Learning Indirect Actions in Complex Domains: Action Suggestions for Air Traffic Control [*]

Adrian Agogino
UCSC, NASA Ames Research Center, Mailstop 269-3
Moffett Field, California 94035, USA
adrian@email.arc.nasa.gov

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

Abstract

Providing intelligent algorithms to manage the ever-increasing flow of air traffic is critical to the efficiency and economic viability of air transportation systems. Yet, current automated solutions leave existing human controllers "out of the loop" rendering the potential solutions both technically dangerous (e.g., inability to react to suddenly developing conditions) and politically charged (e.g., role of air traffic controllers in a fully automated system). Instead, this paper outlines a distributed agent based solution where agents provide suggestions to human controllers. Though conceptually pleasing, this approach introduces two critical research issues. First, the agent actions are now filtered through interactions with other agents, human controllers and the environment before leading to a system state. This indirect action-to-effect process creates a complex learning problem. Second, even in the best case, not all air traffic controllers will be willing or able to follow the agents' suggestions. This partial participation effect will require the system to be robust to the number of controllers that follow the agent suggestions. In this paper, we present an agent reward structure that allows agents to learn good actions in this indirect environment, and explore the ability of those suggestion agents to achieve good system level performance. We present a series of experiments based on real historical air traffic data combined with simulation of air traffic flow around the New York city area. Results show that the agents can improve system wide performance by up to 20% over that of human controllers alone, and that these results degrade gracefully when the number of human controllers that follow the agents' suggestions declines.

[*] Appears in *Advances in Complex Systems*, Vol 12, pp. 493-512, 2009.

1 Introduction

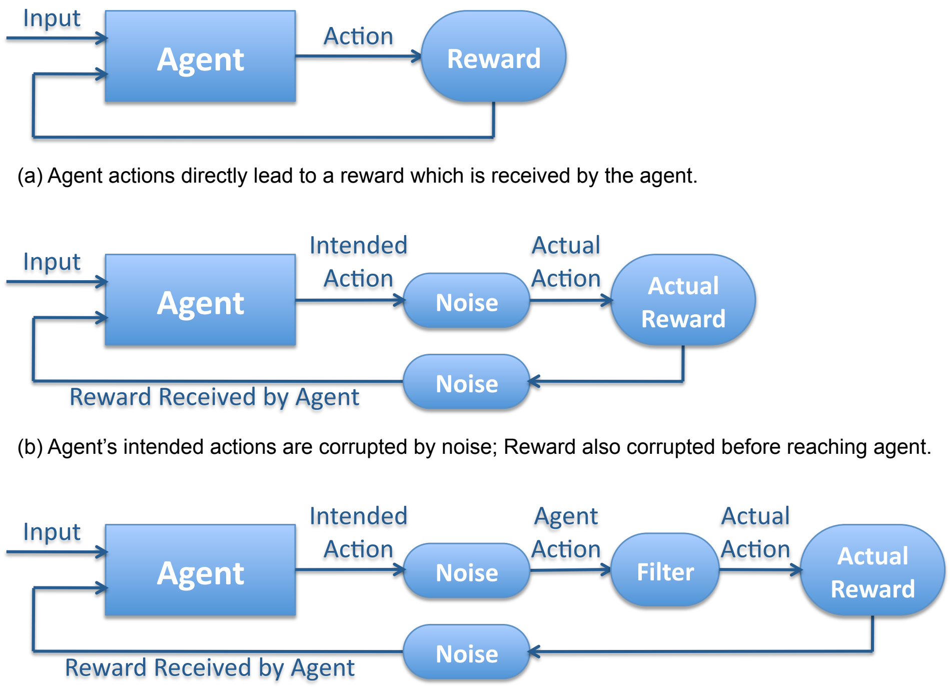
In many large complex learning problems, the actions of learning agents are distorted through interactions with other agents and the environment before resulting in a system outcome. Such systems are prevalent in many domains where the tasks are too complex to be fully automated (e.g., search and rescue where some human target selection is needed; air traffic control where flight plans are human controlled). In such domains, agents providing suggestions to another entity (e.g., human or higher level automation) leads to a new and significantly more complex learning problem: How to account for the "filtering" effect where the actions of an agent may or may not be followed, leading to the credit assignment problem of how to determine whether the agent actions were beneficial to the system.

Traffic flow problems in general, and air traffic flow management provides a perfect example of such a domain [3, 5, 7, 8, 20, 26]. On a typical day, more than 40,000 commercial flights operate within the US airspace [23]. In order to efficiently and safely route this air traffic, current traffic flow control relies on a centralized, hierarchical routing strategy that performs flow projections ranging from one to six hours. As a consequence, the system is slow to respond to developing weather or airport conditions leading potentially minor local delays to cascade into large regional congestions. Federal Aviation Administration (FAA) statistics estimate that weather, routing decisions and airport conditions caused 1,682,700 hours of delays in 2007 [15]. The total cost of these delays was put to over $41 Billion by a Joint Economic Committee study [1]. Unlike many other flow problems where the increasing traffic is to some extent absorbed by improved hardware (e.g., more servers with larger memories and faster CPUs for internet routing) the air traffic domain needs to find mainly algorithmic solutions, as the infrastructure (e.g., number of the airports) will not change significantly to impact the flow problem.

In addition, the FAA estimates that in the next few decades, air traffic will increase threefold with little to no capacity improvements coming from infrastructure upgrades [14]. As a consequence, the only way in which this increased demand can be met is by better utilizing the air space and current air traffic system. Algorithms that will effectively automate air traffic flow will play a critical role in this endeavor. However, before a fully automated system that can control the airspace efficiently and safely without human intervention can even be contemplated, they need to overcome both technical and political obstacles: First, from a technical standpoint, the learning agent-based system needs to successfully handle unusual events that were not encountered during training. Indeed, air transportation has a remarkable safety record, and any agent-based air traffic control system has a very high threshold to reach before being cleared to be deployed. Second, from a political/adoption standpoint, the move towards a fully automated system need to garner the trust of the users (air traffic controllers, pilots, and ultimately passengers). Indeed, any approach that aimed to remove air traffic controllers from the system would be impossible to test or implement, and would not be likely to receive the required certifications from regulators.

As a consequence, there is a strong incentive to design a system that blends learning agent behavior with the experience of air traffic controllers. In this paper, we provide a "suggestion-based" system that benefits from the learning aspects of a multiagent system, but also keeps the human controller "in-the-loop."

But this approach creates the new and complex learning problem. The basic information flow of a learning system is Shown in Figure 1(a). The agent receives input from the environment and takes an action. That actions leads to a reward based on its impact on the environment and that reward is propagated back to the agent. This basic relationship between actions and reward is the foundation of reinforcement learning [24, 28], and is conceptually similar to evolutionary approaches where the system learns through improving its fitness in response to actions [4]. In many real world systems, the agent actions are corrupted by noise (sensor noise, actuator noise) as is the reward (external factors). Figure 1(b) shows the information flow for such systems. Figure 1(c) shows an altogether more complex situation where in addition to noise, the agent actions can be modified or ignored. In such cases, the way in which the agent rewards are designed is critical in the agent succeeding in extracting useful information from its interaction with the environment. It is this scenario that we investigate in this



(a) Agent actions directly lead to a reward which is received by the agent.

(b) Agent's intended actions are corrupted by noise; Reward also corrupted before reaching agent.

(c) Agent actions are filtered through an unknown process before leading to a reward.

Figure 1: The traditional agent acting in a simple environment is shown in (a). The agent takes an action, which leads to a reward, which is then used by the agent to update its internal parameters. In moderately complex environments, the actual agent actions can differ from intended actions (e.g., noisy actuators). Similarly, the reward may contain some noise (e.g, stochastic reward). In some environments, the agent actions may be further "filtered" through a process before resulting in a reward (e.g., an operator over-ruling a remote drone, an air traffic controller ignoring an agent's recommendation). In that case the agent has a complex learning problem where the action to reward mapping is filtered by an unknown process.

paper though the air traffic flow management problem.

1.1 Contributions of this Work

The contributions of this work is to present a learning method where instead of taking actions to control the air traffic, the agents make suggestions that may or may not be followed by human controllers. This approach is based on a natural extension of a cooperative agent-based control algorithm, where a suggestion agent is assigned to a single human controller, or a small group of controllers working in the same area. Through a carefully crafted reward structure, each agent then uses reinforcement learning to make suggestions that lead to high values of a system-wide reward. Such reinforcement learning agents have to overcome two challenges: 1) What actions should be taken by the human controller to maximize the system reward; and, 2) What suggestions can be made that will not be ignored by the human controller. In this formulation, the agent actions go through a "filter" (the air traffic controllers) before resulting in a particular reward. Note, the purpose of this paper is not to model human controllers, but to determine whether effective learning can be achieved when the agent actions are filtered before leading to a system reward.

Addressing this problem successfully will provide solutions with three distinct benefits to the domain of air traffic control and/or to the field of learning multiagent systems:

1. **Locality**: With a distributed agent approach, each agent will specialize in a local space. The human controller will interact with the same agent and will know in what situations it is best to follow the suggestions of the agent, significantly reducing the controllers cognitive workload.

2. **Collaboration**: The multiagent approach though will allow the agents to look at the "big picture" whereas the human controllers are generally concerned with local issues. This interaction between local and global objectives will allow the agents to make suggestions that induce the human controller to take actions that are beneficial to system-wide goals.

3. **Human-in-the-Loop**: The human controller can ignore the suggestion of the automation agent if the suggestion are considered dangerous. This safety feature is not only critical to the adoption of multiagent systems in complex domains such as air traffic control, but will also lead to new learning algorithms better suited to learning when actions can be modified.

In this paper we show how a multiagent reinforcement learning system can overcome the technical challenges and be effective in making suggestions that lead to achieving better system-wide objectives.Below, we provide a brief discussion of recent work on air traffic control. Section 2 describes the air traffic management problem, the system evaluation function and the FACET simulator. Section 3 presents the control structure for air traffic for both humans in the loop and purely automated approaches. Section 4 describes the selection of the agents, their actions and their reward functions for the problem of controlling air traffic flow. Section 5 first shows how the agent-based solution performs in the air traffic flow problem in the simpler situation where human controllers are not part of the system, and shows results for human-in-the-loop control, where agents are giving suggestions to human controllers. Finally, Section 6 provides a discussion on the presented results, the key issues in air traffic control automation and future research directions.

1.2 Related Work

This paper presents a particular agent based solution where a set of distributed agents make suggestions to air traffic controllers. As agents methods are a natural tool to help automate existing air traffic systems, there has been significant research into other agent solutions as well [16, 22, 25]. These solutions typically involve a set of intelligent agents that try to optimize some overall goal either through learning or through negotiation (note these are not closely related to "agents" in other fields that attempt to model or augment human behavior) [11, 12, 17, 18]. For learning agents, one key problem that needs to be addressed is how an objective function can be made so that agents do not learn to hinder each other. In other contexts this has been addressed through a "satisficing" reward that specifically encourages cooperation and penalizes anti-cooperative behavior [17], and difference rewards where the actions of the agents aim to improve the system wide performance criteria [26, 3].

In addition to agent solutions, first principles-based modeling approaches used by domain experts can offer good performance. These methods tend to be divided into Lagrangian models where the dynamics of individual aircraft are taken into account and Eulerian (along with Aggregate) models where only the aggregate flow patterns are modeled. In both cases, creating optimization from the resulting model is a complex process and numerous complexity reduction steps need to be taken. Lagrangian models for air traffic flow management involve computing the trajectories of individual aircraft [6, 19]. These models are very flexible in that they can be used for a range of problems, from collision avoidance to congestion reduction. Instead of predicting the path of individual aircraft, Eulerian and aggregate flow models predict flow patterns in the airspace [9, 13, 21]. While these models lose some of the fine granularity of the Lagrangian approach, they lead to a simplification of the airspace allowing for more optimization techniques to be applied. In general both Lagrangian and Eulerian methods can be effective, especially when a top-down, fully automated system is desired.

2 Air Traffic Flow Management

The management of traffic flow is a complex and demanding problem. Critical issues include efficiency (e.g., reduce delays), fairness (e.g., deal with different airlines), adaptability (e.g., respond to developing weather patterns), reliability and safety (e.g., manage airports). In order to address such issues, the management of this traffic flow occurs over four hierarchical levels. These levels range from operations involving critical separation decisions, to the politics of long term airspace configuration. In between these extremes, the multiagent work presented in this paper focuses on the "regional" and "national flow" where agents look at time horizons between twenty minutes and eight hours. Our domain is amenable to learning systems, fitting between the long term planning by the FAA and the very short term decisions by air traffic controllers.

By focusing on the air traffic flow across multiple regions, we investigate how to integrate agent and air traffic decision processes and how to use agents to provide air traffic controllers with suggestions that do not contradict their intuition. In this section we provide a system evaluation function and a simulator that will allow us to both compute the system performance and simulate the dynamics of the complex air traffic transportation domain.

2.1 System Evaluation

The overall goal of our multiagent system is to optimize a system-wide evaluation criterion. Depending on the organizations involved there are many ways to rate the performance of various parts of the air traffic system, ranging from fairness between the airlines to maximizing profits. In this paper we focus on a system performance evaluation function concerned with the overall performance of air traffic flow. This evaluation focuses on 1) the amount of measured air traffic delay and on 2) the amount of congestion in a particular region of airspace. The linear combination of these two terms gives the full system evaluation function, $G(z)$ as a function of the full system state z [26] (for example, z captures all information needed to compute the reward, including the location of all aircraft at a given time). More precisely, we have:

$$G(z) = -((1-\alpha)B(z) + \alpha C(z)) \ , \tag{1}$$

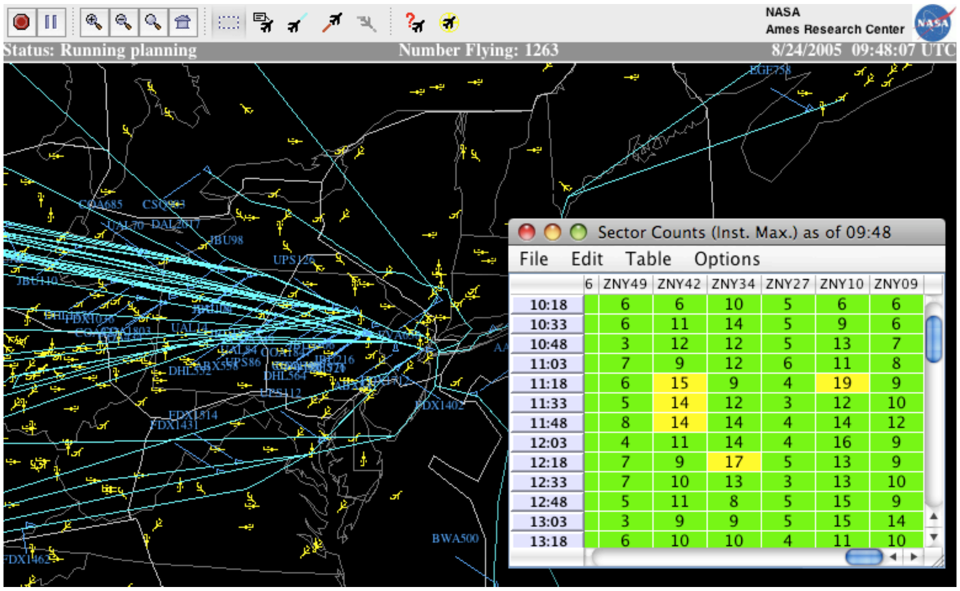
where $B(z)$ is the total delay penalty for all aircraft in the system, and $C(z)$ is the total congestion penalty. The relative importance of these two penalties is determined by the value of α.

Both $B(z)$, and $C(z)$ are measured by looking at the number of aircraft that are present throughout the day in regions of airspace known as "sectors." In the United States, the airspace is broken up into approximately 800 sectors. Each sector has a maximum capacity (usually around 15 aircraft), and the number of aircraft in a sector at any given time should be below the sector capacity. One of the primary goals in air-traffic flow management is to reduce congestions so that these sector capacity constraints are met.

Intuitively, $C(z)$ penalizes a system state where the number of aircraft in a sector exceeds the FAAs official sector capacity. Similarly, $B(z)$ is based on the number of aircraft that remain in a sector past a predetermined time, and scales their contribution by the amount by which they are late. In this manner $B(z)$ provides a delay factor that not only accounts for all aircraft that are late, but also provides a scale to measure their "lateness".

2.2 Simulation: FACET

To provide the values for both $B(z)$ and $C(z)$, we use FACET (Future ATM Concepts Evaluation Tool, where ATM stands for Air Traffic Management), a physics based model of the US airspace that was



Status: Running planning Number Flying: 1263 8/24/2005 09:48:07 UTC

Sector Counts (Inst. Max.) as of 09:48

File Edit Table Options

	6	ZNY49	ZNY42	ZNY34	ZNY27	ZNY10	ZNY09
10:18		6	6	10	5	6	6
10:33		6	11	14	5	9	6
10:48		3	12	12	5	13	7
11:03		7	9	12	6	11	8
11:18		6	15	9	4	19	9
11:33		5	14	12	3	12	10
11:48		8	14	14	4	14	12
12:03		4	11	14	4	16	9
12:18		7	9	17	5	13	9
12:33		7	10	13	3	13	10
12:48		5	11	8	5	15	9
13:03		3	9	9	5	15	14
13:18		6	10	10	4	11	10

Figure 2: FACET screenshot displaying traffic routes and air flow statistics.

developed to accurately model the complex air traffic flow problem [10]. It is based on propagating the trajectories of proposed flights forward in time (Figure 2). FACET is extensively used by the FAA, NASA and industry (over 40 organizations and 5000 users) [10]. In this paper, agents have FACET simulate air traffic based on their control actions. The agents then produce their rewards based on received feedback from FACET about the impact of these actions.

FACET simulates air traffic and using a graphical user interface, allows the user to analyze congestion patterns of different sectors and centers (Figure 2). FACET also allows the user to change the flight paths of the aircraft, leading to changes in delays and congestion. In this paper, agents use FACET directly through "batch mode", where agents send scripts to FACET requesting air traffic statistics based on agent actions, and use those statistics to compute their reward functions.

3 Flow Control Architectures

While the main goal of this paper is to discuss a model where agents give suggestions to a human-in-the-loop system, it is instructive to first describe the simpler system where agents have full autonomy. In this section we first describe such a system, and then insert the human controllers and modify the control loop to account for the actions of the human controllers.

3.1 Human out of the Loop Control

In the purely automated model, the agents directly affect air traffic flow. Figure 3 shows the information and action flow diagram. In this model, each agent first chooses an action. The results of all the agents' actions are then simulated in FACET. From the simulation all congestion and lateness issues are observed, a system reward is computed and agents compute their appropriate rewards. While agents may use the system evaluation as their reward there are other possibilities too, discussed in Section 4. These rewards are then used to modify the agents' control policies, which are then used to choose the next action. This approach was explored first and yielded promising results [26], though the role of air traffic controllers was not addressed.

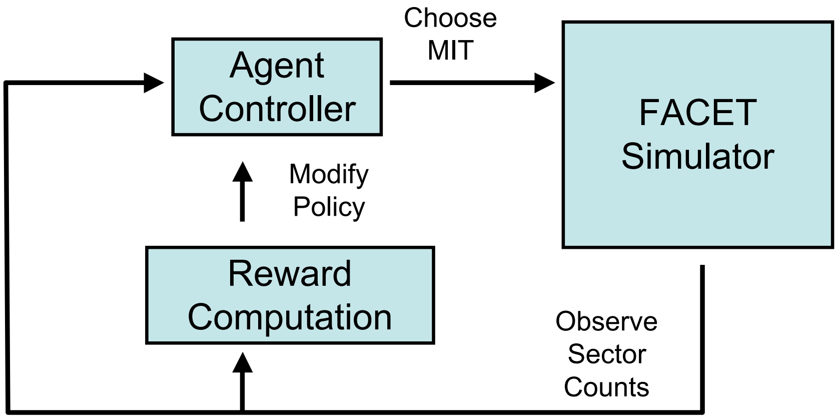


Figure 3: Control structure with agents taking actions to directly maximize system reward. Agents select a "Miles in Trail" (MIT) value that impacts traffic flow (MIT values are described in detail in Section 4.2).

3.2 Human in the Loop Control

While giving agents complete control of a system greatly simplifies the control problem and can lead to high performance, it is unrealistic in an air traffic control domain. Air traffic control is human intensive, and aiming to have air traffic controllers accept being helped by intelligent agents is a critical component in improving the air traffic control problem. In addition even if one could overcome the political hurdle, replacing human controllers would result in a potentially dangerous system whose performance may degrade or fail during unpredictable events.

This section describes a system where the agent only makes a suggestion while another entity takes the actual action. In this work, we assume that human controllers take actions, while receiving suggestions from the agents. As before the agents are reinforcement learning algorithms trying to maximize a system wide reward. Here we will also model the human controller as reward maximizing entities. While the human controller's reward is usually related to the system reward, it is different in that the human controller's focus tends to be comprised of more local and immediate concerns, rather than system wide concerns (see below for a full description of the rewards).

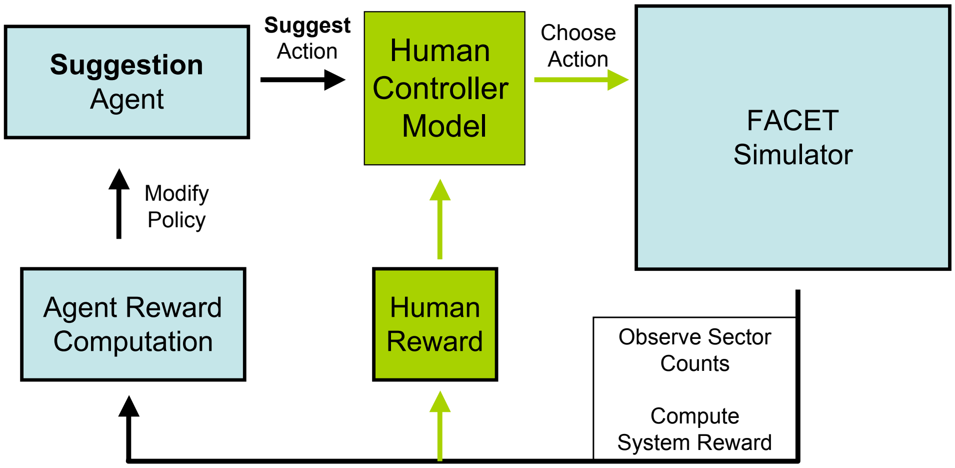


Figure 4: Control structure with reward maximizing agents interacting with reward maximizing humans.

Figure 4 shows the learning process when agents are working in a system where humans are in-the-loop. In this process a suggestion agent first offers a suggested action. A human controller takes that suggestion and issues the action. The effects of the actions from all the human controllers are then simulated in FACET and the resulting congestions and delays are observed. These values are then used to by the suggestion agents and the human controllers to create their respective reward values. At the end of the cycle both the suggestion agents and the human controllers update their control policies and start a new cycle. There are two key issues to note in this model: First, for the suggestion agents to be relevant, the human controller needs some incentive to follow their suggestion. Otherwise the model simply reduces to the right hand cycle. Second, there are two learning entities (controllers and suggestions agents) operating in the same system. The learning cycles need to be clearly delineated to ensure the assumptions of the learning algorithms are met [24].

3.2.1 Human Controller and Agent Models

In our system model, the “human controller” is an entity maximizing a reward and has some incentive to follow suggestions through the shape of that reward. In fact, the “human controller” need not be human at all. It could simply be another reward maximizing algorithm that is responsible for taking actions that affect air traffic flow. While agent modeling of humans is a large field and detailed models of air traffic controllers exist [29], in this paper we use a simple model based on reward maximization. Indeed, the salient properties we wish our models to have is: 1) The human model is trying to adaptively maximize a reward; 2) this reward may not be aligned with the system reward. To achieve this we simply model the human controller as a reward maximizer (e.g., reinforcement learner), using the same basic learning as the suggestion agents. However, the reward maximizers modeling the air traffic controllers differ significantly from the suggestion agents in their inputs, actions and rewards.

The learning cycle for the agents and human controller models happens in three stages:

1. agents make suggestions,
2. human controllers take actions,
3. rewards for both agents and human controllers are computed based on actions.

In stage 1, both the agent and the human controllers are in a start state, and the agent makes a suggestion, while the human controller is idle. In stage 2, the agent’s suggestion becomes the human controller’s input, and the controller takes an action based on the suggestion, while the agent remains idle. In stage 3 both the agent and human controllers update their policies based on the different rewards they have received.

The suggestion agents are trying to maximize the same system reward as the fully automated agents discussed above (given in Section 2.1), while each human controller is trying to maximize his/her own reward. Note that in this model the human controller has an immediate reward based on his/her action, much like the agents had in the human-out-of-the-loop model. In contrast the agent’s reward is now indirectly based on its action: the reward is received two time steps after its action, and the effects of its action are filtered through the human controller. This filtering process leads to a significantly more difficult learning problem for the suggestion agent.

3.2.2 Rewards for the Air Traffic Controllers

In the human-in-the-loop scenario, the agents keep the same system rewards as before, where their goal is to minimize both congestion and lateness. However, unlike before, the agents do not directly select actions. Instead a human controller (modeled as a reward maximizer) selects actions after receiving a suggestion from his/her agent. One important aspect of this domain is that while each agent is trying to minimize both congestion and lateness, we assume that the human controller is only concerned with minimizing congestion. This is a reasonable assumption, since congestion is a primary concern as it affects safety as well as the controllers workload. While the controller's goal is to minimize congestion, we also assume that they have some incentive to follow the suggestion from the agent. This incentive could be induced in a number of ways, but for now we simply assume it is imposed by management. We therefore model a controller's reward as a linear combination of controller's intrinsic goal of wanting to reduce congestion, and the controller's imposed incentive to follow the agent's suggestion:

$$H_i(y) = -(1-w)C(y) + wK_i(y_i, x_i) \ , \tag{2}$$

where y is the control of the human controller, x is the suggestion by the agent, w is a weight and $K(y,x)$ is the incentive for the human to follow the suggestion. In this paper the incentive will simply be the numerical difference between the agent's suggestion and the human's suggestion: $K_i(y_i, x_i) = |y_i - x_i|$. This incentive intuitively reinforces the notion that controllers are more likely to follow suggestions with which they agree. Note that when $w = 0.0$ the agents' suggestions are completely ignored and when $w = 1.0$ the human controller's reward is solely based on complying with the agents' suggestions.

4 Integrating Agents into Traffic Flow Management

So far we have discussed in general terms how agents interact with the FACET simulator and human controllers, but we have not defined how agents can best be used in an air traffic flow management problem. This problem relies on adaptive agents either taking independent actions or making suggestions to air traffic controllers to maximize the system evaluation function (depending on whether humans are in the loop). The selection of the agents, their actions and their reward structure is therefore critical to the success of this approach. These selections are important for a multiagent system to be effective in the air traffic domain, whether the agents are taking actions that directly control air traffic or if they are making suggestion to human controllers that will ultimately influence the air traffic flow.

4.1 Agent Selection

Selecting the aircraft as agents is perhaps the most obvious choice for defining an agent. That selection has the advantage that agent actions can be intuitive (e.g., change of flight plan, increase or decrease speed and altitude) and offer a high level of granularity, in that each agent can have its own policy. However, there are several problems with that approach. First, there are in excess of 40,000 aircraft in a given day, leading to a massively large multiagent system. Second, as the agents would not be able to sample their state space sufficiently, learning would be prohibitively slow.

As an alternative, we assign agents to individual ground locations throughout the airspace called "fixes." Each agent is then responsible for any aircraft going through its fix location. Pairing agents with fixes works well because a single agent can control many aircraft, yet there are enough fixes to allow for a reasonably sized multiagent system with a high degree of flexibility in controlling flow patterns. Furthermore, such fix agents are ideally suited to provide suggestions to air traffic controllers (see below for agent actions).

4.2 Agent Action Sets

Based on this definition of an agent, there are several different ways in which an agent can control aircraft, including ordering reroutes, ground delays and aircraft metering. In this paper we choose to have agents control metering. In this setup the agents set values of "miles in trail" (MIT) which is the distance aircraft have to keep from each other while 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. Using MIT has proven to be an effective method of controlling the flow of aircraft [26, 3]. Figure 5 shows a diagram of how these agents could function. Here a set of

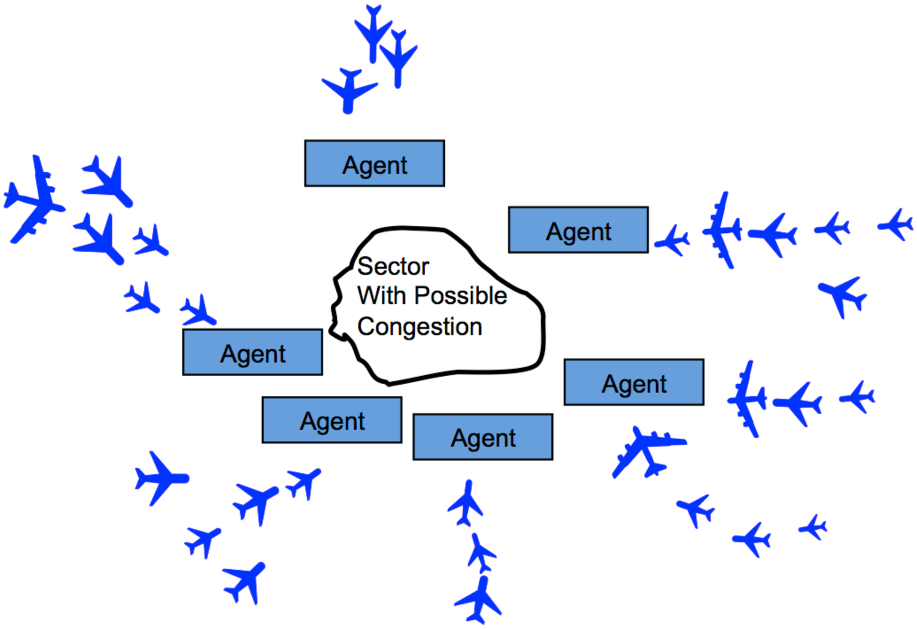


Figure 5: Schematic of agent architecture. The agents corresponding to fixes surrounding a possible congestion set separation distances.

aircraft are heading towards an area with possible heavy congestion. A set of agents surrounding the congestion could then perform metering operations that reduce the congestion. Note that in a system without human controllers, the agents are directly choosing the metering values. When agents are used instead, with a human in the loop, they are simply suggesting MIT values to the human controller. In this work, we only consider agents selecting MIT values (miles in trail) though other actions such as ground holds and re-routes have also been explored [3].

4.3 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 [24] (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 learner is used. Our reinforcement learner is equivalent to an ϵ-greedy action-value learner [24]. 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 value for action a, $V(a)$ (which is its estimate of the value for taking that action [24]) as follows:

$$V(a) \leftarrow (1-\lambda)V(a) + (\lambda)R, \tag{3}$$

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.

4.4 Agent Reward Structure

While the overall goal of the system is to maximize the system reward function, an agent does not need to maximize this reward directly. While maximizing the system reward directly has been successfully used in small, multiagent reinforcement learning problems, it does not scale well since the impact of a single agent's actions of the system reward is relatively small [27]. To alleviate this problem, we also explore using a reward that is more agent-specific. To that end, 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 [27, 30], given by:

$$D_i \equiv G(z) - G(z - z_i + c_i) \ , \tag{4}$$

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 (for example a fixed miles in trail value). In this context the vector $z - z_i$ means the state where components of agent i have been removed.

In many situations, there is 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 causes D to evaluate the agent's contribution to the system performance. The two main advantages of the D reward is that it is aligned with G, and because the second term removes a significant portion of the impact of other agents in the system, it provides agent a "cleaner" signal than G.

5 Experimental Results for New York Area

To show the performance of the agent controllers we perform a set of experiments using air traffic data from the New York City area. In each experiment there are 10 agents, with each agent being assigned a fix located on one of the inbound flows into New York. The fixes chosen are the ones where metering has the most significant influence on congestion. Altogether there are approximately 681 aircraft going through the 10 fixes over a 14 hour time window. Figure 6 shows the FACET shot of the New York airspace with the fix agents superimposed on it.

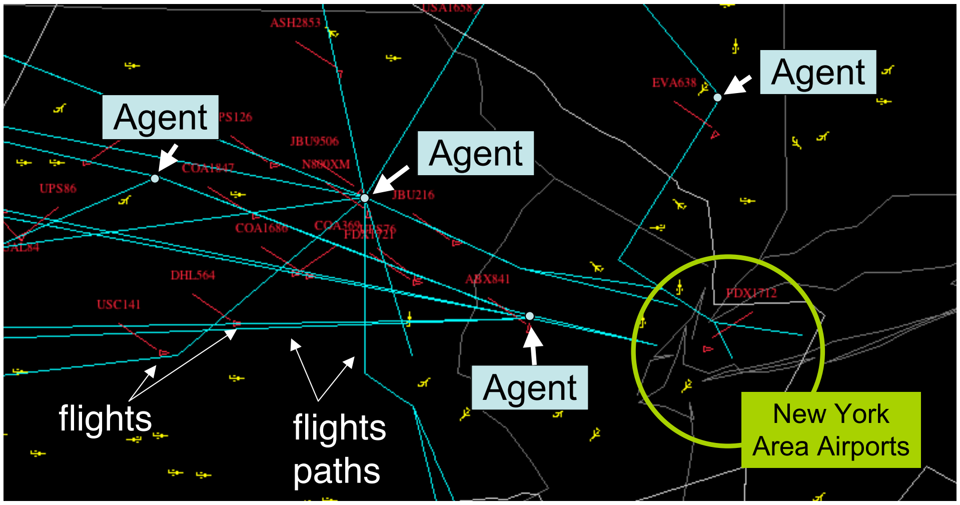


Figure 6: A screen shot of the New York airspace with four of the fix agents superimposed. The objective of each agent is to select the miles in trail that will provide the desired system behavior (e.g., minimize delay and avoid congestion).

5.1 Human out of the Loop Results

Figure 7 shows the results of agents aiming to improve system performance, without having humans in the loop. Here, we compare results of agents that try to directly maximize the system reward (G) and agents indirectly maximizing the system reward by directly maximizing the difference reward (D). The results show that through the learning process, the agents are able to significantly improve from their initially poor policy and converge to a better policy. Note that agents maximizing G learn more slowly than agents maximizing D. This is not surprising since each agent using G has to discern the effect of its action from the effect of the actions of the 9 other agents. However, in this case the difference between agents using these two different rewards is not very large, as 10 agents is not a large system. Also note that agents using both rewards perform significantly better than the random initial solution. They also perform significantly better than best fixed solution of MIT=4, which corresponds to a reward of -6979 on this scale.

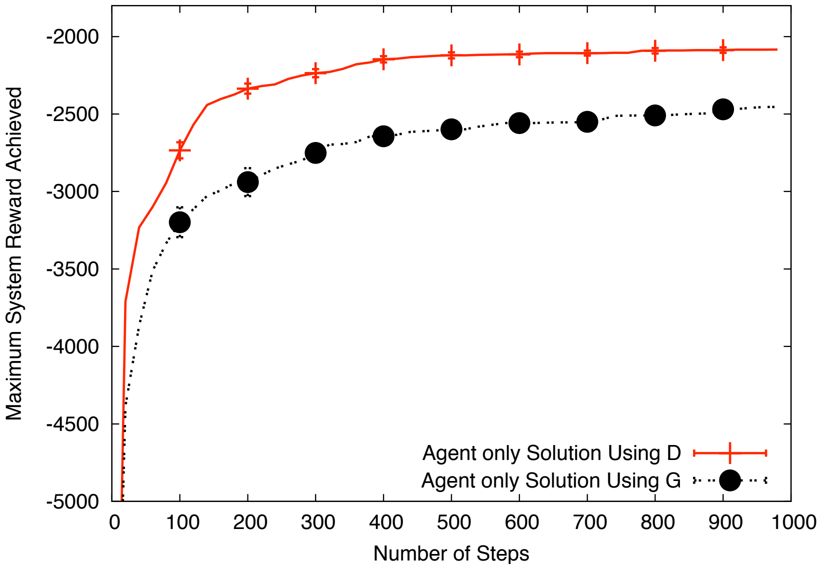


Figure 7: Performance of agent-only system. Agents improve significantly from initial random solution. Note, also agents are significantly better than the best fixed solution of MIT=4, which in this system corresponds to a reward of -6979, not even on the chart at this scale.

Figure 8 explores how the performance of the agents is impacted when their actions are replaced with a random action. As expected the results show that in such a situation system performance goes down. Interestingly when only a small number of agents are ignored the performance only goes down slightly. This can be attributed to the agents that are not being ignored learning to overcome the situations ("picking up the slack" of the ignored agents). However, this becomes increasingly problematic as more agents' actions are replaced with random action and performance inevitably declines.

Note that we do not specify why these agents are ignored. It could be due to hardware failure, or some agents simply being turned off. Another possibility is that the agents are only giving suggestions to a human controller, but are occasionally ignored. This possibility leads to the next section where we discuss agents working in a human-in-the-loop environment. This environment is more complex than the one in this section, as humans will be modeled as reward maximizing entities, not simply random controllers.

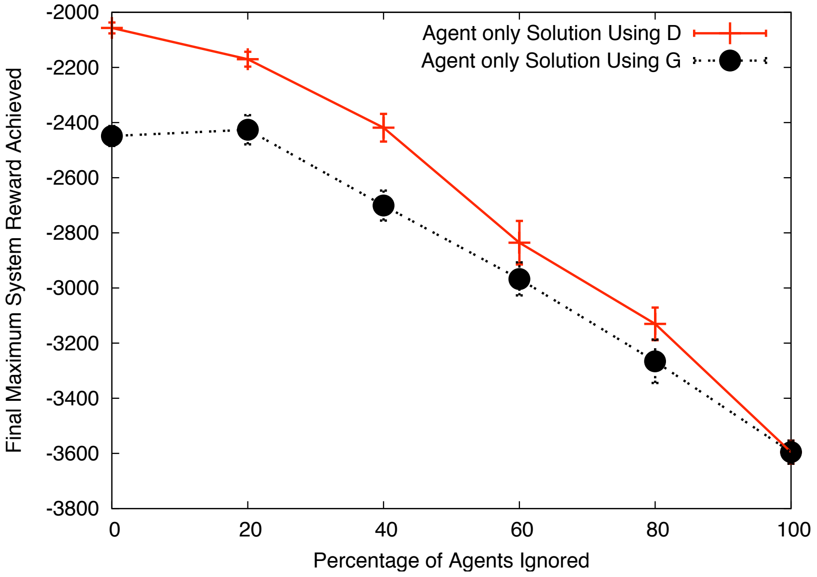


Figure 8: Performance degradation of agent-only system when agents are ignored (e.g., fail or are turned off).

5.2 Human in the Loop Results

We now introduce the human controllers into the system. In this approach, the agents provide suggestions to human controllers rather than directly assign MIT values. The system state and performance is computed based on the human controllers actions, yet the agents pair the reward they receive with their own suggested action (which may or may not have been followed).

Figure 9 shows the performance of the suggestion agent system when $w = 0.6$. It is clear that when human controllers are only minimizing congestion, they do not perform well at maximizing the system reward that includes both congestion and lateness. (This is indeed precisely the situation right now, where air traffic controllers focus on local and congestion based performance criteria, whereas the full system performance has many components including lateness and congestion.) The agent-only solution where the agents directly choose actions to minimize both congestion and lateness performs best, as can be expected. However, in between the extremes the suggestion agents are able to significantly influence the human controllers to take actions that improve the system reward. This shows that suggestion agents can make suggestions that will be followed, even though the suggestions might not be completely aligned with what the human controller assumed would be their best actions.

Figure 10 shows the importance of the incentives for the human controllers to follow the advice of the suggestion agent. This is done by moving the weight w from 0.0 to 1.0. For agents using the difference reward, as expected the more incentive the human controllers have to follow the suggestion agents, the better the system performs. Note however that even at $w = 1$ when the only goal of human controllers is to follow their suggestion agents, the system still does not perform as well as in an agent-only system. This lower performance can be explained by this system having two coupled learning cycles: 1) The suggestion agent has to learn how to give good suggestions, and 2) The human controller has to learn to follow these suggestions. Since the feedback to the agent is more indirect, learning is slower.

The learning interaction between human controllers and agents directly maximizing the system reward, G, is even more interesting. Paradoxically, beyond a certain point increasing the weight of the agents' suggestions reduces performance. This can be explained by an instability when both agents and controllers are learning in the same system. When agents are slow to learn the impact of their

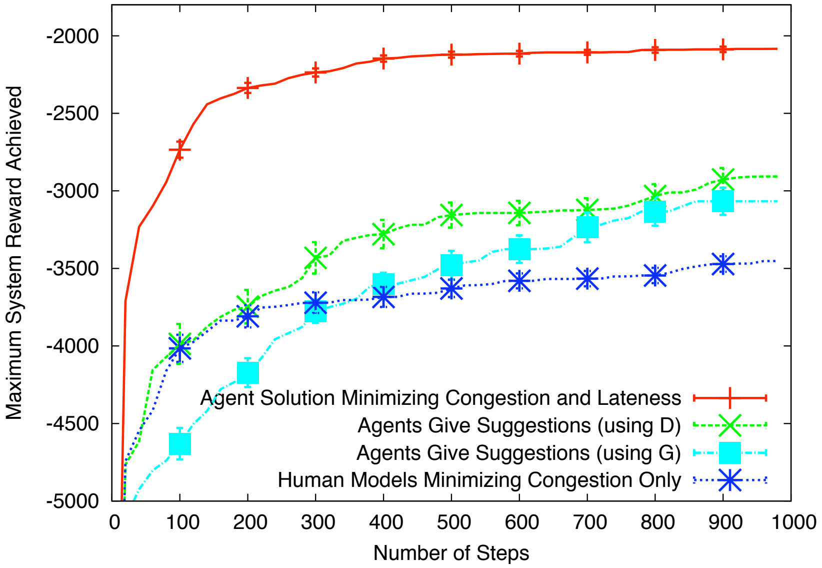


Figure 9: Comparison of agent impact of agent-human systems. Agent suggestions do improve human performance.

suggestions, the human controller may just perceive those suggestions as noise and not be able to learn to follow them. This leads to a situation where no effective signal is being sent back to the agent. This situation is somewhat mitigated when $w < 1$ as the human controllers are also striving to maximize their own rewards and can simply treat the agents suggestions as noise until the system reaches stability. Because of the impact of D in providing a more direct feedback to a particular agent's impact on the system, suggestion agents using D avoid this problem.

6 Discussion

The efficient, safe and reliable management of air traffic flow is a complex problem, requiring solutions that integrate the actions of automated agents and human air traffic controllers. Indeed, regardless of how promising a fully automated solution may be to the air traffic control problem, it is unlikely to be implemented in the real world without some oversight by human operators. There are many reasons for this, ranging from the difficulty in providing performance guarantees for unforeseen circumstances in the complex environment in which such a system is expected to operate, to the politically charged issue of replacing air traffic controllers.

To address such issues, and provide a realistic scenario, in this work we focus on an approach where agents make suggestions that may or may not be followed by the entity making the final decisions (e.g., air traffic controllers). This approach both offers benefits and presents new challenges. By keeping a human-in-the-loop, it provides many safeguards and allows the use of local information by air traffic controllers to be tempered by the "big picture" view an automated system can provide. However, it also presents a significantly more complex learning problem than a fully automated system, as agents not only need to learn good actions, but actions that are likely to be followed by the air traffic controllers. Indeed, from a purely mathematical perspective, this process is akin to having the agent's actions be filtered before resulting a reward, creating an extra layer of uncertainty for the agents.

In this paper, we test our approach on real world data obtained from the New York airspace. We show that agents providing suggestions can improve the system performance by up to 20% over a system

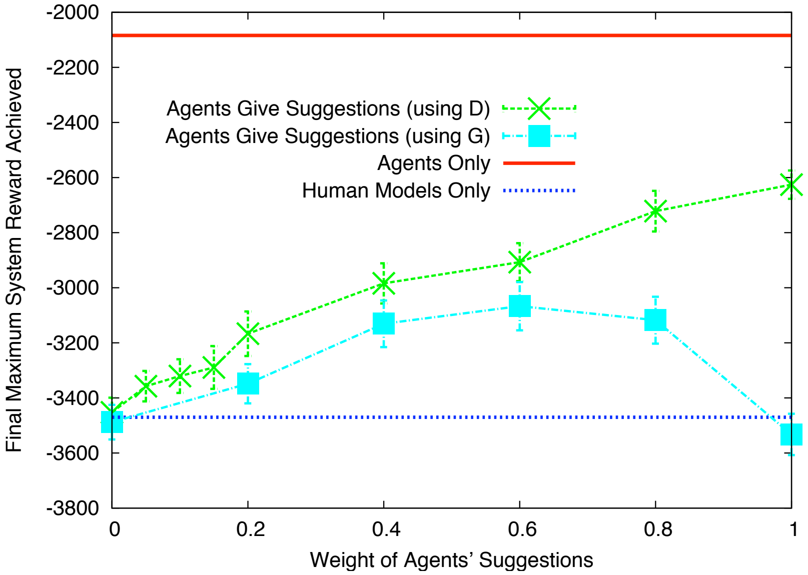


Figure 10: Effects on varying the incentive of a human controller to listen to agent suggestions (0 means agents are ignored and 1 means agent suggestions is the target). Results shown for end of runs (1000 steps).

without suggestion agents in simulations with real world data. Finally, we show that the performance of the system degrades gracefully when the air traffic controllers start to ignore the suggestion agents. Though promising, the results in this paper can be extended in multiple directions. First, the models of the human air traffic controllers can be expanded to account for a larger set of criteria/motivation. Second, the system evaluation function can become more realistic by having time-dependent coefficients that rate lateness and congestion differently based on need. Third, the real world data can cover a larger region than the New York area, incorporating multiple hubs. We are currently investigating such extensions with the ultimate goal of providing a safe, reliable and implementable air traffic control algorithm that while keeping humans in the loop will both improve the efficiency of the humans and allow the capacity of the airspace to increase to accommodate the expected rise in air traffic.

Acknowledgments: The authors thank Banavar Sridhar for his invaluable help in describing both current air traffic flow management and NGATS, and Shon Grabbe for his detailed tutorials on FACET. This work was partially supported by NSF grants 0931591.

References

[1] Your flight has been delayed again, *Joint Economic Committee* (2008) 1–12.

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

[3] Agogino, A. and Tumer, K., Regulating air traffic flow with coupled agents, in *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems* (Estoril,Portugal, 2008).

[4] Back, T., Fogel, D. B., and Michalewicz, Z. (eds.), *Handbook of Evolutionary Computation* (Oxford University Press, 1997).

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

[6] Bayen, A. M., Grieder, P., Meyer, G., and Tomlin, C. J., Lagrangian delay predictive model for sector-based air traffic flow, *AIAA Journal of Guidance, Control, and Dynamics* **28** (2005) 1015–1026.

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

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

[9] Bertsimas, D. and Stock-Patterson, S., The air traffic management problem with enroute capacities, *Operations Research* (1998).

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

[11] Bonaceto, C., Estes, S., Moertl, P., and Burns, K., Naturalistic decision making in the air traffic control tower: Combining approaches to support changes in procedures, in *Proceedings of the Seventh International NDM Conference* (Amsterdam, The Netherlands, 2005).

[12] Campbell, K., Cooper, W. J., Greenbaum, D., and Wojcik, L., Modeling distributed human decision making in traffic flow management operations, in *3rd USA/Europe Air Traffic Management R & D Seminar* (Napoli, 2000).

[13] Christodoulou, C., M.and Costoulakis, Nonlinear mixed integer programming for aircraft collision avoidance in free flight, in *IEEE MELECON*, Vol. 1 (Dubrovnik, Croatia, 2004), pp. 327–330.

[14] Donohue, G. L. and Shaver III, R. D., *TERMINAL CHAOS: Why U.S. Air Travel Is Broken and How to Fix It* (Amer Inst of Aeronautics and Astronautics, 2008).

[15] FAA OPSNET data Jan-Dec 2007, US Department of Transportation website (2007), (http://www.faa.gov/data_statistics/).

[16] G. Jonker, F. D., 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, 2007).

[17] Hill, J. C., Johnson, F. R., Archibald, J. K., Frost, R. L., and Stirling, W. C., A cooperative multi-agent approach to free flight, in *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems* (ACM Press, New York, NY, USA, 2005), ISBN 1-59593-093-0, pp. 1083–1090.

[18] Jennings, N. R., On agent-based software engineering, *Artificial Intelligence* **177** (2000) 277–296.

[19] McNally, D. and Gong, C., Concept and laboratory analysis of trajectory-based automation for separation assurance, in *AIAA Guidance, Navigation and Control Conference and Exhibit* (Keystone, Co, 2006).

[20] Nagel, K., Multi-modal traffic in TRANSIMS, in *Pedestrian and Evacuation Dynamics* (Springer, Berlin, 2001), pp. 161–172.

[21] Pallottino, L., Feron, E., and Bicchi, A., Conflict resolution problems for air traffic management systems solved with mixed integer programming, *IEEE Trans. Intelligent Transportation Systems* **3** (2002) 3–11.

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

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

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

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

[26] Tumer, K. and Agogino, A., Distributed agent-based air traffic flow management, in *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems* (Honolulu,HI, 2007), pp. 330–337.

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

[28] Watkins, C. and Dayan, P., Q-learning, *Machine Learning* **8** (1992) 279–292.

[29] Wolfe, S. R., Sierhuis, M., and Jarvis, P. A., To BDI, or not to BDI: design choices in an agent-based traffic flow management simulation, in *SpringSim '08: Proceedings of the 2008 Spring simulation multiconference* (ACM, New York, NY, USA, 2008), ISBN 1-56555-319-5, pp. 63–70, `http://doi.acm.org/10.1145/1400549.1400558`.

[30] Wolpert, D. H. and Tumer, K., Optimal payoff functions for members of collectives, *Advances in Complex Systems* **4** (2001) 265–279.