In K. Tuyls, et al, editors, *LAMAS 2005*, Lecture Notes In Artificial Intelligence, Springer Verlag, Berlin, 2006.

Multiagent Traffic Management: Opportunities for Multiagent Learning

Kurt Dresner and Peter Stone

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

Abstract. Traffic congestion is one of the leading causes of lost productivity and decreased standard of living in urban settings. In previous work published at AAMAS, we have proposed a novel reservation-based mechanism for increasing throughput and decreasing delays at intersections [3]. In more recent work, we have provided a detailed protocol by which two different classes of agents (intersection managers and driver agents) can use this system [4]. We believe that the domain created by this mechanism and protocol presents many opportunities for multiagent learning on the parts of both classes of agents. In this paper, we identify several of these opportunities and offer a first-cut approach to each.

1 Introduction

Traffic congestion is one of the leading causes of lost productivity and decreased standard of living in urban settings. According to a recent study of 85 U.S. cities [18], annual time spent waiting in traffic has increased from 16 hours per capita to 46 hours per capita since 1982. In the same period, the annual financial cost of traffic congestion has swollen from $14 billion to more than $63 billion (in 2002 US dollars). Each year, Americans burn approximately 5.6 billion gallons of fuel while idling in heavy traffic. Recent advances in artificial intelligence suggest that autonomous vehicle navigation will be possible in the near future. Individual cars can now be equipped with features of autonomy such as cruise control, GPS-based route planning [14, 16], and autonomous steering [10, 12]. It is inevitable that before long many of the cars on the road will have such capabilities, thus opening up the possibility of autonomous interactions among multiple vehicles.

Multiagent Systems (MAS) is the subfield of AI that aims to provide both principles for construction of complex systems involving multiple agents and mechanisms for coordination of independent agents' behaviors [17]. In earlier work published at AAMAS, we have proposed a MAS-based approach to alleviating traffic congestion, specifically at intersections [4].

Current methods for enabling traffic to flow through intersections include building overpasses and installing traffic lights. However, the former is very expensive and forbids turning, while the latter can be quite inefficient, often requiring cars to remain stopped even when no cars are present on the intersecting road.

At this time, it is possible to create a small-scale system in which all cars are piloted by a central computer. Consider, for example, the task of controlling ten vehicles on an open factory floor. However, scaling such a system to handle an intersection in which a city's worth of cars might turn up would involve prohibitively expensive and inefficient communication and control infrastructure. Our goal is to maximize the efficiency of moving cars through intersections with minimal centralized infrastructure. We assume that intersections can be outfitted with a simple wireless communication system and a protocol (which we introduced in a previous paper[2]) for communicating with oncoming traffic and giving permission for cars to pass. In the system we developed, vehicles must traverse intersections according to a set of parameters agreed upon by the vehicle and the intersection manager (as they do today by obeying red and green lights), but otherwise are free to decide for themselves how to drive. Each car is an autonomous agent, and in particular need not surrender control to any centralized decision maker.

We have demonstrated that our novel reservation system dramatically outperforms systems used in common practice, including traffic lights and stop signs. We began with a model in which cars could only go straight and move at constant velocity through the intersection [3]. In our latest results, we have extended the system to allow for turns and acceleration in the intersection [4].

In all of this prior work, the behaviors of both the driver agents and the intersection control agent were all identical and fixed throughout the simulation. However, a main feature of our research has been the definition of an agent-indepedent protocol for car-intersection interaction. In particular, we expect that in general, intersections will have different traffic control algorithms (perhaps depending on the topology of the intersection and/or expected traffic flows), and that indeed each vehicle manufacturer will create proprietary vehicle control algorithms. As long as they adhere to our pre-defined protocol, there is no reason to prevent such diversity.

Once we open the possibility of varying behaviors on the part of the agents, the intersection scenario becomes, in a sense, a multiagent game, admitting for the possibility of strategic behavior on the part of the agents, and ultimately multiagent learning-based approaches.

In this paper, we identify several possible directions for extending our current model that will require such multiagent learning. For each direction, we discuss the strategic issues and propose a first approach towards multiagent learning.

The remainder of this paper is organized as follows. In Section 2, we present a list of properties we believe a multiagent intersection control mechanism should have. In Section 3 we describe the reservation-based system that we have created (in simulation) which we believe has these properties. In Sections 5 and 6 we present several opportunities for using machine learning in the intersection manager and driver agents, respectively. In Section 7, we mention other work that has been done in this area. We conclude in Section 8.

2 Desired Properties

In the process of developing our system we outlined several properties we believed should hold in order for the system to be realistic and practical.

1. The agents should only communicate information which is necessary for the system to function properly.
2. The agents should only have access to information that can be reliably obtained with current technology.
3. Communication failure (dropped messages) should not violate the system's safety properties.
4. The vehicles should be treated as individual agents, and no centralized controller should have any more control over them than necessary.
5. The system should incorporate a simple communication protocol that allows agents to know only a minimal amount about each other. As long as agents obey and understand the protocol, no extra information exchange or other interaction should be required.
6. Every vehicle should eventually make it through the intersection (i.e. no deadlocks or starvation).

Many of these properties also ensure that the system will be amenable to machine learning techniques. Specifically, the simple, reliable protocol ensures that agents are more or less self-contained — the intersection manager isn't extensively involved in the driver agent's decision making process (and vice versa). Furthermore, the requirement that every vehicle makes it through the intersection means that a machine learning algorithm in its early stages will not bring the system to a halt as a result of risky exploration.

3 The Reservation System

In our previous work, we proposed a novel reservation-based multi-agent approach to alleviating traffic, specifically at intersections. This system consisted of two types of agents: *intersection managers* and *driver agents*. Each system consists of an intersection manager for each intersection and a driver agent for each vehicle. Intersection managers are responsible for directing the vehicles through the intersection, while the driver agents are responsible for controlling the vehicles to which they are assigned. To improve the throughput and efficiency of the system, the driver agents "call ahead" to the intersection manager and request space-time in the intersection. The intersection manager then determines whether or not these requests can be met. Depending on the decision the intersection manager makes, the driver agent either records the parameters of the request (the *reservation*) and attempts to meet them, or it makes another request at a later time. We have described our implementation of a driver agent in previous papers [4, 2]. Note that our implementations of the reservation system and the driver agent are just two possibilities. As long as the agents adhere to the protocol, the system will still work. In practice, each agent could run a different algorithm or use a different heuristic to improve performance.

To determine whether or not a request can be met, the reservation manager simulates the journey of the vehicle across the intersection, which it divides into a grid of $n \times n$ tiles. The parameter n is called the *granularity* of the reservation manager. At each time step of the simulation, it determines which tiles the vehicle occupies. If throughout this simulation, no required tile is occupied by another vehicle (from a previous reservation), the manager reserves the tiles for this vehicle.

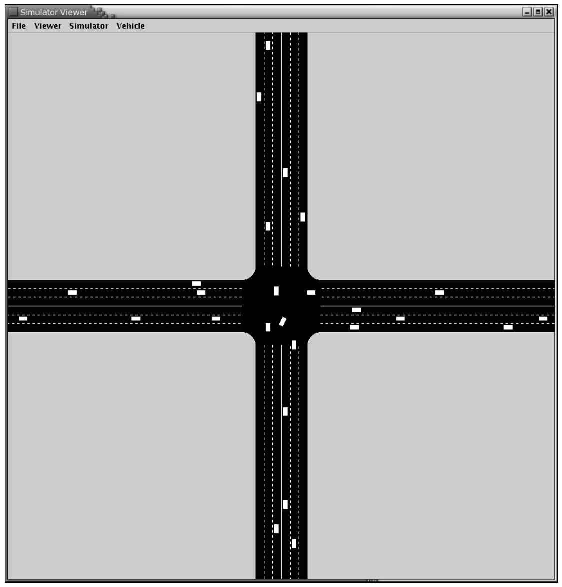

Fig. 1: A screenshot of our simulator in action.

In order to evaluate the performance of the reservation system, we created a custom simulator. A screenshot of the simulator in action can be seen in Figure 1. We tested the reservation system against two other *intersection control policies* - the overpass and the traffic light. An intersection control policy is a method the intersection managers use to determine when specific vehicles are allowed in the intersection. Using the simulator, we showed that using the reservation-based policy, vehicles crossing an intersection experience much lower *delay* (increase in travel time from the optimal) versus the traffic light. Furthermore, we showed that the reservation-based policy also drastically increases the throughput of the intersection. For any realistic intersection control policy, there exists an amount of traffic above which vehicles arrive at the intersection more frequently than they can go through the intersection. At this point, the average delay experienced by vehicles travelling through the intersection grows without bound. Compared to the traffic light, this amount of traffic is much higher for the reservation system. Videos of our most recent developments can be found at `http://www.cs.utexas.edu/users/kdresner/papers/2005aamas/`.

4 Communication Protocol

In our latest work, we added the protocol by which the agents can communicate the bare minimum of information necessary to function appropriately. The protocol consists of several message types for each kind of agent, as well as some rules governing when the messages should be sent and what sorts of guarantees accompany them. A detailed specification of the protocol including full syntax and semantics is available in our technical report [2]. We believe that this protocol will help facilitate the application of machine learning techniques to the intersection domain. Here we give a brief overview of the types of messages available to the agents using this protocol.

4.1 Vehicle → Intersection

There are four types of messages that can be sent from vehicles to the intersection.

1. REQUEST — This is the message a vehicle sends when it does not have a reservation and wishes to make one. It contains the properties of the vehicle (ID number, performance, size, etc.) as well as some properties of the proposed reservation (arrival time, arrival velocity, type of turn, arrival lane, etc.).
2. CHANGE-REQUEST — This is the message a vehicle sends when it has a reservation, but would like to switch to a different set of parameters.
3. CANCEL — This is the message a vehicle sends when it no longer desires its current reservation.
4. RESERVATION-COMPLETED — This message is used when the vehicle has completed its traversal of the intersection. This message can be used to collect statistics for each vehicle, which can be recorded in order to analyze and improve the performace of the intersection manager.

4.2 Intersection → Vehicle

There are three types of messages that can be sent from the intersection to the individual vehicles.

1. CONFIRMATION — This message is a response to a vehicle's REQUEST (or CHANGE-REQUEST) message. It can contain a counter-offer by the intersection. 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. Note that this is safe to faulty communication — the worst that can happen is that the intersection reserves space that does not get used.
2. REJECTION — By sending this message, an intersection can inform a vehicle that the parameters sent in the latest REQUEST (or CHANGE-REQUEST) were not acceptable, and that the intersection either could not or did not want to make a counter-offer. This message also contains a field indicating whether or not the rejection was because the reservation manager requires the vehicle to stop at the intersection before entering. This lets the driver agent know that it should not attempt any more reservations until it reaches the intersection.
3. ACKNOWLEDGMENT — This message acknowledges the receipt of a CANCEL or RESERVATION-COMPLETED message.

5 Learning Opportunities For The Intersection Manager

At this point in the paper we have described the current state of our implementation, describing mainly the aspects required to motivate the multiagent learning opportunities we see in the future. We now turn our attention to those opportunities. Our goal at the outset of this project was to improve the efficiency of intersections. It seems natural, then, to start with the agent controlling which vehicles have access to the intersection: the intersection manager.

5.1 Delayed Response

Incorporating any nontrivial learning into the intersection manager may require a few conceptual changes to the intersection manager. As it stands, all intersection managers in the system respond immediately to requests made by vehicles. Given this constraint, the current reservation system performs as well as it can — it can't tell what is going to happen in the future. However, if we relax this constraint and allow the reservation manager to respond to requests at a later time, the intersection manager would have time to get a feel for the competing requests and can make a more well-informed decision.

This modification suggests a straightforward method for determining whether or not to grant reservations. When the intersection manager receives a request, it can calculate the last possible point at which it can respond without forcing the sending vehicle to slow down for lack of having a reservation. The intersection manager holds on to the reservation request until that time. In the meantime, it considers other vehicles' requests and can then grant reservations more efficiently.

Allowing this delayed response offers an immediate improvement over the current system. Consider the following example in which three vehicles, A, B, and C all send reservation requests to the intersection manager a short time after one another. Now suppose that vehicle A's request conflicts with both B's and C's (that is, they require the same reservation tile at a specific time), but that B's request does not conflict with C's. With our current system, the reservation manager would approve A's request, but reject both B and C. With the new system, only A would be rejected.

In addition to improving the efficiency of the system, adding a delayed response creates some opportunites to apply machine learning. In particular, as the number of outstanding reservation requests increases, the number of possible responses scales exponentially. Since timeliness is an important constraint, the intersection manager will need to intelligently search through set of possible responses in order to optimize the overall performance. Learned search control knowledge based on off-line optimization trials could play an important role in this regard.

Furthermore, projected incoming traffic can also play an important role. Once a reservation is accepted, it can't be cancelled. However, the parameters of reservations made in the near future are going to be related to the parameters of the reservations made now. For example, in heavy traffic, it may be best to reject a reservation request even when it doesn't conflict with many other requests in the same time frame — granting that reservation may cause the system to perform much more poorly at a slightly later time. In this sense, a learned model of incoming traffic as a function of time of day, day of week, and/or recent history could improve performance by serving as an input to the forward simulations of the impact of any given decision.

5.2 Vehicles With Priorities

In our current simulation, all vehicles are treated as equally important with regards to the performance metric. However in practice, the intersection should be able to give preferential treatment to a subset of vehicles, such as emergency vehicles. For example, a normal commuter would have a low priority, a police car would have a high priority, and an ambulance or fire truck en route to a fire would have yet a higher priority.

The first-cut solution to this problem is straightforward: whenever the reservation manager receives a request that conflicts with a request which it is currently holding, it rejects the lower priority request. This does enforce the constraint that higher priority vehicles are given preference, but is not optimal by any stretch of the imagination. Consider again three vehicles: a daily communter, a police car, and an ambulance racing a heart-attack victim to the hospital. If the commuter is in front of the ambulance and it is forced to yield to the police car, it will hold up the ambulance as well. If the intersection manager instead just allowed the commuter through, the ambulance may have been able to pass unhindered. The actual relationship between the times of a particular vehicle's reservation, that vehicle's priority, the characteristics of other approaching vehicles, and how much it is worth to the intersection to accept the reservation is very complicated. However, a reinforcement learning algorithm may be able to capture this relationship. When vehicles complete a trip across the intersection, the intersection manager could be given a reward signal inversely proportional to the delay the vehicle experienced. The manager could eventually learn to grant reservations based on the vehicles' priorities and the current traffic patterns so as to maximize the system's overall future reward.

5.3 The Intersection as a Market

Another consideration is that vehicles might have to pay to use the intersection. With states in the U.S. such as Oregon and California already considering taxing motorists by the mile, this is not far-fetched. Along with reservation requests, vehicles would transmit a bid. The reservation manager's goal would be to collect the most revenue. A first-cut solution would be analagous to the example with vehicle priorities: when a reservation comes in, reject any currently pending reservations that conflict with it and have a lower bid. This is obviously not optimal — consider any set of n vehicles such that for all $0 < i < n$, vehicle i and $i + 1$ conflict. As long as the bid for vehicle $i + 1$ is greater than that of vehicle i, the reservation manager will wind up only letting through vehicle n. Instead, it might have been able to allow through vehicles $1, 3, \ldots$. This is approximately $\frac{n}{2}$ vehicles and would generate a lot more revenue.

In this context, the intersection can be framed as a continually clearing combinatorial auction. The decision for any given grid cell must occur whenever the first car that needs it is about to enter the intersection. There is a tradeoff between letting a car through and retaining flexibility for later that the intersection manager must maintain. That is, letting an individual car through is good for the intersection manager. However, not letting that car through may lead to more positive benifits later on. Since even a single combinatorial auction can be computationally costly to solve, continually clearing, interacting combinatorial auctions are likely to be intractable. However, based on off-line simulation, the intersection manager could learn expected marginal values for granting a request to a given driver and therefore more effectively balance the above tradeoff.

6 Learning Opportunities for the Driver Agent

While there are many opportunities for the intersection manager to improve, they are mostly of the form of a single agent learning how to interact with multiple fixed agents (the drivers). The true *multiagent* learning opportunities lie in the vehicles.

6.1 Bidding in the Market System

In Section 5, we showed how a market could play an important role in the intersection management problem. In the example we gave, it wasn't clear how the agents should determine what bid to place with their reservation requests. An agent could start with a low bid and then continue raising it until one gets accepted, but this process takes time and it could wind up severely delayed just because it wasn't willing to commit to the higher bid up front. This is a very challenging problem — to solve it effectively would require a more detailed response from the intersection manager: the amount of the bid that caused the request to be rejected, the average bid amount for this particular intersection at this time of day, and so forth. Even with this type of information, though, it is unclear how to proceed. Learning the relationship between time of day, day of week, recent traffic reports, and a reasonable price for a reservation is a task well-suited to a neural network or other supervised learning algorithm. In off-line simulation, many vehicles could be run through the intersection, and when one gets a reservation, it could use the cost it eventually had to pay as a target value, weighted perhaps by how quickly it got the reservation.

6.2 Lane Changing

One of the features of our reservation system is the complete autonomy of driver agents while they are outside the intersection. Thus, when considering how to incorporate some sort of lane changing behavior, ideally we'd like to avoid having the intersection manager tell the vehicles which lane they should be in. However, as in the previous example, having the reservation manager (or some other source) provide the vehicle with relevant information could be extremely useful. For example, if an intersection manager realizes that one lane has a lot of cancelled reservations (e.g. from a stalled vehicle in that lane preventing other vehicles from fulfilling their reservations), this information might let vehicles know that they should switch to another lane instead of trying to make it through in the lane with the stalled car. It would then be interesting to explore how much and what kind of information the intersection manager is required to give the vehicles such that they can best choose which lane to use. If the driver agents were able to learn a better policy for lane choice, we could examine which information is useful for making that decision without having to first determine precisely how they are using it.

6.3 Making Better Reservations

In the current implementation, driver agents must find a way to make reservations that they can keep. To do this, they must be able to accurately predict when they will reach the intersection, accounting for delays from other vehicles and road hazards. In a real-life implementation, statistics and data the intersection manager has collected may be useful and thus made available to the driver agent. For example, as in both the bidding and lane-changing examples, the intersection manager may be able to provide vehicles with statistics on recent reservations. Once again, how to use these data is not immediately obvious and certainly depends on the algorithms (learning or otherwise) used by the other drivers. While the sensors in our simulated vehicles do not do it currently, they might be able to track the speed of the vehicle in front over the 10 seconds before making a reservation, or determine that the vehicle in front is a public bus and therefore might stop before the intersection for a long period of time. Given these new inputs, the driver agent could learn to better predict when and how it will arrive at the intersection.

7 Related Work

Rasche and Naumann have worked extensively on decentralized solutions to intersection collision avoidance problems [9, 11]. Many approaches focus on improving current technology (systems of traffic lights). For example, Roozemond allows intersections to act autonomously, sharing the data they gather [15]. The intersections then use this information to make both short- and long-term predictions about the traffic and adjust accordingly. This approach still assumes human-controlled vehicles. Bazzan has used an approach using both MAS and evolutionary game theory which involves multiple intersection managers (agents) that must focus not only on local goals, but also on global goals [1].

Work is also being done with regard to the control of the individual vehicles. Hallé and Chaib-draa have taken a MAS approach to collaborative driving by allowing vehicles to form *platoons*, groups of varying degrees of autonomy, that then coordinate using a hierarchical driving agent architecture [5]. While not focusing on intersections, Moriarty and Langley have shown that reinforcement learning can train efficient driver agents for lane, speed, and route selection during freeway driving [8].

On real autonomous vehicles, Kolodko and Vlacic have created a primitive system for intersection control which is very similar to the granularity-1 reservation system [7].

Actual systems in practice (not MAS) for traffic light optimization include TRANSYT [13], which is an off-line system requiring extensive data gathering and analysis, and SCOOT [6], which is an advancement over TRANSYT, responding to changes in traffic loads on-line. However, almost all of the methods in practice or discussed above still rely on traditional signalling systems.

8 Conclusion

The intersection management problem presents a challenging yet promising domain for multi-agent learning research. The intersection control mechanism we developed is a vast improvement over current methods, but with a few extensions poses some challenging problems. We have provided several examples of such problems where machine learning could be used to improve the performance of both intersection managers and driver agents. These examples are at this point speculative. In ongoing research we are investigating how to bring them and other learning opportunities into practice.

References

1. A. L. C. Bazzan. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multi-Agent Systems*, 10(2):131–164, March 2005.
2. K. Dresner and P. Stone. Multiagent traffic management: A protocol for defining intersection control policies. Technical Report UT-AI-TR-04-315, The University of Texas at Austin, Department of Computer Sciences, AI Laboratory, December 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, New York, 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. S. Hallé and B. Chaib-draa. A collaborative driving system based on multiagent modelling and simulations. *Journal of Transportation Research Part C (TRC-C): Emergent Technologies*, 2005. To appear.
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 1014, TRL Laboratory, 1981.
7. J. Kolodko and L. Vlacic. Cooperative autonomous driving at the intelligent control systems laboratory. *IEEE Intelligent Systems*, 18(4):8–11, July/August 2003.
8. D. Moriarty and P. Langley. Learning cooperative lane selection strategies for highways. In *Proceedings of the Fifeenth National Conference on Artificial Intelligence*, pages 684–691, Madison, WI, 1998. AAAI Press.
9. R. Naumann and R. Rasche. Intersection collision avoidance by means of decentralized security and communication management of autonomous vehicles. In *Proceedings of the 30th ISATA - ATT/IST Conference*, 1997.
10. D. A. Pormerleau. *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Publishers, 1993.
11. R. Rasche, R. Naumann, J. Tacken, and C. Tahedl. Validation and simulation of decentralized intersection collision avoidance algorithm. In *Proceedings of IEEE Conference on Intelligent Transportation Systems (ITSC 97)*, 1997.
12. C. W. Reynolds. Steering behaviors for autonomous characters. In *Proceedings of the Game Developers Conference*, pages 763–782, 1999.
13. D. I. Robertson. TRANSYT — a traffic network study tool. Technical Report TRRL-LR-253, Transport and Road Research Laboratory, Crowthorne, 1969.
14. S. Rogers, C.-N. Flechter, and P. Langley. An adaptive interactive agent for route advice. In O. Etzioni, J. P. Müller, and J. M. Bradshaw, editors, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 198–205, Seattle, WA, USA, 1999. ACM Press.
15. D. A. Roozemond. Using intelligent agents for urban traffic control control systems. In *Proceedings of the International Conference on Artificial Intelligence in Transportation Systems and Science*, pages 69–79, 1999.
16. T. Schonberg, M. Ojala, J. Suomela, A. Torpo, and A. Halme. Positioning an autonomous off-road vehicle by using fused DGPS and inertial navigation. In *2nd IFAC Conference on Intelligent Autonomous Vehicles*, pages 226–231, 1995.
17. P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, July 2000.
18. Texas Transportation Institute. 2004 urban mobility report, September 2004. Accessed at `http://mobility.tamu.edu/ums` in December 2004.