Traffic Congestion Management as a Learning Agent Coordination Problem

Kagan Tumer
Oregon State University
kagan.tumer@oregonstate.edu

Zachary T. Welch
Oregon State University
welch@engr.orst.edu

Adrian Agogino
UCSC/NASA ARC
adrian@email.arc.nasa.gov

Abstract

Traffic management problems provide a unique environment to study how multiagent systems promote desired system level behavior. In particular, they represent a special class of problems where the individual actions of the agents are neither intrinsically "good" nor "bad" for the system. Instead, it is the combinations of actions among agents that lead to desirable or undesirable outcomes. As a consequence, agents need to learn how to coordinate their actions with those of other agents, rather than learn a particular set of "good" actions. In this chapter, we focus on problems where there is no communication among the drivers, which puts the burden of coordination on the principled selection of the agent reward functions.

We explore the impact of agent reward functions on two types of traffic problems. In the first problem, we study how agents learn the best departure times in a daily commuting environment and how following those departure times alleviates congestion. In the second problem, we study how agents learn to select desirable lanes to improve traffic flow and minimize delays for all drivers. In both cases, we focus on having an agent select the most suitable action for each driver using reinforcement learning, and explore the impact of different reward functions on system behavior. Our results show that agent rewards that are both aligned with, and sensitive to, the system reward lead to significantly better results than purely local or global agent rewards. We conclude this chapter by discussing how changing the way in which the system performance is measured affects the relative performance of these rewards functions, and how agent rewards derived for one setting (timely arrivals) can be modified to meet a new system setting (maximize throughput).

1 Introduction

This purpose of this chapter is to quantify how decisions of local agents in a traffic system (e.g., drivers) affect overall traffic patterns. In particular, this chapter explores the system coordination problem of how to configure and update the system so that individual decisions lead to good system level behavior. From a broader perspective, this chapter demonstrates how to measure the alignment between the local agents in a system and the system at large. Because the focus of this chapter is on multiagent coordination and reward analysis, we focus on abstract, mathematical models of traffic rather full fledged simulations. Our main purpose is to demonstrate the impact of reward design and extract the key properties rewards need to have to alleviate congestion in large agent coordination problems, such as traffic.

In this chapter we apply multi-agent learning algorithms to two separate congestion problems. First we investigate how to coordinate the departure times of a set of drivers so that they do not end up producing traffic "spikes" at certain times, both providing delays at those times and causing congestion for future departures. In this problem, different time slots have different desirabilities that reflect user preferences for particular time slots. The system objective is to maximize the overall system's satisfaction as a weighted average of those desirabilities. In the second problem we investigate lane selection, where a set of drivers need to select different lanes to a destination (Moriarty and Langley, 1998; Pendrith, 2000). In this problem, different lanes have different capacities and the problem is for the agents to minimize the total congestion. Both problems share the same underlying property that agents greedily pursuing the best interests of their own drivers cause traffic to worsen for everyone in the system, including themselves.

Indeed, multi-agent learning algorithms provide a natural approach to addressing congestion problems in traffic and transportation domains (Bazzan et al., 1999; Dresner and Stone, 2004; Klügl et al., 2005). Congestion problems are characterized by having the system performance depend on the number of agents that select a particular action, rather on the intrinsic value of those actions. Examples of such problems include lane/route selection in traffic flow (Kerner and Rehborn, 1996; Nagel, 1997), path selection in data routing (Lazar et al., 1997), and side selection in the minority game (Challet and Zhang, 1998; Jefferies et al., 2002). In those problems, the desirability of lanes, paths or sides depends solely on the number of agents having selected them. Hence, multi-agent approaches that focus on agent coordination are ideally suited for these domains where agent coordination is critical for achieving desirable system behavior.

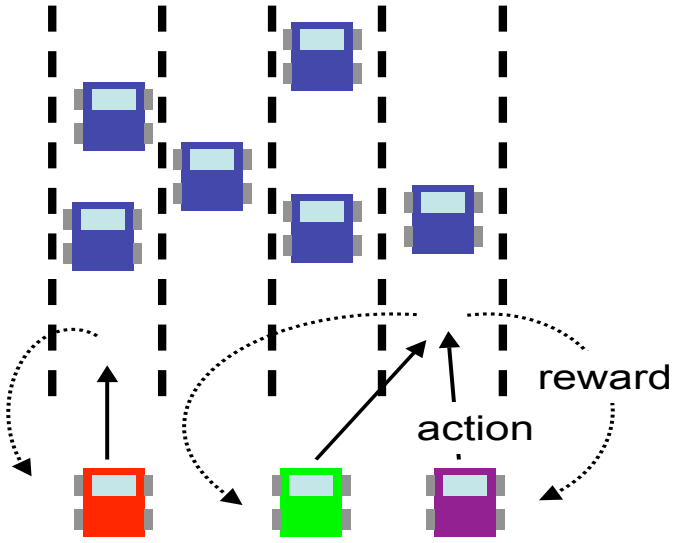

Figure 1: Reinforcement Learning for Congestion Management. A set of agents (cars) take actions. The result of each action is rewarded. Agents then modify their policy using reward.

The approach we present to alleviating congestion in traffic is based on assigning each driver an agent which determines the departure time/lane to select. The agents determine their actions based on a reinforcement learning algorithm (Littman, 1994; Sutton and Barto, 1998; Watkins and

Dayan, 1992). In this reinforcement learning paradigm, agents go through a process of where they take actions and receive rewards evaluating the effect of those actions. Based on these rewards the agents try to improve their actions (see Figure 1). The key issue in this approach is to ensure that the agents receive rewards that promote good system level behavior. To that end, it is imperative that the agent rewards: (i) are aligned with the system reward[1], ensuring that when agents aim to maximize their own reward they also aim to maximize system reward; and (ii) are sensitive to the actions of the agents, so that the agents can determine the proper actions to select (i.e., they need to limit the impact of other agents in the reward functions of a particular agent).

The difficulty in agent reward selection stems from the fact that typically these two properties provide conflicting requirements. A reward that is aligned with the system reward usually accounts for the actions of other agents, and thus is likely to not be sensitive to the actions of one agent; on the other hand, a reward that is sensitive to the actions of one agent is likely not to be aligned with system reward. This issue is central to achieving coordination in a traffic congestion problem and has been investigated in various fields such as computational economics, mechanism design, computational ecologies and game theory (Boutilier, 1996; Sandholm and Crites, 1995; Huberman and Hogg, 1988; Parkes, 2001; Stone and Veloso, 2000). We address this reward design problem using the difference reward (Wolpert and Tumer, 2001; Tumer and Wolpert, 2004), which provides a good balance of alignedness and sensitivity. The difference reward has been applied to many domains, including rover coordination (Agogino and Tumer, 2004), faulty device selection problem (Tumer, 2005), packet routing over a data network (Tumer and Wolpert, 2000; Wolpert et al., 1999), and modeling nongenomic models of early life (Gupta et al., 2006).

The overall objective of this chapter is to show how agent reward design, coupled with reinforcement learning agents can be used to alleviate traffic congestion, and show experimental results illustrating that these methods are both effective and robust to non-compliance (i.e., drivers not following the suggestions of their agents). In Section 2 we discuss the properties agent rewards need to have and present a particular example of agent reward. In Sections 3.1 and 3.2 we present the departure coordination problem. The results in this domain show that total traffic delays can be improved significantly when agents use the difference reward. In Section 3.3 we present the lane selection problem. The results in this domain show that traffic congestion can be reduced by over 30% when agents use the difference reward. In Section 3.4, we investigate how the system performance degrades when the percentage of drivers who do not follow the advice of their agents increases from 0 to 100 %, a critical issue for the adoption of any new traffic algorithm. Finally, in Section 4 we discuss the implication of these results, discuss methods by which they can be applied in the traffic domain, and highlight future research directions.

2 Background

In this chapter we propose modeling cars as agents that individually try to maximize a reward through a reinforcement learning process. The types of rewards the agents receive depend on our goals for the system and our approach to system. While in some cases all the agents will get the same reward, in general an agent will get a reward unique to the agent. Finding appropriate rewards that will entice agents take actions towards a collective goal is critical to the success of this method.

More formally we are modeling traffic congestion management as a multi-agent systems where

[1]We call the function rating the performance of the full system, "system reward" throughout this chapter in order to emphasize its relationship to agent rewards.

each agent, i, tries to maximize its reward function $g_i(z)$, where z depends on the joint move of all agents. Furthermore, there is a system reward function, $G(z)$ which rates the performance of the full system. To distinguish states that are impacted by actions of agent i, we decompose[2] z into $z = z_i + z_{-i}$, where z_i refers to the parts of z that are dependent on the actions of i, and z_{-i} refers to the components of z that do not depend on the actions of agent i.

2.1 Properties of Reward Functions

For learning agents to be able to perform effectively in a multi-agent system it is critical that the rewards have two properties:

- rewards are aligned with the overall goal.
- rewards are sensitive to the agent's actions.

First, the agent rewards have to be aligned with respect to G, quantifying the concept that an action taken by an agent that improves its own reward also improves the system reward. Formally, for systems with discrete states, the degree of **factoredness** for a given reward function g_i is defined as:

$$\mathcal{F}_{g_i} = \frac{\sum_z \sum_{z'} u[(g_i(z) - g_i(z'))\ (G(z) - G(z'))]}{\sum_z \sum_{z'} 1} \tag{1}$$

for all z' such that $z_{-i} = z'_{-i}$ and where $u[x]$ is the unit step function, equal to 1 if $x > 0$, and zero otherwise. Intuitively, the higher the degree of factoredness between two rewards, the more likely it is that a change of state will have the same impact on the two rewards. A system is fully factored when $\mathcal{F}_{g_i} = 1$.

Second, an agent's reward has to be sensitive to its own actions and insensitive to actions of others. Formally we can quantify the **learnability** of reward g_i, for agent i at z:

$$\lambda_{i,g_i}(z) = \frac{E_{z'_i}[|g_i(z) - g_i(z_{-i} + z'_i)|]}{E_{z'_{-i}}[|g_i(z) - g_i(z'_{-i} + z_i)|]} \tag{2}$$

where $E[\cdot]$ is the expectation operator, z'_i's are alternative actions of agent i at z, and z'_{-i}'s are alternative joint actions of all agents other than i. Intuitively, learnability provides the ratio of the expected value of g_i over variations in agent i's actions to the expected value of g_i over variations in the actions of agents other than i. So at a given state z, the higher the learnability, the more $g_i(z)$ depends on the move of agent i, i.e., the better the associated signal-to-noise ratio for i. Higher learnability means it is easier for i to achieve large values of its reward.

In the domain of congestion management a reward with the first property means that actions that help reduce the overall congestion are rewarded. This is in contrast to a greedy reward, which may reward actions taken that help an individual driver, but actually cause overall congestion to increase. However, in general, this property isn't sufficient since it does not concern itself with whether the agents can actually maximize their own reward. Consider the extreme example of a driver only being rewarded with a "good" or "bad" score depending on the traffic report at the end

[2]Instead of concatenating partial states to obtain the full state vector, we use zero-padding for the missing elements in the partial state vector. This allows us to use addition and subtraction operators when merging components of different states (e.g., $z = z_i + z_{-i}$).

of the day summarizing the day's congestion. While this reward is aligned with our overall goal of reducing congestion, if there are thousands (if not millions) of drivers, a driver would not be able to see the effect of his/her individual actions on this reward. Instead we need rewards that balance being aligned with our goal while being sensitive to the driver's actions, so that the drivers can effectively learn to maximize their rewards.

2.2 Difference Reward Functions

Let us now focus on providing agent rewards that are both high factoredness and high learnability. Consider the **difference** reward (Wolpert and Tumer, 2001), which is of the form:

$$D_i \equiv G(z) - G(z_{-i} + c_i) \tag{3}$$

where z_{-i} contains all the states on which agent i has no effect, and c_i is a fixed vector. In other words, all the components of z that are affected by agent i are replaced with the fixed vector c_i. Such difference reward functions are fully factored no matter what the choice of c_i, because the second term does not depend on i's states (Wolpert and Tumer, 2001). Furthermore, they usually have far better learnability than does a system reward function, because the second term of D removes some of the effect of other agents (i.e., noise) from i's reward function. In many situations it is possible to use a c_i that is equivalent to taking agent i out of the system. Intuitively this causes the second term of the difference reward function to evaluate the value of the system without i and therefore D evaluates the agent's contribution to the system reward.

The difference reward can be applied to any linear or non-linear system reward function. However, its effectiveness is dependent on the domain and the interaction among the agent reward functions. At best, it fully cancels the effect of all other agents. At worst, it reduces to the system reward function, unable to remove any terms (e.g., when z_{-i} is empty, meaning that agent i effects all states). In most real world applications, it falls somewhere in between, and has been successfully used in many domains including agent coordination, satellite control, data routing, job scheduling and congestion games (Agogino and Tumer, 2004; Tumer and Wolpert, 2000; Wolpert and Tumer, 2001). Also note that computationally the difference reward is often easier to compute than the system reward function (Tumer and Wolpert, 2000). Indeed in the problem presented in this chapter, for agent i, D_i is easier to compute than G is (see details in Section 3.1.1).

2.3 Reward Maximization

In this chapter we assume that each agent maximize its own reward using its own reinforcement learner (though alternatives such as evolving neuro-controllers are also effective (Agogino and Tumer, 2004)). In this paradigm, an agent will take an action based on a policy and will then receive a reward evaluating its action. The agent will then use this reward to update its action policy. For complex delayed-reward problems, relatively sophisticated reinforcement learning systems such as temporal difference may have to be used. However, the traffic domain modeled in this chapter only needs to utilize immediate rewards, therefore a simple table-based immediate reward reinforcement learning is used. Our reinforcement learner is equivalent to an ϵ-greedy with a discount rate of 0. At every episode an agent takes an action and then receives a reward evaluating that action. After taking action a and receiving reward R a driver updates its table as follows:

$$Q(a) \leftarrow (1 - \alpha)Q(a) + \alpha(R),$$

where α is the learning rate. At every time step the driver chooses the action with the highest table value with probability $1 - \epsilon$ and chooses a random action with probability ϵ. In the experiments described in the following section, α is equal to 0.5 and ϵ is equal to 0.05. The parameters were chosen experimentally, though system performance was not overly sensitive to these parameters.

3 Experiments

To test the effectiveness of our rewards in the traffic congestion domain, we performed experiments using two abstract traffic models. In the first model each agent has to select a time slot to start its drive. In this model we explore both simple and cascading traffic flow. With non-cascading flow, drivers enter and exit the same time slot, while with cascading flow, drivers stuck in a time slot with too many other drivers stay on the road for future time slots.

In the second model, instead of choosing time slots, drivers choose lanes. This model differs from the time-slot model in that different lanes may also have different capacities (for example because of carpool restrictions). In this model we also use a slightly different objective function that seeks to avoid congestion, in contrast to maximizing throughput.

Between these models we performed six sets of experiments as follows:

1. Departure time selection for simple traffic flow model:
   (a) Single peak congestion - Heavy congestion peaks around a single time slot.
   (b) Double peak congestion - Heavy congestion peaks around two time slots.
   (c) Non-Symmetric congestion - Congestion progressively increases with time.
2. Departure time selection for cascading congestion for single peak congestion.
3. Lane Selection - Drivers reduce congestion using lane selection model.
4. Driver compliance - Test ability of learning agents to reduce congestion when some of the drivers are taking random actions instead of trying to reduce congestion.

3.1 Departure Time Selection

In the traffic congestion model we first explore, there is a fixed set of drivers, and the task of the agents is to find the time slot in which their drivers start their commutes. The system performance is measured from the perspective of a "city manager" (as opposed to a social welfare function based on the intrinsic rewards of the drivers) that directly measures a system wide performance criteria. To highlight this, we will denote the system level function of the City Manager by (dubbed G in the previous section) by $S(CM)$:

$$G = S(CM) = \sum_t w_t S(k_t) \; . \tag{4}$$

where weights w_t model rush-hour scenarios where different time slots have different desirabilities, and $S(k)$ is a "time slot reward", depending on the number of agents that chose to depart in the time slot:

$$S(k) = \begin{cases} ke^{-1} & \text{if } k \leq c \\ ke^{-k/c} & \text{otherwise} \end{cases} \; , \tag{5}$$

The number of drivers in the time slot is given by k, and the optimal capacity of the time slot is given by c. Below an optimal capacity value c, the reward of the time slot increases linearly with the number of drivers. When the number of drivers is above the optimal capacity level, the value of the time slot decreases quickly (asymptotically exponential) with the number of drivers. This reward models how drivers do not particularly care how much traffic is on a road until it is congested. This function is shown in Figure 2. In this problem, the task of the system designer is to have the agents choose time slots that help maximize the system reward. To that end, agents have to balance the benefit of going at preferred time slots with the congestion at those time slots.

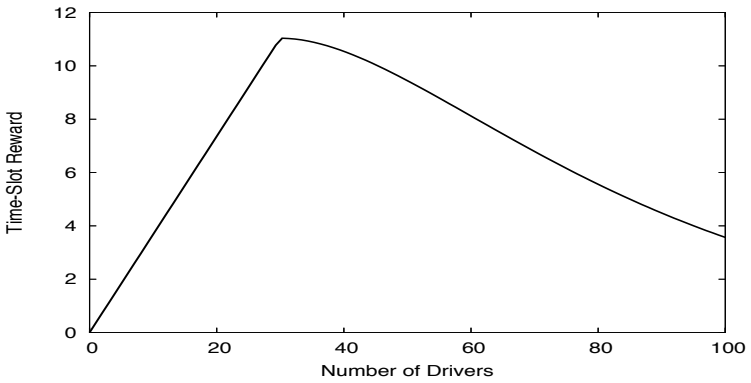

Figure 2: Reward of time slot with $c = 30$.

3.1.1 Driver Rewards

While as a system designer our goal is to maximize the system reward, we have each individual agent try to maximize a driver-specific reward that we select. The agents maximize their rewards through reinforcement learning, where they learn to choose time slots that have expected high reward. In these experiments, we evaluate the effectiveness of three different rewards. The first reward is simply the system reward $G = S(CM)$, where each agent tries to maximize the system reward directly. The second reward is a local reward, L_i where each agent tries to maximize a reward based on the time slot it selected:

$$L_i(k) = w_i S(k_i) \ , \tag{6}$$

where k_i is the number of drivers in the time slot chosen by driver i. The final reward is the difference reward, D:

$$\begin{aligned} D_i &= G(k) - G(k_{-i}) \\ &= \sum_j L_j(k) - \sum_j L_j(k_{-i}) \\ &= L_i(k) - L_i(k_{-i}) \\ &= w_i k_i S(k_i) - w_i(k_i - 1)S(k_i - 1) \ , \end{aligned}$$

where k_{-i} represents the driver counts when driver i is taken out of the system. Note that since taking away driver i only affects one time slot, all of the terms but one cancel out, making the difference reward simpler to compute than the system reward.

3.1.2 Single Peak Congestion Results

In this set of experiments there were 500 drivers, and the optimal capacity of each time slot was 125. Furthermore, the weighting vector was centered at the most desirable time slot (e.g., 5 PM departures), simulating a single peak congestion:

$$w = [1\ 5\ 10\ 15\ 20\ 15\ 10\ 5\ 1]^T\ .$$

This weighting vector reflects a preference for starting a commute at the end of the workday with the desirability of a time slot decreasing for earlier and later times. All performance plots reflect agent daily agent learning (the "time step" is one day, in that each day the agents make new choices).

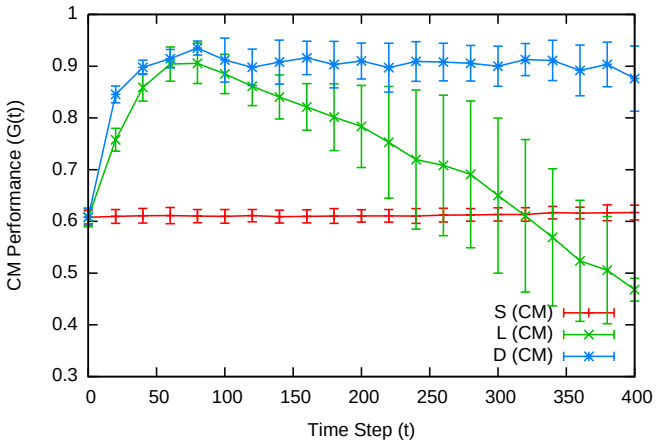

Figure 3: Performance on Departure Time Selection Problem with single peak congestion. In this and all subsequent figures, we present local (L), Difference (D) and System (S) rewards based on the City Manager (CM) perspective. Drivers using difference reward quickly learn to achieve near optimal performance (1.0). Drivers using system reward do not learn at all. Drivers using non-factored local reward slowly learn counterproductive actions.

This experiment shows that drivers using the difference reward are able to quickly obtain near-optimal system performance (see Figure 3). In contrast, drivers that try to directly maximize the system reward do not learn at all and never achieve good performance during the time-frame of the experiment. This lack of learning is a result of the system reward having low learnability to the agents' actions. Even if a driver were to take a system wide coordinated action, it is likely that some of the 499 other drivers would take uncoordinated actions at the same time, lowering the value of the system reward. A driver using the system reward typically does not get proper credit assignment for its actions, since the reward is dominated by other drivers.

The experiment where drivers are using L (a non-factored local reward) exhibit some interesting performance properties. At first these drivers learn to improve the system reward. However, after about episode seventy their performance starts to decline. Figure 4 gives greater insight into this phenomenon. At the beginning of the experiment, the drivers are randomly distributed among time slots, resulting in a low reward. Later in training agents begin to learn to use the time slots that

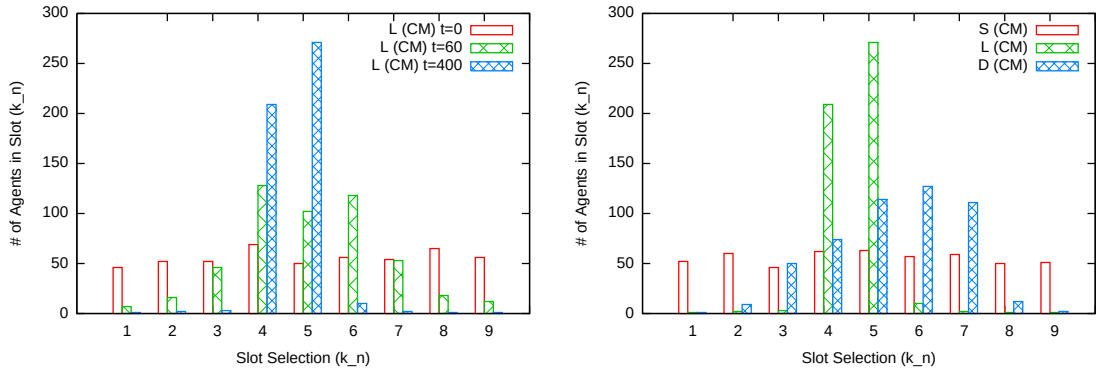

Figure 4: Slot distributions for single peak congestion: (a) Distribution of drivers using local reward. Early in training drivers learn good policies. Later in learning, the maximization of local reward causes drivers to over utilize high valued time slots. (b) Distribution of drivers at end of training for all three rewards. Drivers using difference reward form distribution that is closer to optimal than drivers using system of local rewards.

have the most benefit. When the number of drivers reach near optimal values for those time slots, the system reward is high. However, all agents in the system covet those time slots and more agents start to select the desirable time slots. This causes congestion and system reward starts to decline. This performance characteristics is typical of system with agent rewards of low factoredness. In such a case, agents attempting to maximize their own rewards lead to undesirable system behavior. In contrast, because their rewards are factored with the system reward, agents using the difference reward form a distribution that more closely matches the optimal distribution (Figure 4).

3.1.3 Double Peak Congestion Results

In many situations, there are multiple desirable departure times, resulting in multi-modal peak departure distribution. To verify that the performance obtained in the previous section was not due to the weight vector, we investigated the agent response to a weight profile that provided double peaks:

$$w = [1\ 10\ 20\ 10\ 1\ 10\ 20\ 10\ 1]^T$$

Figures 5 and 6 show performance for the double peak weight vector, along with the histograms of slot counts for agents using the local reward (over time) and all rewards (at the end of the simulation), respectively. In this case, because the problem was more difficult and required some degree of coordination from the starting point, the performance of the local reward never reached the performance of the difference reward. However, the same performance drop is observed in this case, where agents pursuing the local reward start a decline that leads them to very poor solutions. This can be seen in Figure 6(a) where the local reward never finds the "good" distribution found by the difference reward in Figure 6(b). In contrast, the agents using the difference reward were not affected by the difficulty of the problem and reached a good solution in very few training steps.

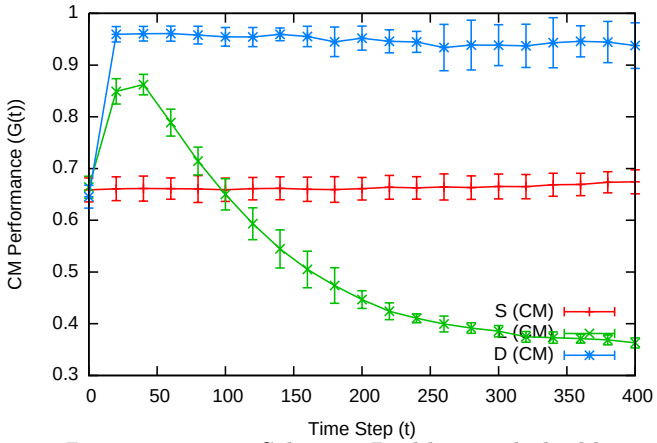

Figure 5: Performance on Departure Time Selection Problem with double peak congestion. Drivers using difference reward quickly learn to achieve near optimal performance (1.0). Drivers using system reward do not learn at all. Drivers using non-factored local reward quickly learn counter-productive actions.

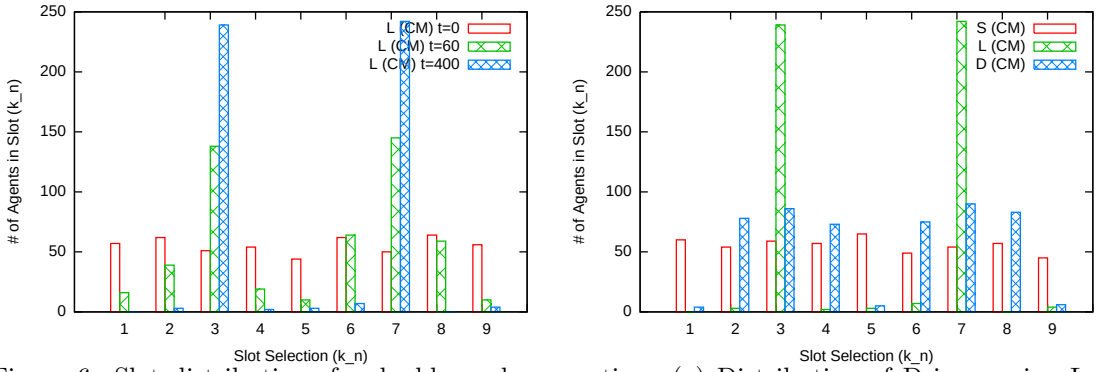

Figure 6: Slot distributions for double peak congestion: (a) Distribution of Drivers using Local Reward. where The maximization of local reward causes drivers to quickly start to over utilize high valued time slots. (b) Distribution of Drivers at end of Training for all three rewards. Drivers using difference reward form near optimal distribution.

3.1.4 Non-Symmetric Congestion Results

Finally, we explored the performance of the various rewards functions for a non-symmetric weight distribution:

$$w = [1\ 1\ 2\ 3\ 5\ 8\ 13\ 21\ 34]^T$$

Figures 7 and 8 show performance for the non-symmetric weight vector, along with the histograms of slot counts for agents using the local reward (over time) and all rewards (at the end of the simulation), respectively. It is clear from this (and the double peak experiment) that the initial, single peak weights were more favorable to agents using the local reward than to agents using either the difference reward or the full system reward. In these two difficult cases, agents using the local reward never reach the performance of the difference reward, and their drop in performance begins almost immediately. In contrast, the original single peak environment had yielded improved performance for a longer time period before succumbing to clustering effects.

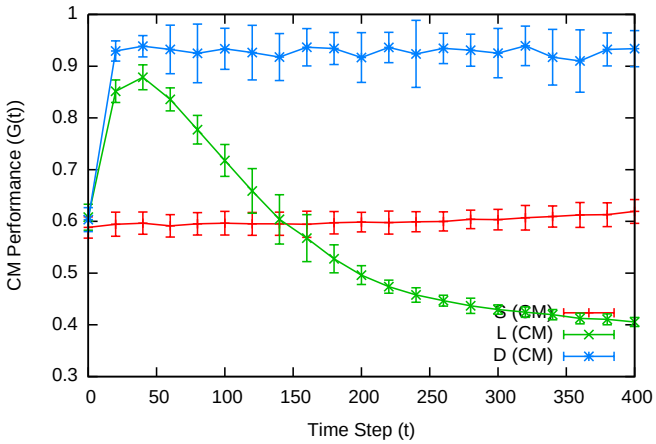

Figure 7: Performance on Departure Time Selection Problem with non-symmetric congestion. Drivers using difference reward quickly learn to achieve near optimal performance (1.0). Drivers using system reward do not learn at all. Drivers using non-factored local reward quickly learn counterproductive actions.

3.2 Cascading Traffic for Departure Time Selection

The previous model assumes that drivers enter and leave the same time slot. Here we introduce a more complex model, where drivers remain in the system longer when it is congested. This property is modeled by having drivers over the optimal capacity, c stay in the system until they reach a time slot with a traffic level below c. When the number of drivers in a time slot is less than c the reward for a time slot is the same as before. When the number of drivers is above c the linear term k is replaced with c:

$$S(k) = \begin{cases} ke^{-1} & \text{if } k \leq c \\ ce^{-k/c} & \text{otherwise} \end{cases} \tag{7}$$

As before the system reward is a sum of the time slot rewards: $G = \sum_t S(k_t)$.

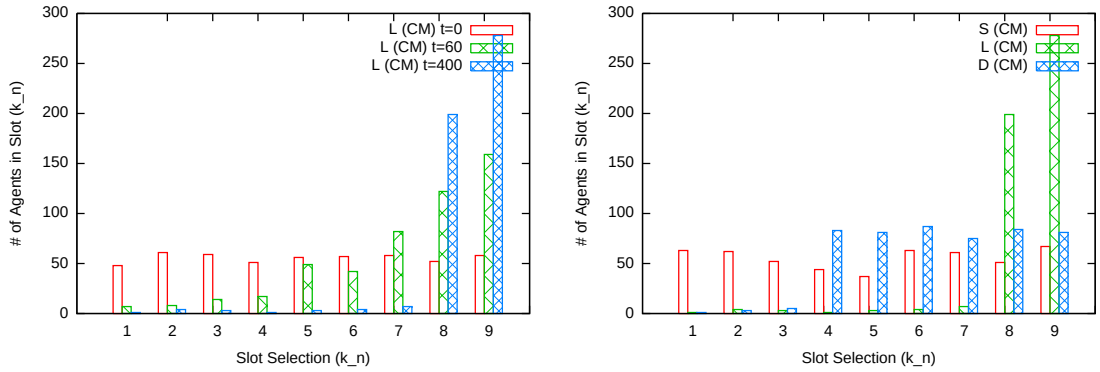

Figure 8: Slot distributions for non-symmetric congestion: (a) Distribution of Drivers using Local Reward. The maximization of local reward causes drivers to quickly start to over utilize high valued time slots. (b) Distribution of Drivers at end of Training for all three rewards. Drivers using difference reward form distribution that is closer to optimal than drivers using system of local rewards.

3.2.1 Driver Rewards

Again the local reward is the weighted time slot reward:

$$L_i = w_i S(k_i) \ , \tag{8}$$

where k_i is the number of drivers in the time slot chosen by driver i. However the difference reward is more difficult to simplify as the actions of a driver can have influence over several time slots:

$$\begin{aligned} D_i &= G(k) - G(k_{-i}) \\ &= \sum_j w_j S(k_j) - \sum_j w_j S(k_{-i_j}) \ , \end{aligned}$$

where k_{-i_j} is the number of drivers there would have been in time slot j had driver i not been in the system.

3.2.2 Results

Figure 9 shows the results for cascading traffic model for the single peak weight vector given by $w = [1\ 5\ 10\ 15\ 20\ 15\ 10\ 5\ 1]^T$. As previously, there are 500 drivers and time slot capacities are 125. Drivers using the different rewards exhibit similar characteristics on this model than on the non-cascading one. Again drivers using the system reward are unable to improve their performance significantly beyond their initial random performance.

In this model drivers using the local reward perform worse than in the simple cascading model (Results in Figure 3) once they become proficient at maximizing their own reward. This is because bad choices have longer lasting impact in this model. As a result, when drivers using the local reward cause congestion for their time slots, the congestion cascades as drivers spill into future time slots causing a significant decrease in performance. The performance of the three different

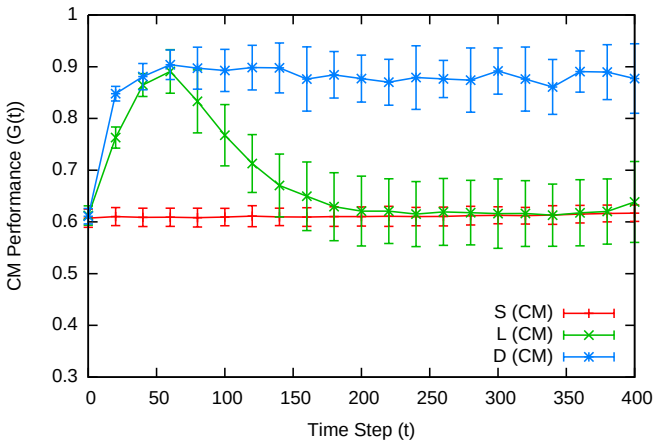

Figure 9: Performance on Cascading Departure Time Selection Problem. In this domain drivers above the capacity in one time slot remain in system in future time slots. Drivers using difference reward quickly learn to achieve near optimal performance (1.0).

rewards for the double peak and non-symmetric weight vectors are similar to those obtained in Sections 3.1.3 and 3.1.4, in that the local rewards degrade faster than for the single peak vector. We omit the details of those experiments for brevity, as they do not provide additional insight into agent behavior.

3.3 Lane Selection Congestion Model

In this model instead of selecting time slots, drivers select lanes. The main difference in this model is the functional form of the reward for a lane as shown in Figure 10. In this model the objective is to keep the lanes uncongested. The system reward does not care how many drivers are on a particular lane as long as that lane is below its congestion point. Each lane has a different weight representing overall driver preference for a lane. Furthermore, each lane has its own capacity, modeling the realities that some lanes having more restrictions such as tolls and/or carpools.

In this model the reward for an individual lane is:

$$S_{Lane}(k,c) = \begin{cases} e^{-1} & \text{if } k \leq c \\ e^{-k/c} & \text{otherwise} \end{cases} \tag{9}$$

The system reward is then the sum of all lane rewards weighted by the value of the lane.

$$G = \sum_i w_i S_{Lane}(k_i, c_i) \ , \tag{10}$$

where w_i is the weighting for lane i and c_i is the capacity for lane i.

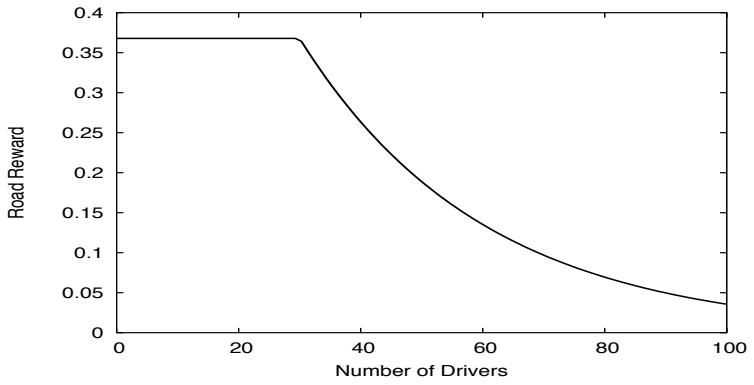

Figure 10: Reward of Road with $c = 30$.

3.3.1 Driver Rewards

Again three rewards were tested: the system reward, the local reward and the difference reward. The local reward is the weighted reward for a single lane:

$$L_i = w_i S_{Lane}(k_i, c_i) \ . \tag{11}$$

The final reward is the difference reward, D:

$$\begin{aligned} D_i &= G(k) - G(k_{-i}) \\ &= L_i(k) - L_i(k_{-i}) \\ &= w_i S_{Lane}(k_i, c_i) - w_i S_{Lane}(k_i - 1, c_i) \ , \end{aligned}$$

representing the difference between the actual system reward and what the system reward would have been if the driver had not been in the system.

3.3.2 Results

Here we show the results of experiments where we test performance of the three rewards in the multi-lane model, where different lanes have different value weightings and different capacities. There were 500 drivers in these experiments and the lane capacities were 167, 83, 33, 17, 9, 17, 33, 83, 167. Each lane is weighted with the weights 1, 5, 10, 1, 5, 10, 1, 5, 10. Figure 11 shows that drivers using the system reward perform poorly, and learn slowly. Again drivers using the difference reward perform the best, learning quickly to achieve an almost optimal solution. Drivers using the local reward learn more quickly early in training than drivers using the system reward, but never achieve as high as performance as those using the difference reward. However in this domain the drivers using the local reward do not degrade from their maximal performance, but instead enter a steady state that is significantly below that of the drivers using the difference reward.

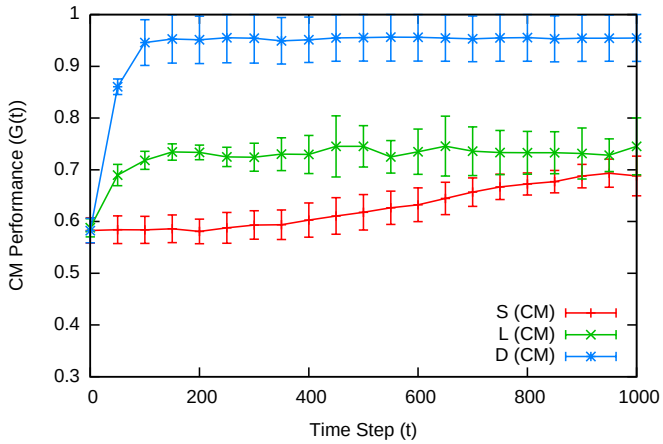

Figure 11: Performance on Domain with Multiple Lanes. Best observed performance = 1.0 (optimal not calculated)

3.4 Non-Compliant Drivers

All the pervious experiments presented in this chapter have assumed that all the drivers are actively participating in the learning system. However in most real-world situations this will not happen. In many traffic scenarios it may be only possible to convince some of the drivers to participate in a particular scheme. In addition even if all the drivers agree participate, due to various information/sensing limitations, some of the drivers may not be able to. To test this situation we conducted a set of experiments where a certain percentage of drivers did not participate in the learning paradigm. Instead they took random actions.

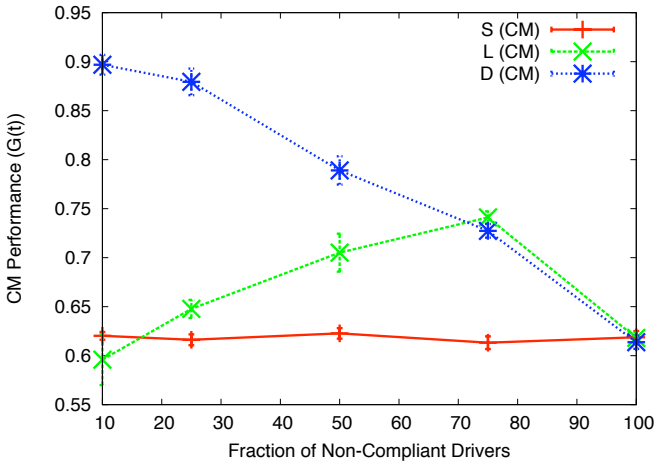

Figure 12: Performance on Time Selection Problem with Non-Compliant Drivers. With a moderate number of non-compliant drivers difference reward is still able to perform well.

The results (Figure 12) show that the proposed paradigm is robust even when a moderate amount of drivers are non-compliant. As before, drivers using the system reward perform uniformly poorly. Interestingly drivers using the local reward actually can improve their performance when the number of non-compliant drivers increases. This is not surprising since the drivers using local rewards where actually learning how to make counter productive actions. When noise is added to the system in the form of non-compliant drivers, the drivers using the local reward were less able to learn these counter productive actions.

Finally, drivers using the difference reward perform better when more drivers conform, but their performance degrades gracefully with the number of non-compliant drivers. This is a key result that implies such a system can be implemented in stages with improvements acting as "advertising" to entice others to participate in the system.

4 Conclusions and Future Research Directions

This chapter presented a method for improving congestion in two different traffic problems. First we presented a method by which agents can coordinate the departure times of drivers in order to alleviate spiking at peak traffic times, demonstrating its effectiveness in two similar congestion models. Second we showed that agents can manage effective lane selection and significantly reduce congestion by using a reward structure that penalizes greedily seeking the lanes with high capacity.

These results are based on agents receiving rewards that have high factoredness and high learnability (i.e., are both aligned with the system reward and are as sensitive as possible to changes in the reward of each agent). In these experiments, agents using difference rewards produced near optimal performance (93-96% of optimal). Agents using system rewards (63-68%) performed comparably to random action selection (62-64%), and agents using local rewards (48-72%) provided performance ranging from mediocre to worse than random in the instances when their own interests did not align with the system reward (i.e., city manager's reward).

Finally, one issue that arises in traffic problems that does not arise in many other domains (e.g., rover coordination) is in ensuring that drivers follow the advice of their agents. We showed that the system is robust when a large number of drivers do not participate in the optimization system. A related problem also arises when the city manager's system reward is at odds with a social welfare function based on timeliness desires of the drivers. Determining what incentives to provide to the agents so that these two seemingly different objectives can be simultaneously maximized is a critical problem that has recently been investigated (Tumer et al., 2008), but bears further study.

However, in this chapter, we did not address the issue of what drivers do when it is not in their interest to follow the advice of their agents. The purpose of this chapter was to show that solutions to the difficult traffic congestion problem can be addressed in a distributed adaptive manner using intelligent agents. Ensuring that drivers follow the advice of their agents is a fundamentally different problem. One can expect that drivers will notice that the departure times/lanes suggested by their agents provide significant improvement over their regular patterns. However, as formulated, there are no mechanisms for ensuring that a driver does not gain an advantage by ignoring the advice of his or her agent. Future work includes investigating this issue, exploring the alignment/mismatch between a city manager's utility and a social welfare reward based on the agent's intrinsic rewards and verifying these results in a traffic simulator.

References

Agogino, A. and Tumer, K. (2004). Efficient evaluation functions for multi-rover systems. In *The Genetic and Evolutionary Computation Conference*, pages 1–12, Seatle, WA.

Balmer, M., Cetin, N., Nagel, K., and Raney, B. (2004). Towards truly agent-based traffic and mobility simulations. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 60–67, New York, NY.

Bazzan, A. L. and Klügl, F. (2005). Case studies on the Braess paradox: simulating route recommendation and learning in abstract and microscopic models. *Transportation Research C*, 13(4):299–319.

Bazzan, A. L., Wahle, J., and Klügl, F. (1999). Agents in traffic modelling – from reactive to social behaviour. In *KI – Kunstliche Intelligenz*, pages 303–306.

Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. In *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge*, Holland.

Challet, D. and Zhang, Y. C. (1998). On the minority game: Analytical and numerical studies. *Physica A*, 256:514.

Dresner, K. and Stone, P. (2004). Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 530–537, New York, NY.

Gupta, N., Agogino, A., and Tumer, K. (2006). Efficient agent-based models for non-genomic evolution. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Hakodate, Japan.

Hall, S. and Draa, B. C. (2004). Collaborative driving system using teamwork for platoon formations. In *The third workshop on Agents in Traffic and Transportation.*

Huberman, B. A. and Hogg, T. (1988). The behavior of computational ecologies. In *The Ecology of Computation*, pages 77–115. North-Holland.

Jefferies, P., Hart, M. L., and Johnson, N. F. (2002). Deterministic dynamics in the minority game. *Physical Review E*, 65 (016105).

Kerner, B. S. and Rehborn, H. (1996). Experimental properties of complexity in traffic flow. *Physical Review E*, 53(5):R4275–4278.

Klügl, F., Bazzan, A., and Ossowski, S., editors (2005). *Applications of Agent Technology in Traffic and Transportation.* Springer.

Lazar, A. A., Orda, A., and Pendarakis, D. E. (1997). Capacity allocation under noncooperative routing. *IEEE Transactions on Networking*, 5(6):861–871.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 157–163.

Moriarty, D. E. and Langley, P. (1998). Learning cooperative lane selection strategies for highways. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 684–691, Madisson, WI.

Nagel, K. (1997). Experiences with iterated traffic microsimulations in dallas. pre-print adap-org/9712001.

Nagel, K. (2001). Multi-modal traffic in TRANSIMS. In *Pedestrian and Evacuation Dynamics*, pages 161–172. Springer, Berlin.

Parkes, D. C. (2001). *Iterative Combinatorial Auctions: Theory and Practice*. PhD thesis, University of Pennsylvania.

Pendrith, M. D. (2000). Distributed reinforcement learning for a traffic engineering application. In *Proceedings of the fourth international conference on Autonomous Agents*, Barcelona, Spain.

Sandholm, T. and Crites, R. (1995). Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:147–166.

Stone, P. and Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Tumer, K. (2005). Designing agent utilities for coordinated, scalable and robust multi-agent systems. In Scerri, P., Mailler, R., and Vincent, R., editors, *Challenges in the Coordination of Large Scale Multiagent Systems*. Springer.

Tumer, K., Welch, Z., and Agogino, A. (2008). Aligning social welfare and agent preferences to alleviate traffic congestion. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Estoril, Portugal. To appear.

Tumer, K. and Wolpert, D., editors (2004). *Collectives and the Design of Complex Systems*. Springer, New York.

Tumer, K. and Wolpert, D. H. (2000). Collective intelligence and Braess' paradox. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 104–109, Austin, TX.

Watkins, C. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3/4):279–292.

Wolpert, D. H. and Tumer, K. (2001). Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(2/3):265–279.

Wolpert, D. H., Tumer, K., and Frank, J. (1999). Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems - 11*, pages 952–958. MIT Press.