets influence the distribution of vehicles within the simulation; placement of new vehicles is determined by a Gaussian distribution centered on the arterial. A Gaussian distribution, centered on the speed, determines the speed of each vehicle. More challenging scenarios use large grid size and high load to simulate congestion in a metropolitan area.

Vehicles in our testbed move based on a squares per second calculation. Each square in our grid representation is a seven by seven feet area. Vehicles make path decisions at intersections. If they are moving toward a specific destination, they will be moving in an overall direction and will choose randomly if options lead equally toward their goal (e.g., if heading due east, a vehicle may go straight or turn right if its destination is southeast). It is impossible for a vehicle to occupy a space that is already occupied by another vehicle. Therefore, we do not model collisions, and assume vehicles will only slow down while traveling behind slower traffic.

MASON Framework

MASON (Luke et al. 2005) is a simulation framework, written in Java, designed to evaluate multi-agent systems. The two most important components of the MASON framework are the `SimState` and `Steppable` classes which can be specialized to specific domains. Our `SimState` component is the interface to our testbed where variables can be changed to create various scenarios. Our `Steppable` classes include the vehicles and the traffic light controllers. Swapping alternate agent designs requires instantiation of different versions of controllers and sometimes vehicles.

Balan et al. (Balan and Luke 2006) used MASON for their History agent, but have not made the traffic control code available. Our design differs from theirs in that it explicitly models vehicle movement, but does not model acceleration and does support multiple agent designs.

Traffic Control Evaluation Metrics

The metrics are calculated as summaries of low level instrumentation in the simulation for each intersection and across a simulation run. Most traffic control metrics fall into two categories. *Timing metrics* summarize time across all vehicles as they move along their paths through the grid:

Delay (D) sums the difference between the actual travel time and the optimal travel time (assuming no stopping due to traffic lights or slow traffic).

Normal Travel Time (NTT) computes the average actual travel time normalized by travel distance.

Wait Time Ratio (WTR) divides cumulative time the vehicles are stopped by the cumulative actual travel time.

Flow metrics indicate vehicle throughput:

Number of Stops (NS) sums the total number of vehicles stopped.

Percent Stopped (PS) divides the total number of stopped vehicles by the total number of vehicles.

Traffic Control Agents

Each intersection is controlled by an agent. We implemented four traffic control agents based on specific publications[1].

Evolutionary Agent

The *Evolutionary Agent* (Bazzan 2005) uses evolutionary game theory. Each agent selects an action and receives a gain or loss due to the action. The actions change the phase lengths of the traffic lights the agent is managing. Gains (or losses) are calculated at discrete time intervals triggered when performance is worse than neighboring controllers. Traffic controllers that have more free flowing traffic (compared to neighbors) receive gains. Performance is measured as travel time and queue size (the number of vehicles stopped in the queue of a traffic controller).

Every few time steps, an *Evolutionary Agent* checks its performance and queries neighbor controllers for their performance. The agent then compares itself to neighbors and decides whether to enter a learning phase. Top ranking neighbors are involved in a cross over operation where their phase and cycle lengths are combined and with some probability, lower ranking neighbors can receive this timing schedule or a new mutated schedule. It was designed to control an arterial street.

History Agent

The *History Agent* (Balan and Luke 2006), uses a credit system: awarding green lights to lanes containing high wait time vehicles. Vehicles stopped at red lights receive credit and spend credit when passing through a green light. Phase and cycle lengths are subject to change every 20 time steps.

Each vehicle starts with no credit; at green lights, credits are dispersed to all other vehicles in the system that are stopped at red lights. The total credits in the system always sums to zero; negative credits are allowed. The *History Agent* is designed to reduce the variance in mean wait time (WTR). The authors tested various loads between 31 and 16K vehicles on four by four and ten by ten grid sizes. The grid, as in our testbed, is an orthogonal mesh with square blocks and sensors with a 0.125 mile range. Their vehicles traveled at exactly 25 mph.

Reservation Agent

(Dresner and Stone 2005) designed traffic agents that utilize cruise control, GPS, and auto-steering. Vehicles request and receive space and time slots from intersection controllers.

The authors define many message types that can be passed between the two agent types. Vehicles request space in an intersection or cancel a reservation. A controller grants or cancels reservations. Both controllers and vehicles send acknowledgments. The authors use a detailed simulation of a single intersection to show that delay can be reduced without sacrificing safety (Dresner and Stone 2008). Total trip

[1] Some of the publications used are not the most recent from the authors. We used the earlier descriptions due to the timing of our implementation and/or because the earlier designs were simpler to understand and validate.

Table 1: The 12 scenarios used in our study.

Arterial	Grid/Load	Grid/Load	Grid/Load
Off	10x10/50	10x10/100	10x10/200
Off	20x20/100	20x20/200	20x20/400
On	10x10/50	10x10/100	10x10/200
On	20x20/100	20x20/200	20x20/400

time was examined, but safety and efficiency were primary targets.

Baseline Agent

The *Baseline Agent* has fixed phase lengths which are set to 200 time steps (about three minutes). The controller at each intersection randomly chooses initial light states making the north/south lights green and the east/west lights red or vice versa.

Evaluating Agents with the Testbed

We use the testbed to evaluate agent performance as influenced by metric, scenario, and agent design combinations.

Experiment Design

Our scenarios use three **environment settings**:

1. *grid size*: small (GS) or large (GL)
2. *load*: low (LL), medium (LM), or high (LH)
3. *arterial*: arterial is on (AN) or arterial is off (AF)

Table 1 summarizes the exact settings for each of these twelve scenarios. The total simulated time is 5000 time steps; the time between data collection visits is 79 time steps. We set the speed limit to 35 *miles per hour*. To model variation in speed, we made one standard deviation in speed equal to ten percent of the speed or 3.5 *miles per hour*.

Our **dependent variables** are the evaluation metrics. For 30 trials of each agent type and scenario combination, we collect 63 timed samples for each of the evaluation metrics: a total of 362,880 samples.

Evaluation Metrics and Agent Performance

First, we consider whether the different metrics actually distinguish performance by examining agreement in effect (same ranking of agents across the metrics). We rank the four agents in each scenario by their mean performance on each metric. To confirm the rankings, we used Wilcoxon Signed-Rank tests[2] to make pair-wise comparisons of rankings for all treatment groups. The null hypotheses were generated by examining the metrics' means across all scenarios. For the timing metrics, the rankings were: Reservation, History, Evolutionary and Baseline; for the flow metric, the rankings were reversed. We find that the agents follow these rankings ($p < 0.0001$) for 39 out of 48 combinations of metric and scenario.

[2] We use this non-parametric alternative to the t-test to compare agents because our data do not display equality of variance and in some cases appear non-normal.

Table 2: Ranking changes for the nine combinations out of 48 in which the agent rankings deviated from expectations.

Combination	Deviation
D, GL, AN, LH	B and E before H
NTT, GS, AN, {LL, LM, LH}	B before E
NTT, GL, AN, LL	B before E
NTT, GL, AN, LM	B before E
NTT, GL, AN, LH	B before E and H E before H
PS, GL, AN, LH	E and B tie H before B H before E
WTR, GL, AN, LH	B before H E before H

Of the nine combinations that deviate from the expected (see Table 2), only one with PS does not invert the time metric rankings. Thus, we observe a performance trade-off between the flow metric and the time metrics. Generally the three timing metrics agree.

Figures 1 and 2 show mean performance of each agent in each scenario on D versus NS and D versus PS, respectively. Specific agent designs tend to cluster together. However, the History, Evolutionary and Baseline agents each have points in the upper middle with NS and upper left with PS which correspond to the most challenging scenario [GL, AN, LH]. The two plots differ especially in the Reservation Agent, which performs relatively well on NS, but not PS.

Influence of Scenario?

Performance is a function of scenario. To assess the strength of each scenario's effect, we ran a four factor analysis of variance (ANOVA) to examine the interaction among load, arterial, grid size, and agent type. We found that each factor (and all combinations of factor interactions) had a significant impact ($p < 0.0001$).

Since every factor is significant, we looked at the sum of squares and found that agent type has the greatest impact, indicating significant differences in the scenarios they handle well.

The presence of the arterial street appears most often in the rankings that deviate from the hypothesized rankings. Every combination in Table 2 has the arterial on. The arterial setting presents a distinct challenge for the designs.

Baseline Agent and *Evolutionary Agent* have nearly the same environment setting sensitivities and appear to be more impacted by the arterial setting than the *History Agent* or *Reservation Agent*. The interaction between arterial and load setting is important for all agent types. The *History Agent* and *Reservation Agent* are more sensitive to the load setting, and the *Baseline* and *Evolutionary* agents are more sensitive to the grid setting.

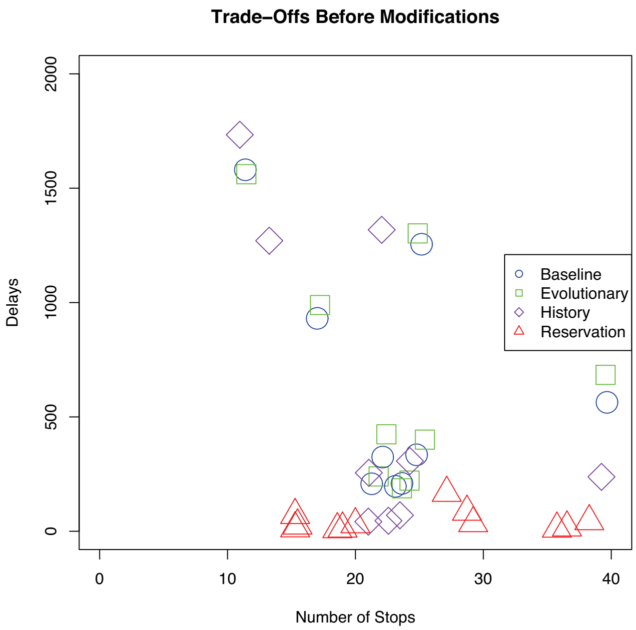

Figure 1: Mean performance on D and NS metrics for each agent in the 12 scenarios.

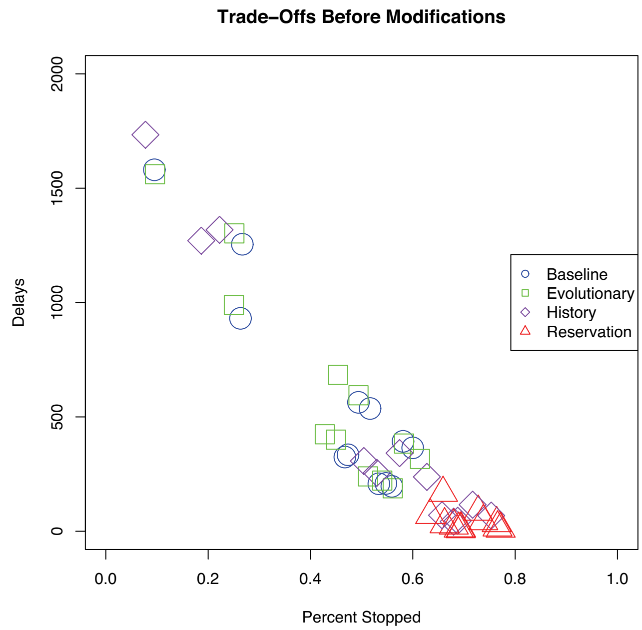

Figure 2: Mean performance on D and PS metrics for each agent in the 12 scenarios.

Redesigning the Agents

Based on the trade-offs that we observed, we identified opportunities for improvements in the current agent designs.

Reservation Agent performed relatively poorly on PS. We hypothesize that PS may be improved by using a reservation priority based on the previous time spent stopped.

History Agent excelled on PS (54% drop over same scenario with AF) for the most challenging scenario [GL, AN, LH] at the marked expense of the timing metrics (e.g., 3.5 times higher delay). We conjecture that it might starve the side streets causing the longest possible waits there, which might be mitigated by having the agent monitor maximum credits and offer priority for higher maximums than their neighbors.

Evolutionary Agent had difficulties with the arterial scenarios. We conjecture that this could be due to controllers along the arterial passing their behavior to controllers with very different traffic patterns; those along non-arterial side streets. Breeding should only commence between controllers with similar traffic patterns. Additionally, breeding and mutation frequency should increase when performance is poor.

By investigating these changes, we show that by comparing performance in the same testbed, we can merge features of the different agents to improve performance.

Specifics of Redesigns

Reservation Agent When a vehicle enters the domain of a controller, it sends a message to check in. The controller keeps track of two data structures organized by stopped time: north/south vehicles within range and east/west vehicles within range. If a vehicle is alone at an intersection, a green is automatically granted. If there are vehicles in both the north/south and east/west structures, then the controller examines the single most stopped north/south vehicle and the single most stopped east/west vehicle. The green is granted to the direction (or maintained in the direction) which has a vehicle within range with the greatest stopped time. When a vehicle exits the intersection, it sends a message to check out and is removed from the appropriate data structure. Vehicles with a greater stopped time have likely been stopped more frequently thus this change introduces the notion of fairness.

History Agent If the ‘stopped credits’ exceeds the ‘moving credits’, then the green is granted to the opposite direction in that intersection. We added a provision so that if any one vehicle is stopped more than 400 time steps (approximately seven minutes), that lane automatically gets a green.

Evolutionary Agent Breeding happens between controllers with similar traffic patterns, and mutation occurs whenever the number of stopped vehicles exceeds the number of moving vehicles. During global evolutions, the agent compares at four values against four neighbors: north and south vehicles stopped, north and south vehicles moving, east and west vehicles stopped, and east and west vehicles moving. Using the current and neighboring controllers’ number of stopped and moving vehicles, the standard deviations of the number of stopped and moving vehicles are

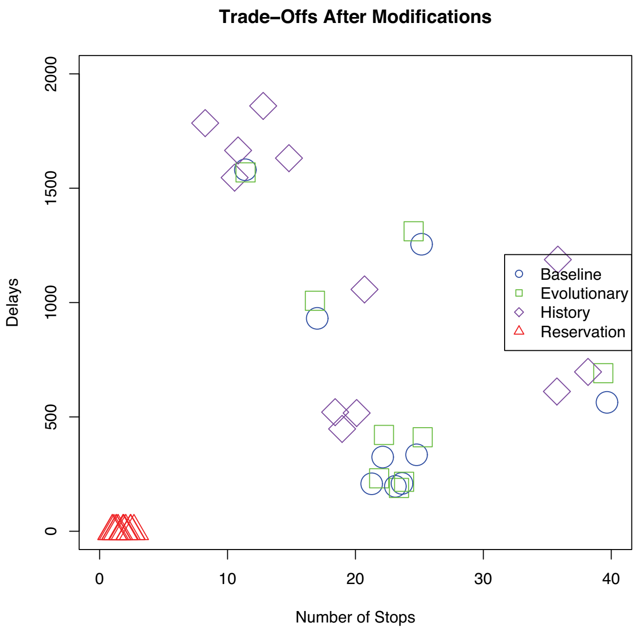

Figure 3: Mean NS and D for the modified agents in the 12 scenarios. Note the difference in D range from Figure 1.

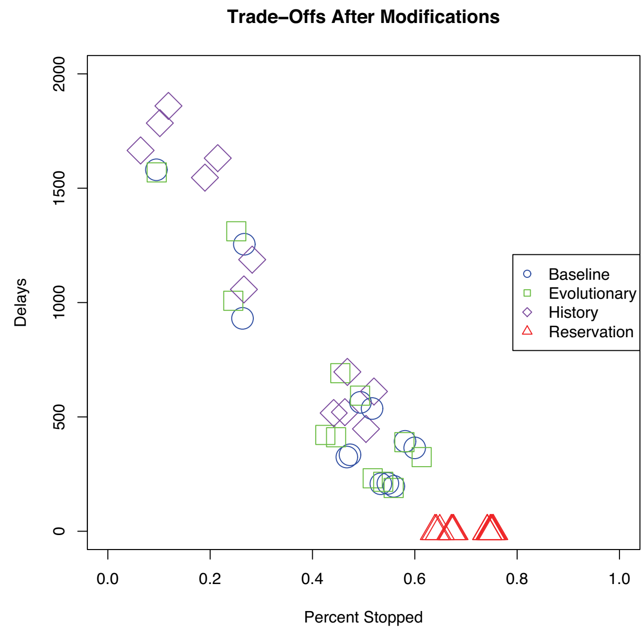

Figure 4: Mean PS and D for the modified agents in the 12 scenarios. Note the difference in D range from Figure 2.

calculated. Controllers are not allowed to breed unless the current number of vehicles stopped and moving (in both the east/west and north/south directions) are within one standard deviation of the standard deviation calculated for the controller and its neighbors.

Performance of the Redesigned Agents

We ran the same scenarios as before with the new agents. We analyzed the data to address three questions. First, was the expected improvement achieved? Second, did the sensitivities to environment change? Third, how was performance across the metrics?

Improvements as Expected? We analyzed the results in two ways: quantitative and qualitative. First, for each agent/scenario combination, we calculated the percentage difference in each metric relative to the performance of the baseline agent. We then looked at how the percentage changed between the original agent and the redesigned version (see Table 3). Second, we reevaluated the agent rankings using the Wilcoxon Signed-Rank test.

The changes to the agents did not at all match our expectations (see Table 3). PS for the *Reservation Agent* improved slightly; however, NS improved dramatically. We did not obtain an improvement in timing metrics for the *History Agent* or an overall improvement in the *Evolutionary Agent* relative to the *Baseline*. Instead, the redesigns accentuated previous performance superiorities. Figures 3 and 4 shows the performance of the modified agents; the separations are more pronounced than before.

The qualitative analysis shows a general change in rankings where the *Evolutionary Agent* outperforms the *Baseline* and the *History Agent* on most of the scenarios with respect to the timing metrics. The *Evolutionary Agent* improves on the [GS,AN,LH] scenario with respect to PS. The *History Agent* improves significantly according to PS by outperforming both the *Evolutionary* and *Baseline* Agents in nearly every scenario, but at the cost of the timing metrics.

The *History Agent*, especially along the arterial, might stop many vehicles to allow perhaps just one vehicle to go that has been stopped a long while. The combined force of the arterial vehicles' stopped time has worsened the timing metrics for this reason.

Table 3: Mean across scenarios change in percentage difference from *Baseline* for each agent and metric. Negative numbers indicate an improvement over prior design. NTT was too coarse to show much difference.

		Metrics		
		PS	D	WTR
Agents	R	2.0	-6.1	-12.7
	H	-31.7	149.9	72.3
	E	0.7	2.0	1.5

Same Environment Impact? We ran three factor ANOVAS to analyze the impact of the scenarios on each modified agent design. For the *Evolutionary Agent*, grid size had more effect on PS than before and the arterial setting became less important. Grid size's effect may be due to the longer simulation run.

The *History* and *Reservation Agents* had the most changes in sensitivities. For the *History Agent*, arterial setting replaced load as most important and performance became more spread out. For the *Reservation Agent*, the grid setting replaced load as most important and the arterial diminished in its influence. Because the controller no longer makes decisions based on individual vehicles, it makes sense that load no longer has such a strong impact.

Overall Performance? Although the changes did not meet our expectations, they underline what are likely traffic control trade-offs. For the *Reservation* and *History* agents, improvements in one type of metric degraded the other; however, the amounts were not necessarily commensurate. For the *Reservation Agent*, the degradation was small relative to the improvement; for the *History Agent*, the degradation was large. These trade-offs crossed all scenarios.

However, the *Evolutionary Agent* showed more variability, relative to *Baseline*, due to environment setting. For some of the scenarios, it improved in either timing (6/12 for D and 3/12 for WTR) or flow metrics (2/12 for PS); in one case [GS,AN,LH], it improved both.

Future Work and Conclusion

Our current study shows the utility of consistent comparison of traffic control agents: performance can be characterized with respect to particular scenarios and strengths and weaknesses, then the analysis can spark new designs.

In the future, we will extend the capabilities of the simulator as well as further explore the trade-offs in assumptions and performance. In particular, we envision three changes to the simulator. First, the set of metrics should be expanded. For example, increases in fuel prices and concern about greenhouse gases motivate metrics of cumulative fuel consumption and vehicle emissions; both require better vehicle models. Second, the drivers should be made more goal oriented; route planning and sink/source distributions should be modeled. (Bazzan and Klügl 2008) has examined the impact of route choices and interaction with traffic control; their designs may be beneficially included. Third, the traffic distributions and model should be made more realistic. We will improve the model of acceleration/deceleration and obtain real grid pattern data to seed the simulation.

The second category of change is the agents. Newer versions of some of the agents have been developed and will be incorporated. For example, as in Dresner and Stone (Dresner and Stone 2008), we should allow for the vehicles to self-organize. In addition, adding more data retention (collective and historical memory) may facilitate moving off the trade-offs we currently observe. For example, if agents retain data such as vehicle trip length, route, and credits then information such as expensive routes can be uncovered. Vehicles can earn a bonus by traveling on less expensive routes.

As a contribution of this research, we offer three lessons.

- Each agent design represents trade-offs between timing and flow metrics as well as between capabilities of agents and controllers. No one is best across all metrics, but each appears to occupy different points on a Pareto frontier.
- The complexity of the interactions between agent implementations and this environment can produce effects to performance that are hard to predict (as in the impact of some of the agents changes).
- A unified, parameterizable testbed allows one to compare potential designs and better understand effect of scenario on different performance metrics.

The future of technology will have an important impact on traffic control. Vehicular automation becomes ever prevalent as advances in computer vision and automated cruise control are made. Sensor sensitivity, accuracy, and standardization will improve. We see vast improvements in wireless network fidelity, security, and cost effectiveness. The fact that technology moves forward so quickly emphasizes the need to compare traffic control agent designs carefully; considering a wide variety of conditions and admitting the availability of technologies currently on the horizon or just beyond (as in (Dresner and Stone 2008)).

References

Balan, G., and Luke, S. 2006. History-based traffic control. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 616–621. New York, NY, USA: ACM Press.

Bazzan, A. L., and Klügl, F. 2008. Re-routing agents in an abstract traffic scenario. In *SBIA '08: Proceedings of the 19th Brazilian Symposium on Artificial Intelligence*, 63–72. Berlin, Heidelberg: Springer-Verlag.

Bazzan, A. L. 2005. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multi-Agent Systems* 10(1):131–164.

Bazzan, A. 2008. To adapt or not to adapt consequences of adapting driver and traffic light agents. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, 1–14. Springer Berlin Heidelberg.

Burmeister, B.; Doormann, J.; and Matylis, G. 1997. Agent-oriented traffic simulation. *Trans. Soc. Comput. Simul. Int.* 14(2):79–86.

Dresner, K., and Stone, P. 2005. Multiagent traffic management: an improved intersection control mechanism. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 471–477. New York, NY, USA: ACM Press.

Dresner, K., and Stone, P. 2008. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research* 31(1):591–656.

Lieberman, E., and Rathi, A. K. 1990. Traffic simulation. In *Journal of Transportation Engineering*, volume 116, 734–743.

Luke, S.; Cioffi-Revilla, C.; Panait, L.; Sullivan, K.; and Balan, G. 2005. Mason: A multiagent simulation environment. *Simulation* 81(7):517–527.

Wiering, M.; Vreeken, J.; van Veenen, J.; and Koopman, A. 14-17 June 2004. Simulation and optimization of traffic in a city. *IEEE Intelligent Vehicles Symposium* 453–458.