Aligning social welfare and agent preferences to alleviate traffic congestion

Kagan Tumer
Oregon State University
204 Rogers Hall
Corvallis, OR 97331
kagan.tumer@oregonstate.edu

Zachary T Welch
Oregon State University
204 Rogers Hall
Corvallis, OR 97331
welch@engr.orst.edu

Adrian Agogino
UCSC, NASA Ames Res. Ctr.
Mailstop 269-3
Moffett Field, CA 94035, USA
adrian@email.arc.nasa.gov

ABSTRACT

Multiagent coordination algorithms provide unique insights into the challenging problem of alleviating traffic congestion. What is particularly interesting in this class of problem is that no individual action (e.g., leave at a given time) is intrinsically "bad" but that combinations of actions among agents lead to 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 general, the traffic problem can be approached from two distinct perspectives: (i) from a city manager's point of view, where the aim is to optimize a city wide objective function (e.g., minimize total city wide delays), and (ii) from the individual driver's point of view, where each driver is aiming to optimize a personal objective function (e.g., a "timeliness" function that minimizes the difference desired and actual arrival times at a destination). In many cases, these two objective functions are at odds with one another, where drivers aiming to optimize their own objectives yield to congestion and poor values of city objective functions.

In this paper we present an objective shaping approach to both types of problems and study the system behavior that arises from the drivers' choices. We first show a top-down approach that provides incentives to drivers and leads to good values of the city manager's objective function. We then present a bottom-up approach that shows that drivers aiming to optimize their own personal timeliness objective lead to poor performance with respect to a city manager's objective function. Finally, we present the intriguing result that drivers that aim to optimize a modified version of their own timeliness function not only perform well in terms of the city manager's objective function, but also perform better with respect to their own original timeliness functions.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Performance

Cite as: Aligning social welfare and agent preferences to alleviate traffic congestion, Kagan Tumer, Zachary T Welch and Adrian Agogino, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May,12-16.,2008,Estoril,Portugal,pp. 655-662.

Keywords

Traffic Management, Transportation, Multiagent Systems, Reinforcement Learning, Optimization

1. INTRODUCTION

Multi-agent learning algorithms provide a natural approach to addressing congestion problems in traffic and transportation domains [5, 10, 16]. 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 [15, 19], path selection in data routing [17], and side selection in the minority game [9, 13]. 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.

In this paper we apply multi-agent learning algorithms to two distinct formulations of the traffic problem. 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 utility is measured from the perspective of a "city manager", where minimizing system-wide delays is the desired goal. Then we investigate this problem from the perspective of the drivers who want to maximize their own "timeliness" functions. Based on those driver timeliness functions, we build a social welfare function to measure the system performance. Both formulations share the same underlying property that agents greedily pursuing their best interests cause traffic to worsen for everyone in the system, including themselves. However, the solution to alleviating congestion takes on a different form in each formulation, as the interaction between driver and system utilities have different characteristics in their respective formulations.

In both perspectives, the approach we present to alleviating congestion in traffic is based on assigning each driver an agent which determines the best departure time. Those agents determine their actions based on a reinforcement learning algorithm [18, 23]. The key issue in this approach is to ensure that the agents receive utilities that promote good system level behavior. To that end, it is imperative that the

agent utilities: (i) be aligned with the system utility[1], ensuring that when agents aim to maximize their own utility they also aim to maximize system utility; and (ii) be 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 utility functions of a particular agent).

The difficulty in agent utility selection stems from the fact that typically these two properties provide conflicting requirements. A utility that is aligned with the system utility 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 utility that is sensitive to the actions of one agent is likely not to be aligned with system utility. 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 [6, 21, 12, 20, 22, 24].

In this paper we show how agents using reinforcement learning can be used to alleviate traffic congestion, particularly when the utilities they are trying to maximize are derived carefully. In Section 2 we discuss the properties agent utilities need to have and present a particular example of an agent utility. In Section 3, we present the two system models, one viewing the problem from on a city manager's perspective and the other from the perspective of the individual drivers. In Section 4 we present the traffic model used in this paper. In Section 5, we present the results showing that not only do difference utilities provide good values of the city manager's utility, but having agents use a modified version of their intrinsic utilities rather than aim to optimize them directly leads them to achieve higher values of their own intrinsic utilities. Finally Section 6 we discuss the implication of these results and highlight experiments to investigate this problem further.

2. MULTIAGENT COORDINATION

In this work, we focus on multi-agent systems where each agent, i, tries to maximize its utility function $g_i(z)$, where z depends on the joint move of all agents. Furthermore, there is a system utility 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 Difference Utility Functions

Let us now focus on the **difference** utility [24] for shaping agent behavior:

$$D_i \equiv G(z) - G(z_{-i}) \quad (1)$$

where z_{-i} contains all the states on which agent i has no effect. In other words, all the components of z that are affected by agent i are removed (i.e., replaced with a domain specific "null" vector). This type of agent utility offers two advantages. First, D_i and G have the same partial derivative with respect to the actions of agent i, because the second term does not depend on the actions of that agent. In other words, difference utilities are perfectly aligned with the system utilities on which they are based [24]. Second, they usually have far better signal-to-noise properties than does a system utility function, because the second term of D removes some of the effect of other agents (i.e., noise) from i's utility function.

The difference utility can be applied to any linear or nonlinear system utility function. However, its effectiveness is dependent on the domain and the interaction among the agent utility functions. At best, it fully cancels the effect of all other agents. At worst, it reduces to the system utility function, unable to remove any terms (e.g., when z_{-i} is empty, meaning that agent i effects all states). Still, the difference utility often requires less compututation than the system utility function [25]. Indeed, for the problems presented in this paper, agent i can compute D_i using less information than required for G (see details in Section 3.1 and Section 3.2).

2.2 Utility Maximization

In this paper we assume that each agent maximizes its own utility using a reinforcement learner (though alternatives such as evolving neuro-controllers are also effective [1]. 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 paper only needs to utilize immediate utilities, therefore a simple table-based immediate reward reinforcement learning is used. Our reinforcement learner is equivalent to an ϵ-greedy Q-learner with a discount parameter of 0. At every episode an agent takes an action and then receives a reward (value of the immediate utility) evaluating that action. After taking action a and receiving reward R a driver updates its table as follows: $Q'(a) = (1 - \alpha)Q(a) + \alpha(R)$, 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. MEASURING SYSTEM PERFORMANCE

The first step in this investigation is determining well defined performance metrics that rate the different outcomes resulting from the drivers' actions. Though for some domains, there may be a single logical utility function (e.g., robots maximizing an exploration based utility), in the traffic domain, this step is far from straight-forward. In this study, we will focus on two distinct perspectives and provide results and insight for both.

First, one can use a *top-down* approach where a "city manager's utility" rates the performance of the system from an average congestion perspective, with little to no regard for individual drivers' preferences. Second, one can use a *bottom-up* approach where the drivers' intrinsic preferences are used to build a social welfare function and rate the per-

[1]We call the function rating the performance of the full system, "system utility" throughout this paper. We will specify "city manager's utility" or "Timeliness social welfare function" to distinguish between the two main system performance criteria.

[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., $z = z_i + z_{-i}$).

formance of the system in terms of that social welfare function, directly reflecting how satisfied the drivers are with the outcomes. Though one may reasonably expect that the city manager's utility and the social welfare function based on driver preferences will aim to lower congestion, there is no reason to assume that they will have the same optima, or promote similar behavior throughout the state space.

3.1 City Manager Perspective

Let us first discuss the case where the system performance is measured by a global utility representing the city manager's perspective. This *city manager's utility* is given by:

$$G_{CM} = \sum_{s_i} w_{s_i} S(k_{s_i}) \ . \tag{2}$$

where w_{s_i} are weights that model rush-hour scenarios where different time slots s_i have different desirabilities, and $S(k_{s_i})$ is a system throughput metric that depends on the number of agents that are in time slot s_i:[3]

$$S(k) = \begin{cases} ke^{-1} & \text{if } k \leq c \\ ce^{-k/c} & \text{otherwise} \end{cases} \ , \tag{3}$$

The number of drivers in the time slot is given by k, and the optimal capacity of the time slot is given by c. This functional form (shown in Figure 1, for a capacity of 30) abstracts the simple throughput concepts that below an optimal capacity value c, the throughput increases linearly with the number of drivers. When the number of drivers exceeds the optimal capacity level, the value of the time slot (throughput) decreases asymptotically exponentially with the number of drivers. This aspect reflects the precipitous decline in throughput when traffic begins to slow down due to congestion.

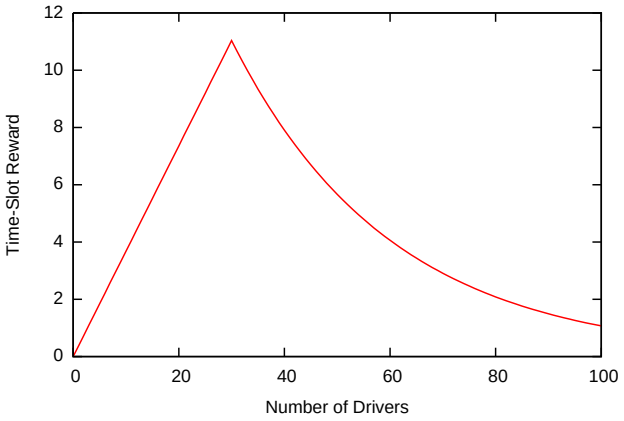

Figure 1: Utility of time slot with $c = 30$.

In this problem formulation, the task of the city manager is to have the agents choose time slots that maximize G_{CM}, the city manager's utility. To that end, agents have to balance the benefit of going at preferred time slots against possible congestion in those time slots.

While the utility of the city manager is to maximize the G_{CM}, the distributed nature of the problem means that each individual agent will try to maximize a driver-specific utility. The agents will maximize their utilities through reinforcement learning (as discussed in Section 2.2).

A key decision then is in determining the type of utility that each agent will receive. In this formulation, we focus on the following three agent utilities:

- The first utility is simply the system utility G_{CM}, where each agent tries to maximize the system utility directly.
- The second utility is a local utility, L_{CM_i} where each agent tries to maximize a utility based on the time slot it selected:

  $$L_{CM_i}(k) = w_i S(k_i) \ , \tag{4}$$

  where k_i is the number of drivers in the time slot chosen by driver i.
- The third utility is the difference utility, D_{CM}:

  $$\begin{aligned} D_{CM_i} &= G_{CM}(k) - G_{CM}(k_{-i}) \\ &= \sum_j w_j S(k_j) - \sum_j w_j S(k_{-i_j}) \ , \end{aligned}$$

  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. This formulation accounts for the fact that when there is a congestion, drivers remain in the system until they reach a time slot with a traffic level below the capacity. In this case, the actions of an agent have influence over multiple time slots, starting with the time slot in which they entered the system. Even with this "cascading" congestion model (see Section 4 for details), an agent will usually need to consider fewer time slots to compute D_{CM_i} than it would have needed to compute G_C.

3.2 Driver Timeliness Perspective

In the discussion above, the city manager's utility measured the health of the whole system, providing a global perspective from which to derive utilities for the individual agents. Though appealing in its formulation, this approach is less representative of traffic than a model in which the drivers' intrinsic preferences are the key factors in shaping system behavior. In this section, we focus on such an agent-centric approach, and introduce individual agent preferences that directly capture the desires of the drivers.

The system performance is then gauged by a social welfare function based on the driver preferences. With this function the individual agent utilities measure the agents' success at attaining personal arrival preference. We selected an exponential decay function to represent the agents' satisfaction with their time slots, resulting in the following *agent timeliness*, L_{AT_i}, functions:

$$L_{AT_i} = e^{\frac{-(a_i - t_i)^2}{b}} \tag{5}$$

where a_i and t_i are the actual and intended arrival slots for agent i, respectively, and b is a parameter that determines the steepness of the decline in utility away from the desired time slot.

After an agent picks some slot s_i, depending on the choices of other agents, it may find itself stuck in congestion, exiting from the route at a later arrival time (a_i) than intended. The

[3]In this work we assume the departure and arrival times are the same unless there is congestion in the system.

agent actually wants to arrive at its target time (t_i), so L_{AT_i} peaks when the agent arrives at its intended target time.

Unlike in the previous section where we started from a system wide perspective (city manager's utility), in this perspective, we start from intrinsic agent preferences. Using those agent timeliness functions, we construct a social welfare function (Equation 6) and use that to measure the system performance. The interesting question we address in this section is what utilities should the agent aim to optimize to also optimize the social welfare function. In this study, we focus on the following three agent utilities:

- The agent timeliness utility given by L_{AT_i}, where each agent tries to directly maximize its own timeliness function.
- The social welfare function based on the agent timeliness functions:

$$G_{AT}(z) = \sum_i L_{AT_i}(z_i) = \sum_i e^{\frac{-(a_i-t_i)^2}{b}} \quad (6)$$

  This function measures the performance of the system as the sum of the individual local agent utilities, so the system does best when all of its agents arrive in their desired target slots.
- The difference utility which in this formulation is derived from the social welfare function computed above is given by:

$$D_{AT_i}(z) = G_{AT}(z) - G_{AT}(z_{-i}) \quad (7)$$

  This utility computes an agent's impact on the system by estimating the gain to the system by the removal of that agent. The net effect of removing agent i from the system is in allowing other drivers to potentially exit the system rather than remain in a congested state. When the time slot chosen by agent i is not congested, agent i's removal does not impact any other agents. Using this reasoning, Equation 8 provides the difference between agent i's own utility and the utility the agents who remain in the system would have received had agent i not been in the system. In terms of the original agent timeliness utilities, this means that agent i's utility is penalized by the contributions of agents that it caused to be delayed.

  The computation of Equation 7 is problematic in practice due to the coupling between the agents actions and their impact of future time slots. To effectively measure this value, an estimate of each time slot following the agent's selected time slot would need to be computed. A more readily computable estimate for the difference utility is given by:

$$D_{AT_i}(z) \simeq e^{\frac{-(a_i-t_i)^2}{b}} - \sum_{j=s_i+1}^{a_i} e^{\frac{-(j-t_i)^2}{b}} \quad (8)$$

  This estimate provides good performance under both system utility functions, because it does accurately measure an agent's impact in states that are very close to the optimal state. The interesting aspect of this agent utility is that agents performed a conversion: $L_{AT_i} \rightarrow G_{AT} \rightarrow D_{AT_i}$, starting from a local utility, going to a social welfare function and then back to an agent utility based on that welfare function.

4. TRAFFIC PROPAGATION MODEL

In this work, we investigate two abstract different traffic propagation models. In both models used here, there is a fixed set of drivers, driving on a single route. The agents choose the time slot in which their drivers start their commutes. Agents that select a time slot that is congested stay on the road for future time slots, delaying their arrival. How the agents "cascade" from one time slot to another is the main difference between our two models.

4.1 Linear Cascading Congestion

In the linear cascading congestion model, the congestion does not affect the capacity of the time slot. Agents that exceed the capacity are simply cascaded into the next slot. They then combine with agents that selected that next slot to provide the total number of drivers on that time slot. More precisely, the number of drivers k_{s_i}, in time slot s_i is given by:

$$k_{s_i} = k_{s_i-1} - exit_{s_i}(k_{s_i-1}) + \sum_j I(s_i, j) \quad (9)$$

where k_{s_i-1} is the number of agents in the preceding time slot (s_1-1) and $I(s_i, j)$ is the indicator function that returns one if agent j selected time slot i and zero otherwise. In the linear system, $exit_{s_i}(k)$ returns the number of agents that exited the system at time slot i. Equation 10 reinforces that this is linear with the number of agents in slot s_i, up to peak capacity:

$$exit_{s_i}(k) = \begin{cases} k & \text{if } k \leq c \\ c & \text{otherwise} \end{cases} \quad (10)$$

For example, for a capacity of 250, if 300 agents are in time slot s_i, 50 agents will remain in the system for the next time slot s_{i+1}, in effect reducing the capacity of s_{i+1} by 50. The cascading process is repeated until agents have exited the system.

Each individual agent is equally likely to be cascaded from a congested slot. This uniform exit model does not account for the position of the agents, but it prevents artificial effects that arise when agents exit in an ordered manner.

4.2 Non-Linear Cascading Congestion

The cascading effects in the linear model can be further compounded by exponentially decreasing the number of agents that can exit the system in times of high congestion. Replacing Equation 10 in Equation 9, Equation 11 gives the number of drivers exiting the route at time slot i in the non-linear cascading traffic model:

$$exit_{s_i}(k) = \begin{cases} k & \text{if } k \leq c \\ ce^{-\frac{k-c}{c}} & \text{otherwise} \end{cases} \quad (11)$$

This equation states that up to c agents may exit the route at slot i, and the number of agents that may exit decreases exponentially once that capacity has been reached. The remaining drivers would remain in the system. The net effect of the nonlinear congestion model is in decreasing the effective capacity of a time slot based on the number of drivers who selected that time slot.

For example, if 300 drivers are on a given time slot with capacity 250, the capacity is reduced by 18%, resulting in an effective capacity of 205 and causing 95 drivers to be cascaded to the next time slot. In contrast, in the linear model, only 50 drivers would have been cascaded to the next time slot. This effect becomes more pronounced as congestion increases (for example for 500 drivers on the road, the capacity is reduced by 63%, to an effective capacity of only 91 and 409 drivers are cascaded to the next time slot).

This models behavior of traffic systems where congestion can appear almost instantly but then takes longer to disperse, causing longer travel times for agents that are in a congestion.

5. EXPERIMENT RESULTS

To test the effectiveness of the different agent utilities in promoting desirable system behavior, we performed experiments for both the city manager's utilities and the driver timeliness utilities. For the city manager's utility function, we explored both the linear and non-linear cascading models, but focused only on the more difficult non-linear cascading model for the agent timeliness utilities. In all the following figures, regardless of what utilities the agents used, the system performance is measured by the appropriate system utility (e.g., city manager's utility or social welfare function, as the case may be).

5.1 City Manager Results

In this set of experiments there were 1000 drivers, and the optimal capacity of each time slot was 250. Furthermore, the weighting vector was centered at the most desirable time slot (e.g., 5 PM departures):

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

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.

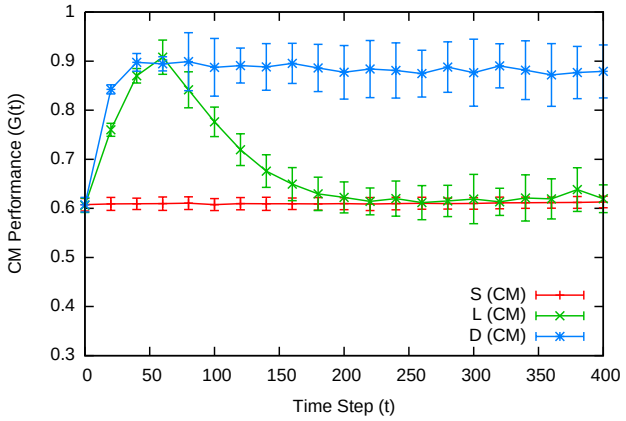

Figure 2: Performance on city manager's utility for linear cascading. In this model, drivers above the capacity in one time slot remain in system in future time slots, but do not affect the capacity of future time slots. Drivers using D quickly learn to achieve near optimal performance (1.0). After an initial peak, the performance of drivers using L degrades as drivers learn to be selfish.

5.1.1 *Linear Cascading Model*

This experiment shows that drivers using the difference utility are able to quickly obtain near-optimal system performance (see Figure 2). In contrast, drivers that try to directly maximize the system utility learn very slowly and never achieve good performance during the time-frame of the experiment. This slow learning rate is a result of the system utility having low signal-to-noise with respect to the agents' actions (an agent's action is masked by the "noise" of the other agents' actions). Even if a driver were to take a system wide coordinated action, it is likely that some of the 999 other drivers would take uncoordinated actions at the same time, lowering the value of the city manager's system utility. A driver using the system utility typically does not get proper credit assignment for its actions, since the utility is dominated by other drivers.

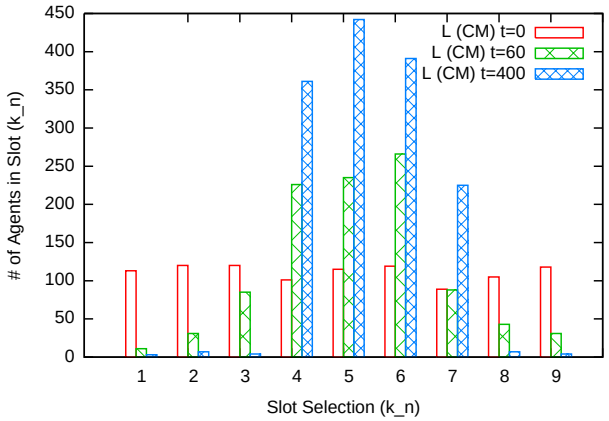

Figure 3: Distribution of drivers using local utility at different times in the experiment (for linear congestion model). Early in training drivers learn good policies. Later in learning, the maximization of local utility causes drivers to over utilize high valued time slots.

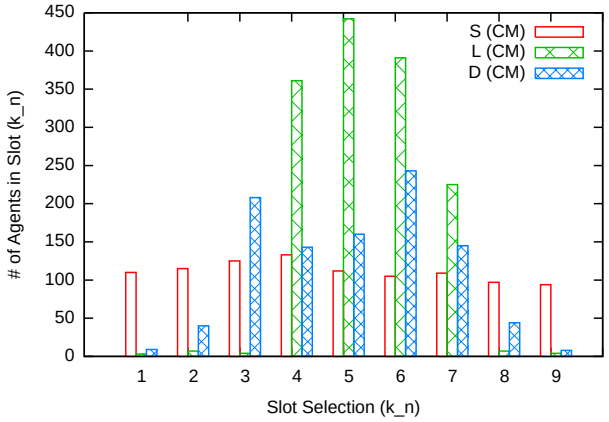

Figure 4: Distribution of drivers at end of experiment for linear congestion model. Drivers using difference utility (based on city manager's utility) form distribution that is closer to optimal than drivers using system of local utilities.

The experiment where drivers use L_{CM_i} (a non-aligned local utility) exhibits interesting performance properties. At first these drivers learn to improve the system utility. However, after about episode seventy their performance starts to decline. Figure 3 shows the change in the agent departure histogram as the agents learn, and provides greater insight into the reasons for the steep decline at $t = 70$. In the early phases of learning, the drivers are randomly distributed among time slots, resulting in a low utility. Later in training (t=70) agents begin to learn to use the time slots that have the most benefit. When the number of drivers reach near optimal values for those time slots, the system utility 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 which leads to a drop in the city manager's utility. This performance characteristic is typical in systems with poorly aligned agent utilities; in such a case, agents attempting to maximize their own utilities lead to undesirable system behavior.

Figure 4 shows the histograms of all three utilities at the end of the runs. The agents using the city manager's utility directly perform near randomly (confirming the results of Figure 2). In contrast, because their utilities are aligned with the system utility, agents using the difference utility form a distribution that nearly matches the optimal distribution proving that not only is having agents attempt to maximize D_{CM_i} is good for the system, but that the agents receive good feedback to actually maximize this function, and consequently the city manager's utility.

5.1.2 *Non-linear Cascading Model*

This experiment investigates the case where a congestion clears more slowly, with drivers exceeding capacity causing exponential harm to the system. Coordination becomes more critical in this case as even a small congestion can cascade into causing significant delays, a pattern that more closely matches real traffic patterns. Figure 5 shows the performance of the three utilities in the non-linear cascading model. Neither the performance of agents directly using the city manager's utility, nor the performance of agents using local utilities is impacted in this case, since they did not perform well to begin with. The performance of agents using the difference utility is still consistently good in this significantly more difficult problem.

The attendance profiles in Figure 6 show the arrival slots of the drivers at the end of the simulation. The increase in late time slots (e.g., slots 7 and 8) show the reasons for the drop in performance in this model when compared to the linear model. Because drivers take longer to get through the system, there is rightward shift in arrivals, causing the attendance profile to differ from the optimal profile.

5.2 Agent Timeliness Experiment Results

Continuing with the non-linear congestion model, let us focus on the alternative perspective of having the desirability of time slots be directly derived from the intrinsic utilities of the agents. In these experiments, all 1000 agents aimed to leave and arrive in the middle slot ($s_i = 5$). The interesting question in this setting is whether agents considering their preferences learn to share the road better than agents that aim to account for the road capacities and congestion in their decision.

Figure 7 shows the performance of the local agent time-

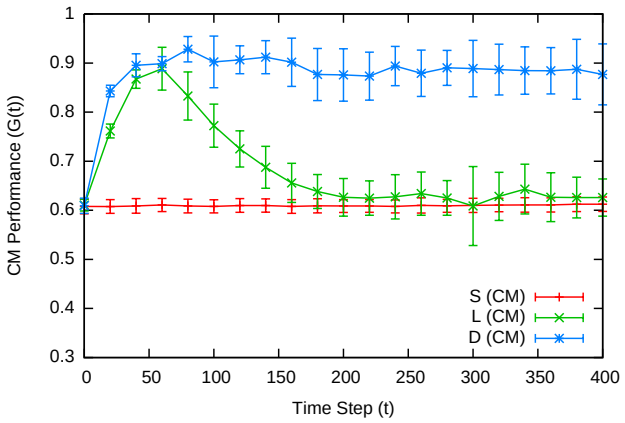

Figure 5: Performance on city manager's utility for nonlinear cascading. In this model, drivers above the capacity in one time slot both remain in system in future time slots and reduce the capacity of those time slots. Drivers using difference utilities quickly learn to achieve near optimal performance (1.0).

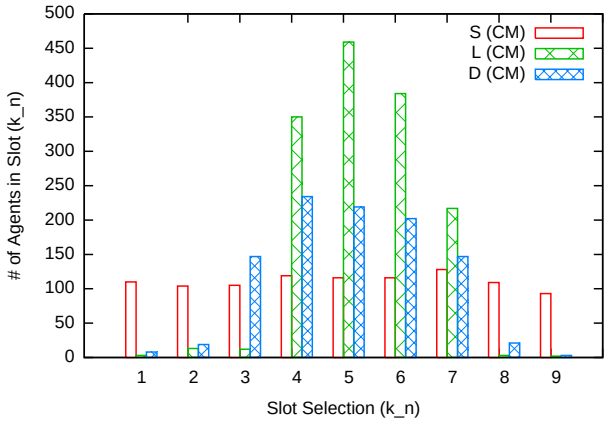

Figure 6: Distribution of drivers at end of experiment for nonlinear congestion model. Drivers using difference utility (based on city manager's utility) form distribution that is closer to optimal than drivers using system of local utilities.

liness, social welfare, and difference utilities in this new model. All performance is measured by the social welfare function. Agents using the social welfare function directly are unable to improve their performance, for reasons discussed in 5.1.1. The local agents are able to learn to obtain modest improvements in their overall utility, while agents using the difference utility reach near optimal performance.[4]

The counterintuitive result is that average agent utility improves when agent drivers do not follow their local timeliness utilities. Because of the non-linear interactions between slot congestion and arrival rates, agent drivers that selfishly pursue their own utility cause congestion while those agents that consider their impact on others arrive on-time more frequently. Figure 8 shows the attendance of drivers in each slot at the end of the simulation ($t = 400$) when using each

[4]In the linear traffic model, drivers using the local utility were also able to obtain nearly optimal performance.

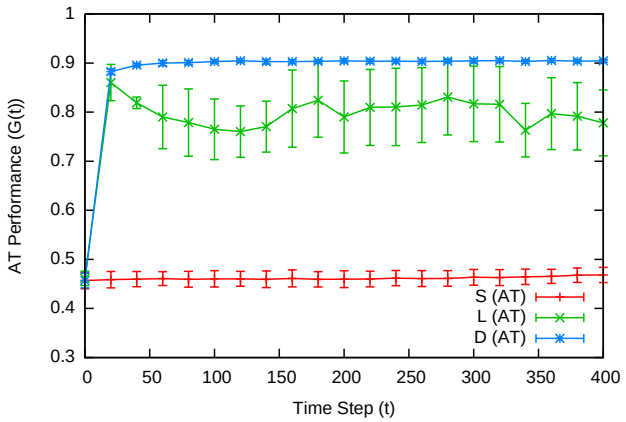

Figure 7: Performance on agent timeliness social welfare function for nonlinear cascading. Agents using their intrinsic preferences perform adequately (.75-.8), but agents using difference utilities achieve better performance (0.9).

of the agent timeliness utilities. Agents using the timeliness difference utility manage to keep the critical slots remarkably free of congestion, excepting noise from ongoing ϵ-greedy selection.

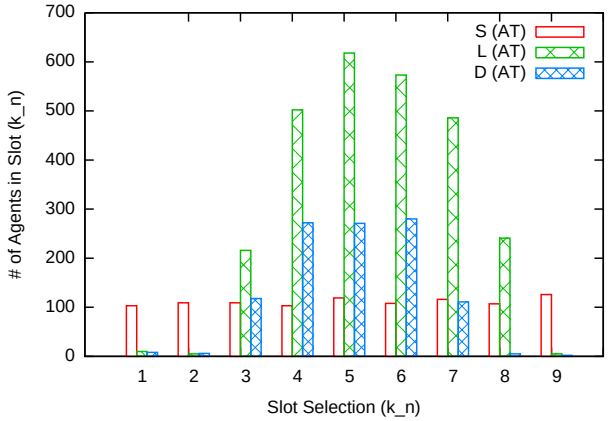

Figure 8: Distribution of drivers at end of experiment for nonlinear congestion model. Drivers using difference utility (based on social welfare function) form distribution that is closer to optimal than drivers using system of local utilities.

5.3 Agent/System Utility Mismatch

In this experiment we measure the performance of the utilities reported in Section 5.2 using the city manager's utility. It is crucial to emphasize that none of the three utilities used by the agents during training aimed to directly optimize the utility by which we measure the system performance (city manager's utility). We are investigating this utility mismatch as it is critically relevant to many applications where - though there may be a system-wide metric (e.g., city manager's utility) - the individuals in the system use different criteria to choose their actions (e.g., their own arrival times).

Figure 9 shows the performance of the agent timeliness utilities, social welfare functions and difference utilities based on the social welfare function on the *city manager's utility.*

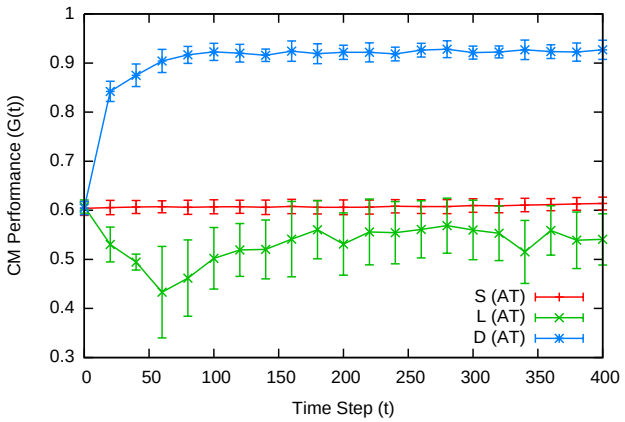

Figure 9: Performance with respect to city manager's utility of agents trained using agent timeliness (local, difference and social welfare) utilities (for nonlinear cascading). Though none of the agents were aiming to maximize the city manager's utility (directly or indirectly), the system behavior promoted by the difference utility provided high values of the city manager's utility. Agents pursuing their own preferences lead the system to perform worse than random, as agents aiming to arrive one time frustrate each other and cause disastrous congestion.

Agents using the social welfare function directly are again unable to improve their performance, and agents using the their own intrinsic timeliness harm the city manager's utility. Their performance can be explained by revisiting the arrival profile for the local timeliness utility in Figure 8; most agent drivers select a slot close to their target and create heavy congestion in the highest weighted slot.

Agents using the difference utility learn to perform well from the city manager's perspective, and the agent arrival profile in Figure 8 demonstrates how this performance is achieved. The agent drivers are able to learn how their actions impact the system by estimating their impact on the system, resulting in less congestion and overall better performance for the city manager.

6. DISCUSSION

This paper presented a method for improving congestion in two different traffic problems. First we presented a *top-down* 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 presented a *bottom-up* method for improving social welfare when drivers use an estimated difference utility instead of their own timeliness utility. This is an interesting result that states that agents *on average* reach higher values of their own intrinsic utilities if they do not aim to maximize it directly, but rather aim to optimize a modified version of that utility.

These results are based on agents receiving utilities that are both aligned with the system utility and are as sensitive as possible to changes in the utility of each agent. In these experiments, agents using difference utilities produced near optimal performance (93-96% of optimal). Agents using system utilities (63-68%) performed comparably to random action selection (62-64%), and agents using local utilities (48-

72%) provided performance ranging from mediocre to worse than random in some instances, when their own interests did not align with the social welfare function.

In addition to their good performance, difference utilities also provided consistency and stability to the system. Because regardless of the system utility on which it is based, the difference utility aims to remain aligned with that utility, it promotes beneficial system-wide behavior in general. In the traffic domain, the city manager's utility and the social welfare function based on the agent timeliness were aiming to promote the concept of a "good traffic pattern". It is therefore not totally surprising that a utility aiming to maximize one utility performed well on the other. However, it is worth nothing that neither the agents directly aiming to optimize the social welfare function, nor the agents aiming to maximize their own preferences achieved this result. An interesting note here is that the city manager's utility is more concerned with congestion, whereas the agents are more concerned with delays (arriving on time).

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. In this work, we did not address this issue, as our purpose 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. On one hand, drivers will notice that the departure times/routes 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.

7. REFERENCES

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

[2] M. Balmer, N. Cetin, K. Nagel, and B. Raney. 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, July 2004.

[3] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, 51(2):1035–1042, 1995.

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

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

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

[7] B. Burmeister, A. Haddadi, and G. Matylis. Application of multi-agent systems in traffic and transportation. *IEEE Proceedings in Software Engineering*, 144(1):51–60, 1997.

[8] C. R. Carter and N. R. Jennings. Social responsibility and supply chain relashipships. *Transportation Research Part E*, 38:37–52, 2002.

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

[10] K. Dresner and P. Stone. 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, July 2004.

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

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

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

[14] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, 1998.

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

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

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

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

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

[20] D. C. Parkes. *Iterative Combinatorial Auctions: Theory and Practice*. PhD thesis, University of Pennsylvania, 2001.

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

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

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

[24] K. Tumer and D. Wolpert. A survey of collectives. In *Collectives and the Design of Complex Systems*, pages 1,42. Springer, 2004.

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