Regulating Air Traffic Flow with Coupled Agents

Adrian Agogino
UCSC, NASA Ames Research Center
Mailstop 269-3
Moffett Field, CA 94035, USA
Adrian.K.Agogino@nasa.gov

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

ABSTRACT

The ability to provide flexible, automated management of air traffic is critical to meeting the ever increasing needs of the next generation air transportation systems. This problem is particularly complex as it requires the integration of many factors including, updated information (e.g., changing weather info), conflicting priorities (e.g., different airlines), limited resources (e.g., air traffic controllers) and very heavy traffic volume (e.g., over 40,000 daily flights over the US airspace). Furthermore, because the Federal Flight Administration will not accept black-box solutions, algorithmic improvements need to be consistent with current operating practices and provide explanations for each new decision. Unfortunately current methods provide neither flexibility for future upgrades, nor high enough performance in complex coupled air traffic flow problems.

This paper extends agent-based methods for controlling air traffic flow to more realistic domains that have coupled flow patterns and need to be controlled through a variety of mechanisms. First, we explore an agent control structure that allows agents to control air traffic flow through one of three mechanisms (miles in trail, ground delays and rerouting). Second, we explore a new agent learning algorithm that can efficiently handle coupled flow patterns. We then test this agent solution on a series of congestion problems, showing that it is flexible enough to achieve high performance with different control mechanisms. In addition the results show that the new solution is able to achieve up to a 20% increase in performance over previous methods that did not account for the agent coupling.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance

Keywords

Air Traffic Control, Multiagent Systems, Optimization

Cite as: Regulating Air Traffic Flow with Coupled Agents, Adrian Agogino and Kagan Tumer, *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. 535-542.

1. INTRODUCTION

The ability to effectively control air traffic with traditional control/optimization algorithms is decreasing rapidly as the air traffic levels and aircraft heterogeneity increase and restrictions on flight plans decrease. New strategies are needed to cope with this added complexity and to provide robust safety levels while ensuring that air traffic delays do not reach unacceptable levels. Indeed, even at current air traffic levels, in 2005 alone there were an estimated 322,272 hours of delays within the United States airspace with a total cost estimated to exceed three billion dollars by industry [7]. With an expected increase in air traffic, unless the air traffic management processes are overhauled, these delays are expected to become significantly worse. The Next Generation Air Transportation Systems (NGATS) initiative is designed to address future issues in air traffic management without requiring major infrastructure changes (e.g., airports, runways, and towers) or adding large numbers of additional air traffic controllers. To accomplish this, new robust algorithms are needed that can safely manage complex air traffic flows while making optimal use of current infrastructure.

Controlling air traffic flow is a complex task that involves multiple controls, including ground delays, airplane separation and rerouting [15, 16]. It is also complicated by the fact that while typically the airspace as a whole is loosely coupled, at certain times coupling is tight and has to be taken into account. To tackle this difficult problem we propose to extend the multi-agent solution introduced in [16]. While this previous approach provided good solutions, it only handled one mode of control (separation) and assumed that there was little coupling between agents[1]. The contributions of this paper are to extend those results in three directions, and to investigate:

- the impact of two new agent actions: ground delays and re-routes (in addition to setting aircraft separation);
- the impact of coupling between agent actions; and
- the benefits of estimating agent rewards using pre-computed values.

Other agent-based work on the air traffic problem includes multi-agent learning, satisficing utilities, agent negotiation and mechanism design approaches to lead agents to reach

[1]This previous work assumed no coupling in terms of airflow dynamics, but did assume an overall coupling in terms of reward. The work presented here handles coupling in both.

a global satisfactory goal [18, 10, 12, 8]. Additionally there have been numerous traditional modeling approaches to this problem [11, 13, 9]. We seek to extend ideas from both communities to produce a compelling multi-agent solution.

This paper shows how these extensions allow the multi-agent air traffic management algorithms both to achieve higher performance and to be applicable to more realistic environments. In Section 2, we describe the air traffic flow problem and the simulation tool, FACET. In Section 3, we present the agent-based approach, focusing on the agents' action space, learning algorithms and reward structures. In Section 4 we present results in multiple domains with different types of congestions and different agent actions. Finally, in Section 5, we discuss the implications of these results.

2. COUPLED AIR TRAFFIC DOMAIN

Large numbers of flights, unpredictable weather, complex route patterns, concerns for safety and fairness are just some of the issues that make controlling traffic flow management a demanding problem. Automated air traffic methods not only have to create solutions for all these issues on a system with 5000 flights an hour in the United States alone, they have to provide solutions that are trustworthy and palatable to the human operators controlling the airspace [13].

Fortunately the current operating assumptions that traffic is mostly uncoupled (both in air traffic controller actions and in current agent based solutions) is acceptable in most cases. While most flights have local interactions, they do not interact across the airspace (e.g. flights on the East Coast have little affect on flight on the West Coast), making the current air traffic system workable. However, there are some significant interactions that cannot be assumed away (e.g. intersecting jet routes) and accounting for such coupling is critical in improving upon the current approaches to air traffic management.

2.1 Airspace Configuration

The airspace in the US is decomposed into a hierarchy that has both decentralized and centralized components. At a high level the airspace consists of 20 regional centers (handling 200-300 flights per hour) and at a lower level 830 sectors (handling 10-40 flights per hour). The flow control problem has to address the integration of policies across these sectors and centers, account for the complexity of the system (e.g., over 5200 public use airports and 16,000 air traffic controllers) and handle changes to the policies caused by weather patterns. Two of the fundamental issues in addressing the flow problem are: (i) modeling and simulating such a large complex system as the fidelity required to provide reliable results is difficult to achieve; and (ii) establishing the method by which the flow management is evaluated. In the following sections we address both issues: we describe the air traffic simulator FACET, and present our system evaluation function based on both system congestion and total traffic delay.

2.2 FACET Simulator

FACET (Future ATM Concepts Evaluation Tool), is a modeling tool that accurately models the complex air traffic flow problem over the US airspace [4]. It is based on propagating the trajectories of proposed flights forward in time. FACET can be used to either simulate and display air traffic (a 24 hour slice with 60,000 flights takes 15 minutes to simulate on a 3 GHz, 1 GB RAM computer) or provide rapid statistics on recorded data (4D trajectories for 10,000 flights including sectors, airports, and fix statistics in 10 seconds on the same computer) [1]. FACET is extensively used by the FAA, NASA and industry (over 40 organizations and 5000 users) [1].

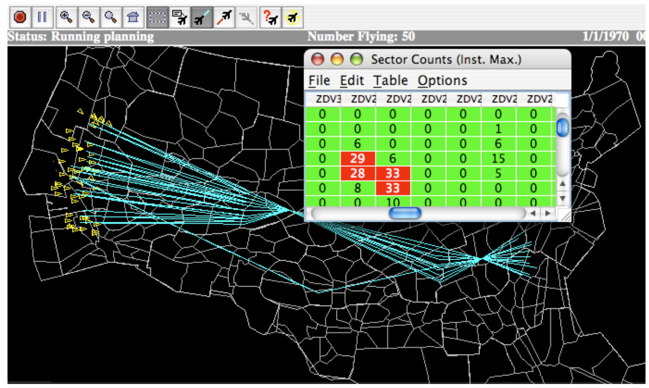

Figure 1: FACET screenshot displaying traffic routes and air flow statistics.

FACET simulates air traffic based on flight plans and through a graphical user interface it allows the user to analyze congestion patterns of different sectors and centers (Figure 1). FACET also allows the user to change the flow patterns of the aircraft through a number of mechanisms. The user can then observe the effects of these changes to congestion. In this paper, agents use FACET directly through "batch mode", where agents send scripts to FACET asking it to simulate air traffic based on metering, ground delay or rerouting orders imposed by the agents. The agents then produce their rewards based on received feedback from FACET about the impact of these actions.

2.3 Evaluating Congestion and Delay

In air traffic management there are many different types of delays and congestions along with many ways of evaluating them. The best measure is very subjective and depends on the goals of the evaluator. In this paper we focus on a system-metric based both on the congestion in a particular set of sectors and on the measured air traffic delay. The linear combination of these two terms gives the full system evaluation function, $G(z)$, as a function of the full system state z. More precisely, we have:

$$G(z) = -((1-\alpha)B(z) + \alpha C(z)) \ , \qquad (1)$$

where $B(z)$ is the total delay penalty for all aircraft in the system, and $C(z)$ is the total congestion penalty.

The total delay, B, is a sum of delays over a set of sectors S and is given by:

$$B(z) = \sum_{s \in S} B_s(z) \ . \qquad (2)$$

where $B_s(z)$ is the delay for sector s which is evaluated differently depending on how air traffic is being manipulated. When air traffic is not being rerouted around a delay the following measure is used:

$$B_s(z) = \sum_t t(k_{t,s} - k^b_{t,s}) \ , \qquad (3)$$

where $k_{s,t}$ is the count of the number of aircraft in sector s at time t and $k^b_{t,s}$ the count for the baseline case where there are no agent controls. For methods that reroute aircraft around congestions instead of delaying their arrival at a congestion, we have a different penalty:

$$B^{rr}_s(z) = p \sum_t (k^b_{t,s} - k_{t,s}) \;, \tag{4}$$

where p is a fixed penalty for the reroute.

Similarly, the total congestion penalty is a sum over the congestion penalties over the sectors of observation, S:

$$C(z) = \sum_{s \in S} C_s(z) \tag{5}$$

where

$$C_s(z) = a \sum_t \Theta(k_{s,t} - c_s) e^{b(k_{s,t} - c_s)} \;, \tag{6}$$

where $\Theta(x) = 1$ when $x > 0$ and 0 otherwise, a and b are normalizing constants, and c_s is the capacity of sector s as set by the FAA. Intuitively, $C_s(z)$ penalizes a system state where the number of aircraft in a sector exceeds the FAA's official sector capacity. Each sector capacity is computed using metrics which include the number of air traffic controllers available. The exponential penalty is intended to provide strong feedback to return the number of aircraft in a sector to below the FAA mandated capacities.

3. AGENTS IN COUPLED AIR TRAFFIC

In this paper we present a distributed multi-agent solution to air traffic flow where adaptive agents take actions independently, but are coupled through their common system evaluation function discussed above. The definition of the "agents", determining the agent actions, selecting the agent learning algorithms, and selecting the agent reward structures are critical design decisions [16]. In this section we summarize our choices for each of these decisions.

3.1 Agent Selection and Action Space

There are many complex tradeoffs in selecting agents in the air traffic flow domain as discussed in [16]. In this work we assign agents to individual ground locations throughout the airspace called "fixes." Each agent is then responsible for any aircraft going through its fix. Fixes offer many advantages as agents:

1. Their number can vary depending on need. The system can have as many agents as required for a given situation(e.g., agents coming "live" around an area with developing weather conditions).

2. Because fixes are stationary, collecting data and matching behavior to reward is easier.

3. Because aircraft flight plans consist of fixes, agent will have the ability to affect traffic flow patterns.

4. They can be deployed within the current air traffic routing procedures, and can be used as tools to help air traffic controllers rather than compete with or replace them.

Based on this definition of an agent, we explore three methods for the agent based fixes to control the flow. Allowing agents to have the flexibility to control aircraft in multiple ways is essential to their ability to be integrated into existing systems. Even if all the methods work relatively well, an organization or a sector controller may only be comfortable with a particular form of flow control. Agents that are not flexible enough to conform to these needs will not be used. The methods used in this paper are as follows:

1. **Miles in Trail (MIT)**: Agents control the distance aircraft have to keep from each other wile approaching a fix. With a higher MIT value, fewer aircraft will be able to go through a particular fix during congested periods, because aircraft will be slowing down to keep their spacing. Therefore setting high MIT values can be used to reduce congestion downstream of a fix.

2. **Ground Delays**: An agent controls how long aircraft that will eventually go through a fix should wait on the ground. Imposing a ground delay will cause aircraft to arrive at a fix later. With this action congestion can be reduced if some agents choose ground delays and others do not, as this will spread out the congestion. However, note that if all the agents choose the same ground delay, then the congestion will simply happen at a later moment in time.

3. **Rerouting**: An agent controls the routes of aircraft going through its fix, by diverting them to take other routes that will (in principle) avoid the congestion.

Note that all of these control methods can result in some degree of coupling between agents when aircraft go through fixes associated with multiple agents. For instance if one agent enforces an MIT, the impact of agents down stream setting MITs may be reduced. Ground delays can be coupled if multiple agents are ordering the same aircraft to be delayed. Finally agents performing reroutes can be highly coupled if a series of reroutes create a new congestion.

3.2 Agent Learning

The objective of each agent is to select the action that leads to the best system performance, G (given in Equation 1). Each agent will have its own reward function and will aim to maximize that reward using a reinforcement learning algorithm [14] (though alternatives such as evolving neuro-controllers are also effective [2]). For delayed-reward problems, sophisticated reinforcement learning systems such as temporal difference may have to be used. However, due to our agent selection and agent action set, the air traffic congestion domain modeled in this paper only needs to utilize immediate rewards. As a consequence, a simple table-based immediate reward reinforcement learning is used. Our reinforcement learner is equivalent to an ϵ-greedy Q-learner with a discounting parameter of 0 [14]. At every episode an agent takes an action and then receives a reward evaluating that action. After taking action a and receiving reward R an agent updates its Q table (which contains its estimate of the value for taking that action [14]) as follows:

$$Q'(a) = (1 - \lambda)Q(a) + (\lambda)R, \tag{7}$$

where λ is the learning rate. At every time step, the agent 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 this paper, λ is equal to 0.5 and ϵ is equal to 0.25. The parameters were chosen experimentally, though system performance was not overly sensitive to these parameters.

3.3 Difference Rewards

The first and most direct approach to evaluating agent performance is to let each agent receive the system performance as its reward. However, in many domains such a reward leads to agents learning slowly at best, and at worst, not learning at all. We will therefore also set up a second set of rewards based on the impact of an agent on system performance. Ultimately, we desire agents aiming to optimize their own rewards to also optimize a system performance criteria. In this work we focus on **difference rewards** which aim to provide a reward that is both sensitive to that agent's actions and aligned with the overall system reward [3, 17]:

$$D_i \equiv G(z) - G(z - z_i + c_i) \ , \tag{8}$$

where z_i is the action of agent i. All the components of z that are affected by agent i are replaced with the fixed constant c_i [2].

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 to evaluate the performance of the system without agent i and therefore D evaluates the agent's contribution to the system performance. There are two advantages to using D: First, because the second term removes a significant portion of the impact of other agents in the system, it provides an agent with a "cleaner" signal than G. Second, it remains aligned with G.

3.4 Estimated Difference Rewards

Though providing a good compromise between aiming for system performance and removing the impact of other agents from an agent's reward, one issue that may plague D is computational cost. Because it relies on the computation of the counterfactual term $G(z - z_i + c_i)$ (i.e., the system performance without agent i) it may be difficult or impossible to compute, particularly when the exact mathematical form of G is not known. Let us focus on G functions in the following form:

$$G(z) = G_f(f(z)), \tag{9}$$

where $G_f()$ is non-linear with a known functional form and,

$$f(z) = \sum_i f_i(z_i) \ , \tag{10}$$

where each f_i is an unknown non-linear function. We assume that we can sample values from $f(z)$, enabling us to compute G, but that we cannot sample from each $f_i(z_i)$. In addition, we assume that G_f is much easier to compute than $f(z)$, or that we may not be able to even compute $f(z)$ directly and must sample it from a "black box" computation.

Previously it has been shown that given this form of G that D can be computed as follows [16]:

$$D_i^{ind-est} = G_f(f(z)) - G_f(f(z) - E(f(z)|z_i) + E(f(z)|c_i)) \ ,$$

where $E(f(z)|z_i)$ is the expectation of $f(z)$ leaving the value of z_i intact, and $E(f(z)|c_i)$ is the expectation of $f(z)$ given that the value of z_i is changed to c_i. We are now left with the task of estimating the values of $E(f(z)|z_i)$ and $E(f(z)|c_i))$. These estimates can be computed by keeping a table of averages where we average the values of the observed $f(z)$ for each value of z_i that we have seen. This estimate improves as the number of samples increases [16].

3.5 Pre-Computed Difference Rewards

In addition to the estimates to the difference rewards, it is possible to pre-compute certain values that can be later used by the agents. In particular, $-E(f(z)|z_i) + E(f(z)|c_i)$ can be computed exactly if the function f can be interrogated for certain values of z:

$$\begin{aligned}
& -E(f(z)|z_i) + E(f(z)|c_i) \\
= \ & -E(f_{-i}(z_{-i})) - E(f_i(z_i)|z_i) \\
& \quad +E(f_{-i}(z_{-i})) + E(f_i(z_i)|c_i) \\
= \ & -E(f_i(z_i)|z_i) + E(f_i(z_i)|c_i) \\
= \ & -f_i(z_i) + f_i(c_i) \\
= \ & -f_{-i}(k_{-i}) - f_i(z_i) + f_{-i}(k_{-i}) + f_i(c_i) \\
= \ & -f(k_{-i} + z_i) + f(k_{-i} + c_i) \ ,
\end{aligned}$$

where k_{-i} is a constant set of actions for all the agents other than i and f_{-i} is equal to $\sum_{j \neq i} f_j(z_j)$. Then by precomputing $f(k_{-i}+z_i)$ for each action for each agent, the value of D can be computed exactly. With n agents and m possible actions this requires

$$N_{pre} = 1 + m * n \tag{11}$$

computations of f. We refer to this reward as: D_i^{ind}. This approach provides a computationally expensive option for the case where forming estimates to compute D may not provide accurate results (e.g., some assumptions do not hold).

3.6 Coupled Group Difference Rewards

Unfortunately the above estimation for D is not valid if the values of $f_i(z_i)$ are coupled. However this problem can be resolved if the agents can be decomposed into a set disjoint groups that are uncoupled with respect to each other. We do this by replacing the vector z with a vector x where each component x_j represents the possible actions from the set of agents in group j. Here the number of possible values of x_j grows combinatorialy with the number of agents in the group and the number of actions each agent can take. With l agents in a group and m possible actions for each agent the number of possible values for x_j is m^l. With the space x, the same analysis that was performed in Sections 3.4 and 3.5 can be repeated with:

$$f(z) = h(x) = \sum_j h_j(x_j) \ ,$$

where the function h performs similar computations as f except that it is a function of group actions instead of agent actions. Then D can be estimated as:

$$\begin{aligned}
& D_i^{group-est} = \\
& \quad G_f(h(x)) - G_f(h(x) - E(h(x)|x_j) + E(h(z)|c_{i,j})) \ ,
\end{aligned} \tag{12}$$

where $c_{i,j}$ is the group action x_j with the value of z_i changed to c_i with all the other members of the group held constant. Note again that the estimate can be made exact by pre-computing certain values of h. However, in this case the

[2]This notation uses zero padding and vector addition rather than concatenation to form full state vectors from partial state vectors. The vector "z_i" in our notation would be $z_i e_i$ in standard vector notation, where e_i is a vector with a value of 1 in the ith component and is zero everywhere else.

number of possible actions for each group action x_j may be quite large. We refer to this reward as: D_i^{group}.

3.7 Reward Analysis and Summary

The air traffic domain matches well with the coupled group model, in that the assumptions that lead to the computation of the group difference rewards are generally accurate. In this domain agents along the same jet route tend to be coupled, while agents along different jet routes tend not to be. Therefore agents tend to naturally form groups where agents are coupled within the group, but where the groups are uncoupled with each other. Each group affects the number of aircraft in a sector, which can be represented as $h_j(x_j)$. However, we just know the total aircraft count for a sector, which is a sum of the independent paths that go through a sector: $h(x)$. In addition once we know the counts for a sector we can compute the congestion reward G_f which is of known form given the airplane counts.

In summary, we will investigate the performance of the following four rewards:

1. The estimated difference reward $D_i^{ind-est}(z)$, assuming no coupling between agents.
2. The difference reward $D_i^{ind}(z)$ using precomputed counterfactuals, assuming no coupling between agents.
3. The estimated group difference reward $D_i^{group-est}(x)$, where agents estimate the group counterfactual using $E(f(x)|x_j)$ and $E(f(x)|c_{i,j})$.
4. The group difference reward $D_i^{group}(x)$ using precomputed counterfactuals, assuming agents within a group are coupled.

Note that we show the first two rewards as a function of agent actions z and the next two as a function of group actions x for clarity, even though there is a one-to-one mapping between the two representations.

4. SIMULATION RESULTS

In this paper we test the performance of our agent based air traffic optimization method on a series of simulations using the FACET air traffic simulator. In all experiments we test the performance of six different methods. The first method is Monte Carlo estimation, where random policies are created, with the best policy being chosen. The second method is agents directly using the system reward, G, as defined in Equation 1. The next four methods use the different formulations/estimates of the difference reward summarized above in Section 3.7.

Agents using these rewards are tested using the following actions:

1. Miles in Trail: Agents maintain separation distance between aircraft going through their fix.
2. Ground Delay: Agents delay aircraft on the ground that will go through their fix.
3. Rerouting: Agents reroute aircraft going through their fix.

We test all six methods and the three actions in a scenario that consists of two independent congestions with a total of 300 aircraft over the course of five hours of flight time. The first congestion is relatively light and has a total of 75 aircraft. The main goal of agents in this congestion is to minimize delay. The second congestion is heavy and has a total of 225 aircraft. Here agents have to make firm actions to minimize the congestion. In this scenario, the agents are mostly in pairwise couplings, as most aircraft travel through two fixes before they enter a congested area. In addition to this scenario, agents using rerouting as their action are tested in two more congestion scenarios designed to test heavy coupling, as described later.

In all experiments the parameter for the tradeoff between congestion and lateness is set to $\alpha = 0.5$ and the normalizing constants for the congestion term are set to $a = 50$ and $b = 0.3$. For rerouting problems the reroute penalty p is set to one hour. These parameters are setup so that congestion and lateness have approximately the same impact. Note that the absolute performance between experiments with different actions is not comparable because of the different methods used to evaluate the penalties. All results are based on 30 runs and though they are plotted, the error bars are in most cases smaller than the symbols used to distinguish the rewards.

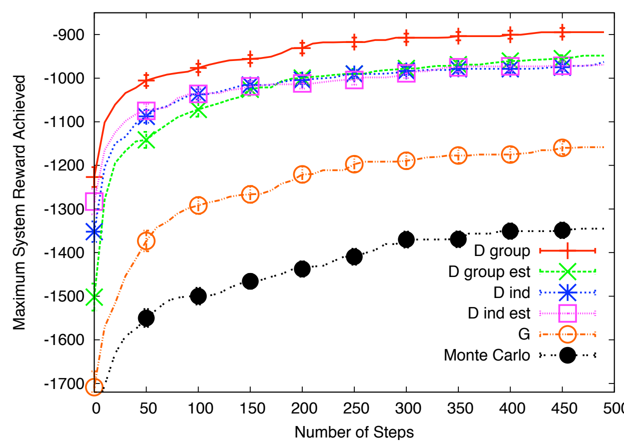

Figure 2: Performance for agents controlling miles in trail on dual independent congestion problem. 300 Aircraft, 40 Agents.

4.1 Controlling Miles in Trail

In our first set of experiments, agents control Miles in Trail (MIT): the distance aircraft have to separate themselves from each other when approaching a fix. Here agents choose between the three actions of setting the MIT to 0, 25 and 50 miles. Setting the MIT to 0 produces no effect, while setting it to high values forces the aircraft to slow down to keep their separation distance. Therefore setting high MIT values upstream of a congestion can alleviate a congestion, at the cost the increased delay.

The results shown in Figure 2 illustrate the benefit of using difference rewards. While agents directly optimizing G perform better than a Monte Carlo system, agents using any of the difference rewards perform considerable better. In addition agents using D^{group} perform the best, as they are able to best handle the coupling in the system. While performing well, agents using the independent D rewards do not account for the coupling are not able to converge to as good of solution since their computation of D is inexact and not fully aligned with G.

Interestingly while agents using $D_i^{group-est}(x)$ start out learning more slowly, they reach high performance in the end. These rewards are trying to estimate D taking coupling into account, by estimating "$E(h(x)|x_j) + E(h(z)|c_{i,j})$" and keeping a table entry for each possible group action x_j. In contrast $D_i^{ind-est}(z)$ only keeps table values for each agent's possible individual action z_i. With two agents per group, there are three individual agent actions and nine group actions. Agents taking groups into account therefore have to fill in a larger table, which makes them need more samples to produce an accurate estimate. While this causes them to learn more slowly, ultimately they can perform better since they are taking coupling into account.

4.2 Controlling Ground Delays

In the second set of experiments, agents control aircraft through ground delays. Here an agent can order aircraft that are scheduled to go through its fix to be delayed on the ground. In this scenario agents choose between one of three actions: no delay, 2 hour delay and 4 hour delay. Note that the dynamics of ground delays are quite different than with MITs since if all the agents choose the same ground delay, the congestion will still happen, just at a later time. Instead agents have to form the correct pattern of ground delays. Also in this system, agents are inherently coupled since more than one agent can order the ground delay of the same aircraft. In this situation we have the aircraft choose the smallest delay. Here coupling needs to be accounted for since even if an agen's action is subsumed by an action of another agent, it still forms a mapping from its own action (which was disregarded) and the reward it received.

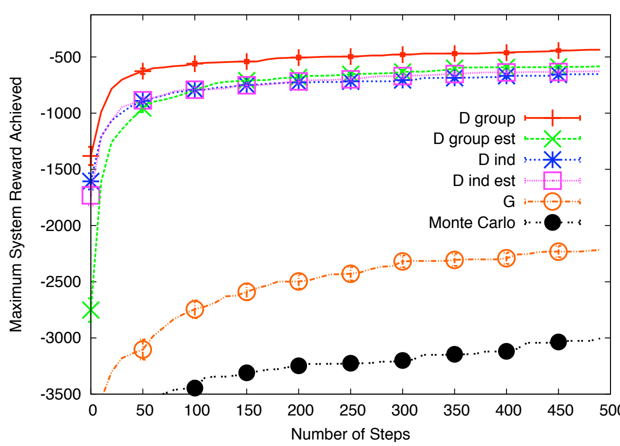

Figure 3: Performance for agents controlling ground delays on dual independent congestion problem. 300 Aircraft, 40 Agents.

The results show (Figure 3) that the different rewards' performance is qualitatively similar to the case where agents control MITs. Agents using the group difference reward perform the best, while agents using any variety of difference reward perform well. Note however, that agents using G or Monte Carlo estimation perform particularly poorly in this problem. This can be attributed to the problem being more difficult, since the action-reward mapping is more dependent on the actions of other agents. In essence, there is more "noise" in this system, and agent rewards that do not deal well with noise perform poorly.

4.3 Controlling Reroutes

In this experiment agents alleviate congestions by rerouting aircraft around congestions. Here an agent's action is to set the probability that it will reroute an aircraft that goes through it's associated fix. In this experiment agents choose between one of three probabilities: 0%, 50% and 100%. As before, the results show that using a reward that can handle the coupling is important in obtaining high performance.

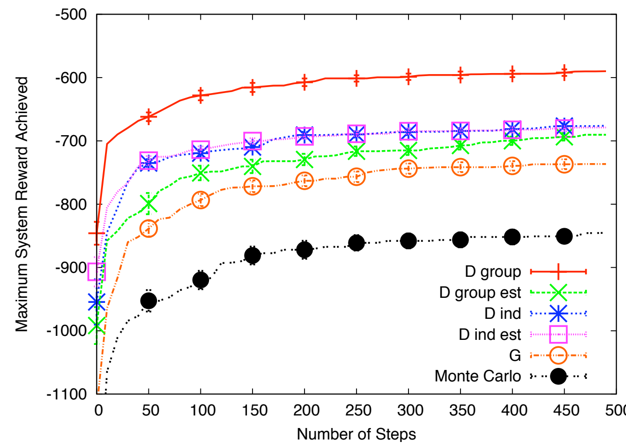

Figure 4: Performance for agents controlling rerouting on dual independent congestion problem. 300 Aircraft, 40 Agents.

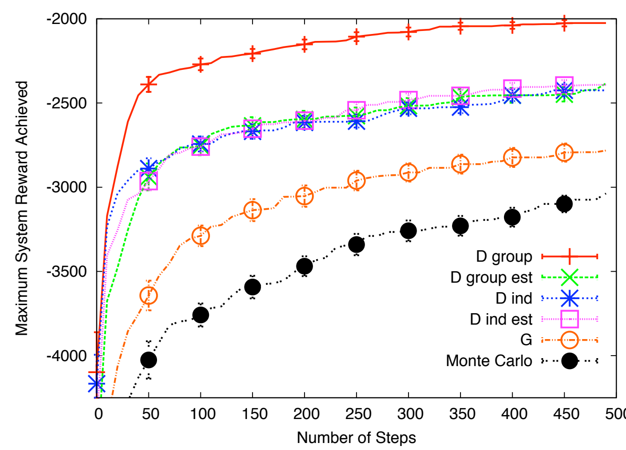

Figure 5: Performance for agents controlling reroutes on cascading congestion problem. 100 Aircraft, 40 Agents.

4.3.1 Coupled Cascading Congestions

In previous experiments the coupling between agents affects the how aircraft flow through congestion points that they are trying relieve. In this experiment (displayed in Figure 1) each aircraft goes through two congestions and is associated with two agents, each of which can reroute the aircraft around the congestion. This scenario is complicated by the fact that if too many aircraft are rerouted, they can create a new third point of congestion. This causes a pairwise coupling between agents. To isolate the effect of the coupling, we explore the case where desired capacity of a congested sector is set to zero (e.g., when background traffic has already filled sectors to capacity and the agent's job is to prevent any new traffic from entering congested areas). This situation strongly encourages each agent to reroute all aircraft around its congestion, meaning that unless good rerouting decisions are made, it is likely that the reroutes will cause a new congestion.

The results of this experiment (Figure 5) show that keeping track of coupling between agents is critical in this domain. As a consequence the agents using the group reward $D_i^{group}(x)$, perform significantly better than agents that do not account for the coupling. However note that agents using the estimate $D_i^{group-est}(x)$ do not perform better than the agents ignoring the coupling. This disparity can be attributed to the difficulty of estimating the rewards in this complex problem. In essence, because the estimates of the coupled group reward are inaccurate, the agents trade off one problem (not accounting for coupling) for another (poor estimates).

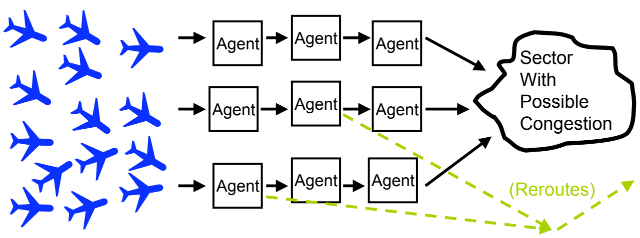

Figure 6: Aircraft going through multiple fixes.

4.3.2 Larger Coupled Groups

The results above demonstrate the need to account for the coupling among the agents. To continue this analysis we run a set of experiments where we explore how well agents perform when the size of the coupled groups changes. Due to the complexity of the large amount of couplings, these experiments are conducted on a relatively simple scenario with one congestion and one hundred aircraft. In this experiment each aircraft traveling through the point of congestion first goes through l fixes associated with l agents (Figure 6). Here the sets of l fixes are disjoint, so aircraft going through a fix in one set will never go through a fix in another set. This scenario leads to groups of coupled agents of size l.

Here agents take one of two actions: reroute no aircraft, or reroute 50% of aircraft going through their fix (chosen randomly). All agents within a group have the potential to order reroutes on the same aircraft. When one agent reroutes an aircraft, a second agent ordering the same aircraft to be rerouted will have little effect. This creates a coupling between agents in a group, since the actions of other agents affect the impact of an agent's action.

When agents are not coupled (group size 1), the performance of all the difference rewards is nearly identical, all performing better than agents directly maximizing G or agents using Monte Carlo estimation (Figure 7). This is expected because the formulations for the coupled rewards reduces to the rewards for the uncoupled cases for a group size of 1. In contrast when agents are coupled in group sizes of 4, the performance of the two rewards that take coupling into account perform considerably better (Figure 8). The issue

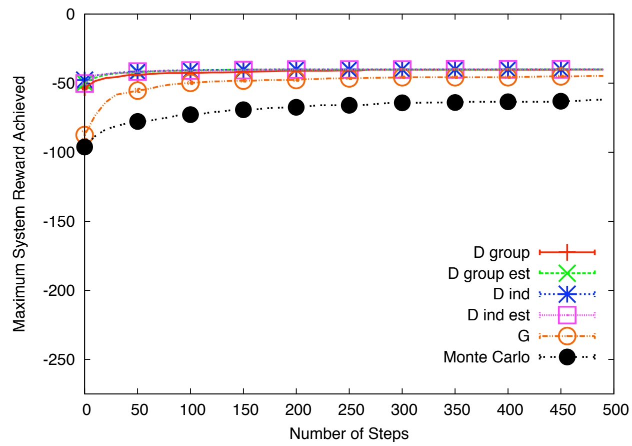

Figure 7: Performance for agents controlling reroutes for single congestion problem. 100 Aircraft, 40 Agents, Group Size 1.

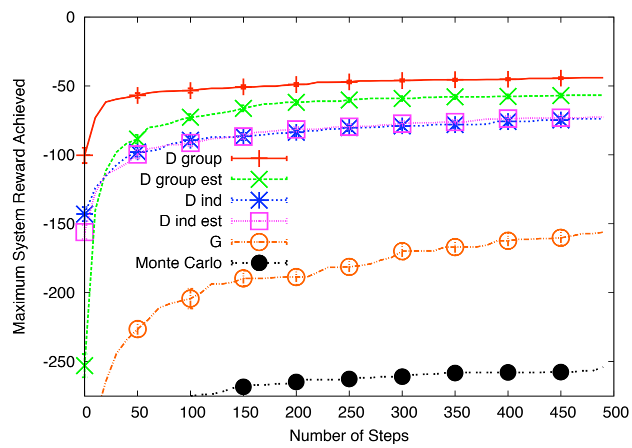

Figure 8: Performance for agents controlling reroutes for single congestion problem. 100 Aircraft, 40 Agents, Group Size 4.

here is that with a group size of 4, assuming independence is problematic. For instance take the case where an agent sees all three other agents in its group take the action of rerouting 50% of the aircraft. Assuming independence, this agent would assume that if it ordered a reroute of 50%, then it would impact 50% of the aircraft that go through its fix. However, in reality 86% (in probability) of the other aircraft have already been rerouted by the other agents so its impact would only be 7%.

Figure 9 summarizes this result. Here we show the performance of the agents with group sizes varying from 1 to 5 and scale the results by the value of achieved by Monte Carlo estimation to normalize to the relative difficulty of each problem. As expected, as the size of the coupled groups grows, the relative benefit of using rewards that account for the coupling increases.

5. DISCUSSION

In this paper we show how distributed coupled agents can efficiently use a high-fidelity air traffic flow simulator to form a powerful solution to the air traffic flow problem. In ad-

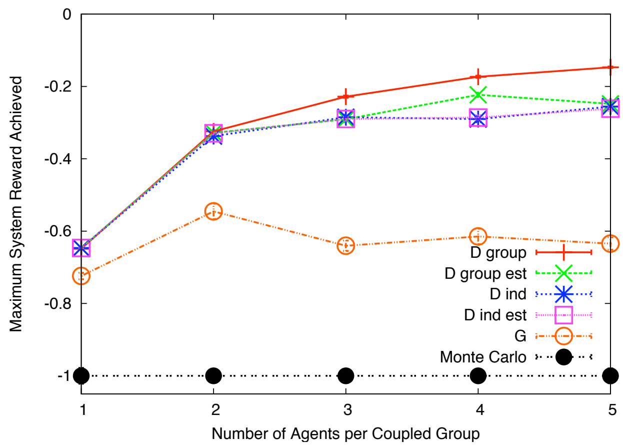

Figure 9: Relative performance for agents controlling reroutes for single congestion problem. 100 Aircraft, 40 Agents, Variable Group Size.

dition the agent solution is flexible in that agents can be turned on and off when needed, and they can be used to control air flow with a wide variety of methods. Both these capabilities are critical in the eventual use of agents in real domains where they need to interact and be trusted by human operators that can remain "in the loop."

The agent based solution presented in this paper extends existing agent-based control methods by allowing them to work in more coupled situations. While previous solutions allowed for coupling between agents to occur in the penalty function, they assumed the actions of agents affect the counts of aircraft within a sector independently. By allowing coupling between agents with respect to these counts, this paper allows multi-agent systems to be extended to a much larger variety problems, with only a small increase in complexity. Note that the coupling method in this paper is effective with only coupled groups with few agents due to the exponential nature of the encoding. However, we believe that this is applicable to many air traffic problems because typically an aircraft only goes through a few fix locations therefore only couples a few agents together. However, if large coupled groups exist, general purpose function approximation algorithms may be able to help in computing difference rewards, or in estimating aircraft counts affected by the coupled groups.

6. REFERENCES

[1] 2006 NASA Software of the Year Award. FACET: Future ATM concepts evaluation tool. Case no. ARC-14653-1, 2006.

[2] 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.

[3] A. Agogino and K. Tumer. Multi agent reward analysis for learning in noisy domains. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Utrecht, Netherlands, July 2005.

[4] K. D. Bilimoria, B. Sridhar, G. B. Chatterji, K. S. Shethand, and S. R. Grabbe. Facet: Future atm concepts evaluation tool. *Air Traffic Control Quarterly*, 9(1), 2001.

[5] 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.

[6] M. S. Eby and W. E. K. III. Free flight separation assurance using distributed algorithms. In *Proceedings of Aerospace Conference, 1999*, Aspen, CO, 1999.

[7] FAA OPSNET data Jan-Dec 2005. US Department of Transportation website. (http://www.faa.gov/data_statistics/), 2005.

[8] F. D. G. Jonker, J.-J. Ch. Meyer. Achieving cooperation among selfish agents in the air traffic management domain using signed money. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Honolulu,HI, May 2007.

[9] S. Grabbe and B. Sridhar. Central East pacific flight routing. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, 2006.

[10] J. C. Hill, F. R. Johnson, J. K. Archibald, R. L. Frost, and W. C. Stirling. A cooperative multi-agent approach to free flight. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1083–1090, New York, NY, USA, 2005. ACM Press.

[11] P. K. Menon, G. D. Sweriduk, and B. Sridhar. Optimal strategies for free flight air traffic conflict resolution. *Journal of Guidance, Control, and Dynamics*, 22(2):202–211, 1999.

[12] M. Pechoucek, D. Sislak, D. Pavlicek, and M. Uller. Autonomous agents for air-traffic deconfliction. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Hakodate, Japan, May 2006.

[13] B. Sridhar, T. Soni, K. Sheth, and G. B. Chatterji. Aggregate flow model for air-traffic management. *Journal of Guidance, Control, and Dynamics*, 29(4):992–997, 2006.

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

[15] C. Tomlin, G. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transaction on Automatic Control*, 43(4):509–521, 1998.

[16] K. Tumer and A. Agogino. Distributed agent-based air traffic flow management. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 330–337, Honolulu,HI, May 2007. **Best paper award**.

[17] K. Tumer and D. Wolpert, editors. *Collectives and the Design of Complex Systems*. Springer, New York, 2004.

[18] S. Wollkind, J. Valasek, and T. Ioerger. Conflict resolution for air traffic management using cooperative multiagent negotiation. In *AIAA Guidance, Navigation, and Control Conference*, Providence, RI, August 2004.