In *AAMAS 2007 Workshop on Adaptive and Learning Agents* (ALAg 2007), pp. 34-39, Honolulu, HI, USA, May 2007.

Learning Policy Selection for Autonomous Intersection Management

Kurt Dresner and Peter Stone
University of Texas at Austin
Department of Computer Sciences
Austin, TX 78712 USA
{kdresner, pstone}@cs.utexas.edu

ABSTRACT

Few aspects of modern life inflict as high a cost on society as traffic congestion and automobile accidents. Current work in AI and Intelligent Transportation Systems aims to replace human drivers with autonomous vehicles capable of safely and efficiently navigating through the most hazardous city streets. Once such vehicles are common, interactions between multiple vehicles will be possible. Traffic lights and stop signs, which were designed for human drivers, may no longer be the best method for intersection control.

Previously, we made the case for a reservation-based intersection control mechanism designed for autonomous vehicles, but compatible with human drivers. Including human drivers allows incremental deployability as well as support for those who drive for pleasure, but may result in significantly suboptimal performance, as human drivers may be present in dramatically varying proportions. In this paper, we develop a learning-based approach to determine which variant of the control mechanism will be most effective under given conditions, and then combine the resulting predictor with our multiagent intersection management mechanism, enabling it to determine when and how it should alter its configuration to best suit the current traffic conditions. Our extension is fully implemented and tested in simulation, and we provide experimental results demonstrating its efficacy.

1. INTRODUCTION

With the average American wasting 46 hours per year in traffic and accidents sapping upwards of $230 billion from the US economy annually, few activities take as high a financial or emotional toll on people as automobile travel [7, 9]. Intelligent Transportation Systems (ITS) is the field that focuses on integrating information technology with vehicles and transportation infrastructure to make travel safer and more efficient. Current work in AI and ITS aims to replace error-prone human drivers with autonomous vehicles capable of safely and efficiently navigating the most hazardous and congested roadways.

Once such vehicles become common, interactions between multiple vehicles will also be possible. Traffic lights and stop signs, designed for human drivers, may no longer be the best method for intersection control. We recently proposed a multiagent intersection control mechanism for autonomous vehicles [3, 5]. This 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. While alluding to a mechanism for altering the configuration of the system online, we did not fully specify, implement, or experiment with it, nor did we provide any method for choosing an appropriate configuration given the current traffic conditions. Even if such a method were to exist, measuring the current proportion of human drivers at an intersection would necessitate expensive infrastructure beyond that already required by the intersection control mechanism.

In this paper, we make three main contributions. First, we fully specify a configuration switching mechanism, implement it in simulation, and analyze its performance. Second, we demonstrate that a classifier trained on data from the multiagent communication protocol can select the most appropriate configuration for the current traffic conditions without additional protocol or infrastructure requirements. Third, we integrate this classifier into our switching mechanism, producing a fully-implemented system that smoothly and efficiently alters its configuration to suit changing traffic patterns. Despite the lack of specialized sensory equipment, the performance of our system approaches that of an omniscient agent able to select the optimal configuration based on knowledge of the upcoming traffic conditions.

2. RESERVATION-BASED INTERSECTION CONTROL

In our 2004 paper, we make the case for a new type of intersection control mechanism [3]. This mechanism, instead of communicating with human drivers through lights, communicates directly with the *driver agents* piloting autonomous vehicles. Driver agents "call ahead" to an agent stationed at the intersection, called an *intersection manager*, to reserve a region of space-time in the intersection. As part of the request, the driver agents include information about the physical qualities and capabilities of the vehicle, as well as a 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. 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. In scenarios comprising only autonomous vehicles, such a system can vastly outperform current intersection control mechanisms like stop signs and traffic lights. Vehicles using such a system on average experience much lower *delay*, which is increase in the time it takes for them to reach their destinations due to the presence of the intersection. In recent work, we have extended this mechanism to incorporate human drivers [5]. In the remainder of this section, we briefly review our relevant previous results.

2.1 FCFS

The intersection control policy is the central part of the intersection manager, and thus a crucial component of the whole system. Our first policy operates on a "first come, first served" basis, earning it the name "FCFS". Much more efficient than current intersection control mechanisms like traffic lights, FCFS enables the intersection manager to interleave traffic from all directions simultaneously, orchestrating "close calls" which human drivers would not be capable of performing. In situations involving only autonomous vehicles, FCFS can reduce average delay by as many as two orders of magnitude compared to traditional intersection control mechanisms [4]. To accomplish this, FCFS divides the intersection into a grid of $n \times n$ *reservation tiles*, where n is the *granularity* of the policy. When determining whether to grant a reservation request, FCFS simulates the trajectory of the requesting vehicle according to the parameters of the request. Throughout the simulation, FCFS determines which tiles will be occupied by the vehicle. If the vehicle does not occupy any reserved tiles, FCFS grants the reservation and reserves the occupied tiles for the appropriate times. Otherwise, FCFS rejects the request.

2.2 FCFS-Light

To accommodate humans, the FCFS-LIGHT policy must be able to communicate to them, and it does so using the existing intersection infrastructure (traffic lights). FCFS-LIGHT associates a set of reservation tiles with each light, called *off-limits tiles*. For the purposes of granting reservations, FCFS-LIGHT treats these tiles as "reserved" whenever the associated light is green, yellow, or has very recently turned red. The reservation process for FCFS-LIGHT identical to that of FCFS, except that vehicles approaching a green light are not restricted by the off-limits tiles for that light.

The part of the policy that controls the lights is called the policy's *light model*. In addition to manipulating the physical lights, the light model is responsible for providing the policy with information about the current and future state of the lights. Here we describe two light models: ALL-LANES and SINGLE-LANE.

2.2.1 ALL-LANES

The ALL-LANES light model is very similar to modern-day traffic lights. In this model, all lanes in each direction are turned green simultaneously, one direction after another in a cycle. ALL-LANES is particularly well-suited to conditions with significant portions of human drivers. Figure 1 shows each phase ALL-LANES. For the remainder of this paper, if we refer to ALL-LANES as a policy, we mean FCFS-LIGHT using the ALL-LANES light model.

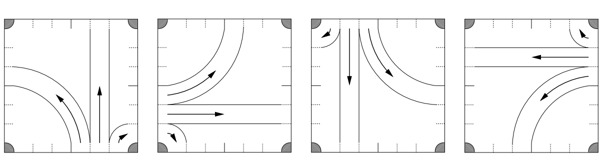


Figure 1: The ALL-LANES light model. Each direction is given all green lights in a cycle: north, east, west, south.

2.2.2 SINGLE-LANE

As opposed to ALL-LANES, the SINGLE-LANE light model is much more like standard FCFS than a traffic light. Instead of turning all lights in each direction green simultaneously, the lights in each lane are turned green one by one. This reduces the number of simultaneously off-limits tiles, thereby freeing up more of the intersection to be used by autonomous vehicles. As can be expected, this model performs well when the proportion of human drivers is small. Figure 2 shows the first half of the SINGLE-LANE light model. If we refer to SINGLE-LANE as a policy, we mean the FCFS-LIGHT policy using the SINGLE-LANE light model.

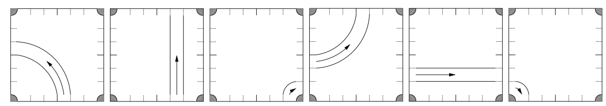


Figure 2: The first half-cycle of SINGLE-LANE. The lights in each individual lane are turned green in succession.

3. POLICY SWITCHING

Because policies using different light models perform differently under various traffic conditions, it would be useful to have an intersection manager that can switch policies, without having to bring the whole system to a halt. In this section, we fully specify a method for switching smoothly between policies, which we implement and empirically validate.

3.1 Smoothly Switching Between Two Policies

The simplest way for an intersection manager to switch between two intersection control policies is to turn all the lights red and refuse all reservation requests until the intersection is empty, at which point the manager could resume with the new intersection control policy. This naïve approach ignores the ability of the vehicles and intersection manager to plan ahead and schedule around the switchover. Our more efficient solution places only a small additional requirement on intersection control policies: each policy P must keep track of the latest time $last_P$ for which any vehicle could be in the intersection. Initially, $last_P$ is the current time, and P ensures that $last_P$ is always at least as late as the current time. Whenever P grants a reservation, it updates $last_P$. If P's light model ever turns a light green, it updates $last_P$ such that $last_P$ is after the time a vehicle using that light could leave the intersection.

When the intersection manager decides to switch from policy P to policy P', it *freezes* P. If P is frozen, it rejects any reservation requests that would cause it to modify $last_P$. When the intersection manager receives a new request, it uses P' to determines its response if the arrival time in the request is after $last_P$. Otherwise, it uses P. The transition is complete once $last_P$ has passed. Thus, the intersection manager preserves the safety properties of the mechanism as a whole, ensuring that each request is handled exclusively by either P or P', each of which is assumed to be safe.

3.2 The Cost Of Switching

The main benefit of this switching mechanism is that the intersection manager need not always choose a policy capable of handling the maximum possible proportion of human drivers. Instead, as traffic conditions change, the manager can adjust the policy to compensate, increasing efficiency during periods in which human drivers are more scarce. However, during a switch between P and P', the mechanism insists that no vehicle can enter the intersection before $last_P$ unless it also exits before $last_P$. For a brief instant at time $last_P$, there can be no vehicles in the intersection; there is a "wall" (in time) that cannot be crossed. Autonomous vehicles that would otherwise be in the intersection at $last_P$ must accelerate or decelerate such that they get a reservation which will be completed before $last_P$ or begin after $last_P$. Placing additional constraints on the vehicles could decrease the overall efficiency of the intersection, increasing delays. This would create an interesting tradeoff: switching policies could have a benefit, but it might not outweigh the cost of making the switch.

However, we determined that with the FCFS-LIGHT policy, no real tradeoff exists. FCFS-LIGHT's off-limits tiles already create many "walls" (in space) that cannot be traversed, and the addition of the constraint made by switching does not have a significant effect. To quantify the effects of switching, we ran a series of 24-hour simulations in which the intersection manager repeatedly "switched" from an FCFS-LIGHT policy with the SINGLE-LANE light model to an identical policy at regular intervals. In the experiment, we set the vehicle spawning probability to a moderate 0.01 — enough that vehicles would actually compete for passage through the intersection, but also low enough to make random congestion unlikely. The baseline time for a vehicle to complete its trip is 10 seconds — 250 meters at the speed limit of 25 m/s. By varying the time between switches from 24 hours (effectively ∞) to 5 seconds, we determined that the policy switching has no significant negative effects until the switches occur extremely frequently. At the highest frequencies, the "walls" created by the switch create compartments in space-time that are only slightly longer than the time it takes a vehicle to traverse the intersection. At this point, it becomes more difficult for the intersection manager to fit a vehicle into the available space-time. Table 1 presents the results from this experiment.

Period	Delay(s)	CI(95%)
∞	2.03	±0.01
1h	2.03	±0.01
10m	2.03	±0.01
1m	2.13	±0.01
30s	2.20	±0.01
10s	4.25	±0.1
5s	5.14	±0.07

Table 1: The policy switching mechanism has no effect on delay until the time between switches approaches the time it takes to traverse the intersection.

4. POLICY SELECTION

The two light models described earlier, ALL-LANES and SINGLE-LANE, each define a different intersection control policy when combined with FCFS-LIGHT. ALL-LANES is suited to scenarios involving many humans, while SINGLE-LANE is better for scenarios in which humans are scarce. Determining which policy to use should thus be as simple as determining how many of the vehicles using the intersection are not autonomous. Unfortunately, this would involve additional expensive infrastructure, either sensors at the intersection or signaling devices on the human-driven vehicles. The humans drivers may not even be willing to place such signaling devices on their vehicles due to privacy concerns.

Instead, we base our choices on the information already available to the intersection manager via the reservation requests made by the autonomous vehicles. If an autonomous vehicle is stuck behind a human vehicle waiting at a red light, the parameters of the autonomous vehicle's next reservation request will change. It may even be forced to cancel. One altered message may not contain much information, but the intersection manager communicates with many vehicles. By maintaining a sliding window of statistics from these messages, we can gather enough information about the current state of traffic such that a trained classifier can select the most appropriate policy. Our first instinct was to use a regression learner to estimate the average delay under the various candidate policies, allowing the intersection manager to choose the policy with the lowest estimate, but the regression learner proved unreliable. Instead, we learn the choice the intersection manager must make: which policy to use.

4.1 Classifier Inputs

Obtaining information about the human-driven vehicles is nontrivial. To gather information directly, either all human-driven vehicles must be retrofitted with signaling devices, or elaborate sensing mechanisms would need to be installed at all intersections. Both options are expensive and difficult. Instead, we collect information about humans indirectly, via the parameters of the requests made by autonomous vehicles. None of the parameters give any explicit information about human vehicles, nor is it immediately obvious how one could extract such information from these parameters. Furthermore, these statistics are gathered from information that the autonomous vehicles already need to provide to the intersection manager — no additional responsibility is placed on the autonomous vehicles.

The classifier has 7 inputs:

- The current policy
- The rate (requests/second) at which the intersection manager is receiving reservation requests
- The rate (cancellations/second) at which the intersection manager is receiving reservation cancellations
- The rate (changes/second) at which the intersection manager is receiving reservation change requests
- The average time before the start of a reservation that requests are made
- The average velocity at which autonomous vehicles expect to arrive at the intersection
- The ratio of accepted reservations to total requests

4.2 Generating Training Data

We created a large body of training data by simulating over 800 one-hour episodes, half using ALL-LANES and half using SINGLE-LANE. Each episode included a five-minute "warm-up" period during which no data were recorded, to eliminate the effects of starting with an empty intersection. Classifier input data were then recorded in sliding windows from 2.5 to 30 minutes long. At the end of the episode, we set the target policy for each generated instance to the policy that had the lowest average delay at the end of the episode. Each episode used randomized traffic conditions, however every randomly-generated configuration was used twice — once for each policy. The spawning rate was chosen uniformly from the interval $(0.001, 0.025]$, which represents everything from very light to extremely heavy traffic. The proportion of human drivers was chosen uniformly from the interval $(0, 0.25]$. Above 25% humans, all but the lightest traffic scenarios favor ALL-LANES.

4.3 Choosing a Classifier

With this data, we tested many different classifiers using the WEKA machine learning software [10]. We evaluated each classifier on each sliding window size with 10-fold cross-validation. Table 2 presents results from four representative classifiers: JRip (rules), J48 (decision tree), AdaBoost with decision stumps, and a neural network. Each classifier used WEKA's default settings.

The neural network performed the best overall, followed closely by the J48 decision tree. All results were reported by WEKA as statistically significant (with respect to the constant classifier) with

	Classifier				
Window	Const.	AdaB.	J48	JRip	N.N.
2.5 min.	66.94	69.03	78.94	79.21	80.19
5 min.	66.95	71.15	80.33	81.23	82.19
10 min.	66.84	70.05	83.23	82.18	83.16
20 min.	65.64	74.10	81.66	81.44	84.88
30 min.	67.82	73.27	85.89	83.16	88.48

Table 2: Percentage of correctly classified instances on the training data using 10-fold cross-validation.

95% confidence. As we had initially suspected, the longer sliding windows were much less noisy, and therefore easier to learn. For the rest of the paper, unless otherwise specified, when we refer to the classifier, we mean the neural network as implemented in WEKA and trained on the data from the 10-minute sliding windows. While exploring the space of potential training data might make for an interesting optimization, it is not the main focus of this paper, and thus we fix this variable in order to study other aspects of the mechanism more closely. Because performance does not vary dramatically over the range tested, we chose a value near the middle of the range.

4.4 Putting the Classifier to Work

We combine the classifier with policy switching by maintaining a sliding window of data in the intersection manager, which the classifier uses to select a policy at pre-specified intervals. If the classifier chooses the policy already in use, no switch occurs. By integrating the trained classifier with the policy switching method, we produce an intersection manager capable of selecting a policy based on current traffic conditions, inasmuch as the traffic conditions are communicated through the reservation requests of autonomous vehicles. It is interesting to note that the classifier's target task and the simulations which generated its training data are subtly different. When generating training data, each policy was essentially in a steady state. However, in the target task the classifier must tolerate the fact that although current simulator settings may be best served with a particular policy, congestion created earlier in the experiment — perhaps the result of different simulator settings or a poor choice of policy — will affect the vehicles currently making reservation requests.

5. EXPERIMENTAL RESULTS

To evaluate the performance of the switching intersection manager, we use the same custom discrete-time simulator presented in our earlier work [5]. Time is discretized into intervals of 0.02 seconds, and traffic flow is regulated by manipulating the probability with which the simulator spawns a new vehicle during each time step as well as the probability with which the simulator spawns a human-driven vehicle. We also use identical parameters: the simulator models a 250m $\times$ 250m area, three lanes of traffic travel in each cardinal direction, vehicles are limited to a maximum speed of 25m/s, and the granularity of each policy is 24.

The test scenario comprises a series of 72 randomly generated simulator traffic settings. As with the training data generation, the spawning probability and human driver proportion were chosen uniformly at random from the intervals $(0.001, 0.025]$ and $(0, 0.25]$, respectively. Although the settings are randomly generated, we use the same sequence for each trial. Each trial lasts 72 simulated hours, with each configuration used for exactly one hour. Performance is measured by calculating the average delay over all vehicles spawned in the 72 hours. The switching managers use a sliding window to keep an average of all input values from the last time a decision was made; the size of the sliding window is equal to the time between potential switches.

5.1 A Lower Bound

After analyzing the performance of ALL-LANES and SINGLE-LANE throughout the 72-hour trial, we determined which policy worked best for each configuration and created an intersection manager that switches accordingly. We call this manager "omniscient", as it knows when to switch *a priori*. The omniscient manager is not technically optimal — it chooses the policy that performs best on each configuration, but it cannot adapt to changes in traffic conditions that result from the stochastic nature of the traffic generation.

5.2 Switch Frequency

While the configuration changes took place at regular intervals, a robust switching manager should not rely on such assumptions. By allowing switching more frequently, the intersection manager gains agility, but may pay a price in terms of stability. However, as we showed, stability is not as important as one might think — the cost for making a policy switch is negligible. Agility turns out to be much more important: not only can the intersection manager react quickly to changing conditions, but it can also switch back quickly if it chooses the wrong policy. We created five versions of the intersection manager, varying the switching period from 20 minutes to 30 seconds. Note that whenever the manager selects the policy it is already using, no switch takes place. Table 3 shows the results of running each of these five versions, as well as ALL-LANES, SINGLE-LANE, and the omniscient intersection manager.

Policy		**Delay(s)**	**CI(95%)**
ALL-LANES		57.70	$\pm$0.43
SINGLE-LANE		48.30	$\pm$0.40
Switching	20m	43.28	$\pm$0.51
	10m	41.77	$\pm$0.46
	5m	41.53	$\pm$0.33
	1m	41.45	$\pm$0.66
	30s	41.05	$\pm$0.42
Omniscient		37.50	$\pm$0.45

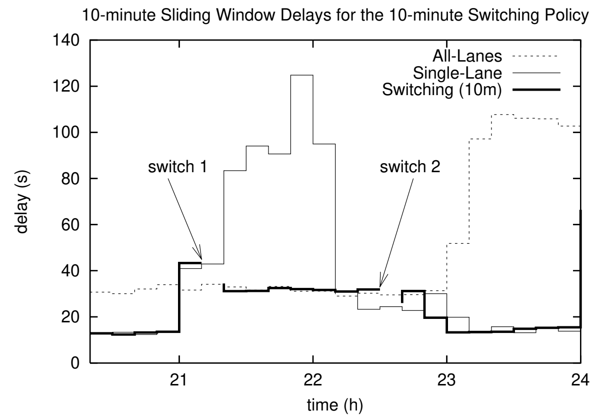
Table 3: Average delay during a 72-hour simulated period. As the intersection manager switches policies more often, it can react to changing conditions more quickly, leading to lower average delay.

Every switching manager performed significantly better than either ALL-LANES or SINGLE-LANE alone. The switching manager with the shortest period (30 seconds) delayed the average vehicle only a few seconds more than the omniscient switcher. However, with the exception of the 20-minute version, the difference in performance between the learned switchers was not statistically significant. Perhaps most importantly, re-evaluating the policy choice as frequently as every 30 seconds does not negatively affect performance — the classifier rarely recommends switches at successive 30-second decision points. These results do not indicate that SINGLE-LANE is better than ALL-LANES — it is trivial to adjust the traffic settings (specifically the proportion of human drivers) so that either policy performs better than the other. However, intelligent policy switching should always perform about as well or better than the best of the two static policies.

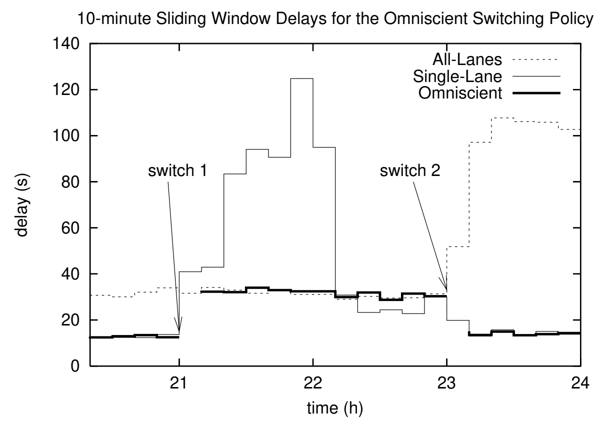
5.3 Outperforming The Omniscient Policy

The results in Table 3 show that the switching intersection manager can handle varying proportions of human drivers, even though it never directly senses or communicates with them — all information used by the classifier is readily available from the reservation requests of the autonomous vehicles. When we used the same 72 sets of simulator settings, but without any humans, the switching intersection manager behaved exactly as we expected: quickly switching to SINGLE-LANE and never going back. If the intersection manager were aware that the simulator was not spawning any human drivers, this would not be remarkable. However, the intersection manager gleans all its information about the current state of traffic from the reservation requests made by the autonomous vehicles. In some sense, the classifier-based switcher can outperform even the omniscient intersection manager at times, because the settings on the simulator do not precisely determine the actual traffic conditions — there is a lot of stochasticity.

Figure 3 shows the performance of the various policies on one representative section of the test scenario, with delay reported in 10-minute sliding windows. In 3(a), the classifier-based switcher, re-evaluating every 10 minutes, makes the switch from SINGLE-LANE to ALL-LANES as soon as it senses the change in traffic conditions. When conditions become more favorable to SINGLE-LANE, it makes a second switch back. However, the second switch is in the middle of the hour — it does not correspond to a change in the simulator settings, but rather the actual traffic conditions. In contrast, Figure 3(b) shows the performance of the omniscient intersection manager on the same scenario. Notice that it makes the first switch preemptively, before the classifier-based manager would have any chance to sense a change. The traffic parameter change at 22 hours does not change the optimal policy, so the omniscient agent stays with ALL-LANES. Because it is hard-coded to switch based on the actual simulator settings, it cannot sense the opportunity to switch later in that hour. Overall, however, the perfect prediction of the omniscient policy is more important than its inflexibility; the classifier does not always make the correct choice and this proves more significant (see Table 3).



(a) The classifier reacts to actual changes in traffic.



(b) The omniscient manager knows which policy is best overall for each configuration.

Figure 3: In 3(a), the classifier-based switcher first switches to ALL-LANES once it senses traffic conditions have changed, then switches back to SINGLE-LANE when conditions change the second time. In 3(b), the omniscient switcher knows in advance that conditions will change and preemptively switches to ALL-LANES. However, because it is not adapting online, it does not switch back to SINGLE-LANE until the traffic settings change at the end of the hour.

6. CONCLUSION

In this paper, we described how a human-compatible multiagent intersection control mechanism, combined with a learned classifier and a method for switching policies, can adapt online to varying traffic conditions. The policy-switching system presented significantly outperforms each of the previously proposed static intersection control policies. Prior to this work, we assumed that in order to efficiently handle varying numbers of human drivers, the intersection would need to be able to reliably detect their presence. A large positive implication of the results presented here is that the explicit detection of human drivers, as well as the expensive infrastructure required to do so, is unnecessary.

Aside from our own work, a large body of research addresses the problem of creating efficient, adaptive intersection control. The most popular offline traffic signal timing optimization algorithm is TRANSYT [8]. SCOOT represents an advancement over TRANSYT, in that it can adapt online [6]. More recently, Abdulhai et al. and Bull et al. have used Q-learning and Learning Classifier Systems (LCS), respectively, to create even more robust adaptive systems of traffic signals [1, 2]. However, in each of these cases, the system works only with traditional traffic signals and vehicles, and must sense the traffic conditions directly.

While this paper answers some important questions, it leaves several avenues for further inquiry. We hope to explore the effects of allowing intersection managers to choose between more policies, or create new policies on the fly. We would also like to examine the type of data with which the intersection manager makes switching decisions; perhaps using a larger sliding window of data will allow the intersection manager to choose the correct policy more reliably. Autonomous vehicles will soon be a reality. This work takes one more step toward ensuring that we can fully exploit their capabilities to create a safer and more efficient transportation system.

7. REFERENCES

[1] B. Abdulhai, R. Pringle, and G. J. Karakoulas. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3), 2003.

[2] L. Bull, J. Sha'Aban, A. Tomlinson, J. D. Addison, and B. G. Heydecker. Towards distributed adaptive control for road traffic junction signals using learning classifier systems. In L. Bull, editor, *Applications of Learning Classifier Systems*, pages 276–299. Springer, 2004.

[3] K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *The Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 530–537, New York, NY, USA, July 2004.

[4] K. Dresner and P. Stone. Multiagent traffic management: An improved intersection control mechanism. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 471–477, Utrecht, The Netherlands, July 2005.

[5] K. Dresner and P. Stone. Sharing the road: Autonomous vehicles meet human drivers. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1263–68, Hyderabad, India, January 2007.

[6] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton. SCOOT - a traffic responsive method of co-ordinating signals. Technical Report TRRL-LR-1014, Transport and Road Research Laboratory, 1981.

[7] National Highway Traffic Safety Administration. Economic impact of U.S. motor vehicle crashes reaches $230.6 billion, new NHTSA study shows. NHTSA Press Release 38-02, May 2002. `http://www.nhtsa.dot.gov`.

[8] D. I. Robertson. TRANSYT — a traffic network study tool. Technical Report TRRL-LR-253, Transport and Road Research Laboratory, 1969.

[9] Texas Transportation Institute. 2004 urban mobility report, September 2004. Accessed at `http://mobility.tamu.edu/ums` in December 2004.

[10] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.