Adaptive Management of Air Traffic Flow: A Multiagent Coordination Approach

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

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

Abstract

This paper summarizes recent advances in the application of multiagent coordination algorithms to air traffic flow management. Indeed, air traffic flow management is one of the fundamental challenges facing the Federal Aviation Administration (FAA) today. This problem is particularly complex as it requires the integration and/or coordination of many factors including: new data (e.g., changing weather info), potentially conflicting priorities (e.g., different airlines), limited resources (e.g., air traffic controllers) and very heavy traffic volume (e.g., over 40,000 flights over the US airspace).

The multiagent approach assigns an agent to a navigational fix (a specific location in 2D space) and uses three separate actions to control the airspace: setting the separation between airplanes, setting ground holds that delay aircraft departures and rerouting aircraft. Agents then use reinforcement learning to learn the best set of actions. Results based on FACET (a commercial simulator) show that agents receiving personalized rewards reduce congestion by up to 80% over agents receiving a global reward and by up to 85% over a current industry approach (Monte Carlo estimation). These results show that with proper selection of agents, their actions and their reward structures, multiagent coordination algorithms can be successfully applied to complex real world domains.

Introduction

The efficient, safe and reliable management of our ever increasing air traffic is one of the fundamental challenges facing the aerospace industry today. On a typical day, more than 40,000 commercial flights operate within the US airspace (Sridhar *et al.* 2006). 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. In 2005, weather, routing decisions and airport conditions caused 437,667 delays, resulting in 322,272 hours of delays. The total cost of these delays was estimated to exceed three billion dollars by industry (FAA OPSNET data Jan-Dec 2005 2005). 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.

This paper presents a solution to this problem that is based on agents associated with a "fix," or a specific location in 2D. Because aircraft flight plans consist of a sequence of fixes, this representation allows localized fixes (or agents) to have direct impact on the flow of air traffic. In this approach, the agents' actions are to:

- set the separation between approaching aircraft;
- order ground delays; and
- reroute traffic

These simple agent-action pairs allow the agents to slow down or speed up local traffic, shift the burden from one region to another and prevent congestion from occurring. The first multiagent solution to this problem was presented in (Tumer & Agogino 2007), with extensions to multiple actions and coupled agent actions following in (Agogino & Tumer 2008).

Significance of Results

An adaptive, multi-agent approach is an ideal fit to this naturally distributed problem where the complex interaction among the aircraft, airports and traffic controllers renders a pre-determined centralized solution severely suboptimal at the first deviation from the expected plan. Though a truly distributed and adaptive solution (e.g., free flight where aircraft can choose almost any path) offers the most potential in terms of optimizing flow, it also provides the most radical departure from the current system. As a consequence, a shift to such a system presents tremendous difficulties both in terms of implementation (e.g., scheduling and airport capacity) and political fallout (e.g., impact on air traffic controllers).

The method summarized in this paper though focuses on a system that can be implemented readily. In addition the use of a commercial simulator (FACET) allows the domain experts to compare these results to current methods and removes a key barrier to the adoption of new innovative multiagent algorithms: a mismatch between the degree of fidelity expected by a domain expert and that provided by a multiagent researcher.

Air Traffic Flow Management

The management of traffic flow is a complex and demanding problem, where over 40,000 flights a day operate over the US airspace. 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. 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. The solution is therefore not directly affected by guidelines for separation assurance (2-30 minute decisions) and political and business concerns for airspace configuration (long term management). Instead our solution fits between long term planning by the FAA and the very short term decisions by air traffic controllers.

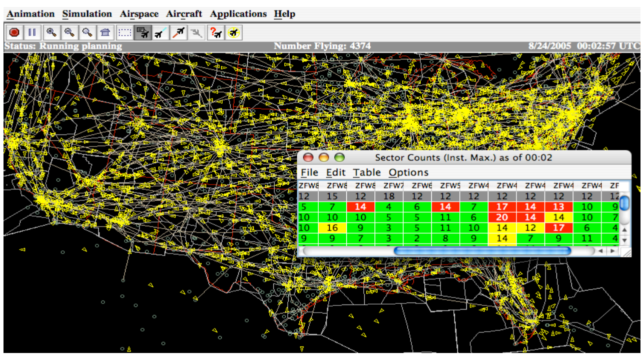

Figure 1: FACET screenshot displaying traffic routes and air flow statistics.

FACET

FACET (Future ATM Concepts Evaluation Tool), a physics based model of the US airspace was developed to accurately model the complex air traffic flow problem (Bilimoria *et al.* 2001). It is based on propagating the trajectories of proposed flights forward in time (Figure 1). FACET is extensively used by the FAA, NASA and industry (over 40 organizations and 5000 users) (2006 NASA Software of the Year Award 2006). In this paper, agents have FACET simulate air traffic based on their control actions. The agents then produce their rewards based on receive feedback from FACET about the impact of these actions.

Agent Based Air Traffic Flow

The multi agent approach to air traffic flow management we present is predicated on adaptive agents taking independent actions that maximize the system evaluation function discussed above. To that end, there are four critical decisions that need to be made: agent selection, agent action set selection, agent learning algorithm selection and agent reward structure selection.

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 multi-agent 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. 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, agents can collect data and readily match behavior to reward.
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.

Agent Action Sets

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.

Agent Learning and Reward Structure

In this paper we assume that each agent will have a reward function and will aim to maximize its reward using its own reinforcement learner (Sutton & Barto 1998). At every episode an agent takes an action and then receives a reward evaluating that action. It then uses this reward to update its action policy in such a way that it will try to take actions in the future that will lead to higher reward (for details see (Tumer & Agogino 2007)).

The system performance evaluation function focuses on delay and congestion. The linear combination of these two terms gives the full system evaluation function, $G(z) = -((1-\alpha)B(z) + \alpha C(z))$ as a function of the full system state z. where $B(z)$ is the total delay penalty for all aircraft in the system, and $C(z)$ is the total congestion penalty, which penalizes a system state where the number of aircraft in a sector exceeds the FAAs official sector capacity. The relative importance of these two penalties is determined by the value of α. (Details provided in (Tumer & Agogino 2007; Agogino & Tumer 2008)).

We explored three different reward functions for the agents. The first option was to let each agent receive the system performance as its reward. While this form of reward has been successfully used in small multi-agent reinforcement learning problems, it does not scale well, since the impact of a single agent's actions of the system reward is relatively small. To alleviate this problem, we explored 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 state/actions and aligned with the overall system reward (Tumer & Wolpert 2004; Tumer & Agogino 2007), given by:

$$D_i \equiv G(z) - G(z - z_i + c_i) \;, \qquad (1)$$

where z_i is the state of agent i. All the components of z that are affected by agent i are replaced with the fixed constant c_i. While the difference reward is effective in allowing an agent to see the impact of its own actions, 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. Our third reward is therefore an estimate of D that is computationally tractable and requires far fewer calls to the FACET simulator (one per time step, rather than one per agent).

Simulation Results

In all experiments we test the performance of four different methods. In addition to the three methods discussed above (G, D, D_{est}), we also provide a Monte Carlo estimation, where random policies are created, with the best policy being chosen. The next three subsections provide results for agents following the three actions: Miles in Trail, Ground Delay and Rerouting. The results are based for all three action sets are based on 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 take firm actions to minimize the congestion.

In all experiments the parameter for the tradeoff between congestion and lateness is set to $\alpha = 0.5$. 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.

Controlling Miles in Trail

In our first set of experiments, agents control Miles in Trail (MIT). 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.

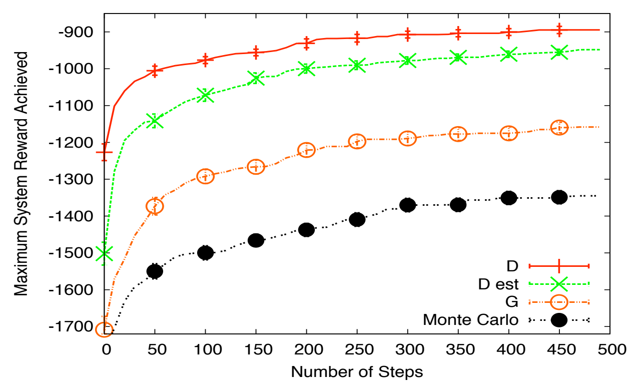

Figure 2: Performance for agents controlling miles in trail. 300 Aircraft, 40 Agents.

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. The results show (Figure 3) that the different rewards' performance is qualitatively similar to the case where agents control MITs. 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.

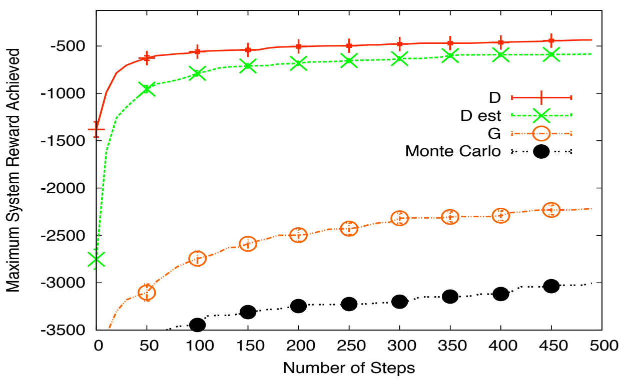

Figure 3: Performance for agents controlling ground delays. 300 Aircraft, 40 Agents.

Controlling Reroutes

In this experiment agents alleviate congestions by rerouting aircraft around congestions. Here an agent's action is 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.

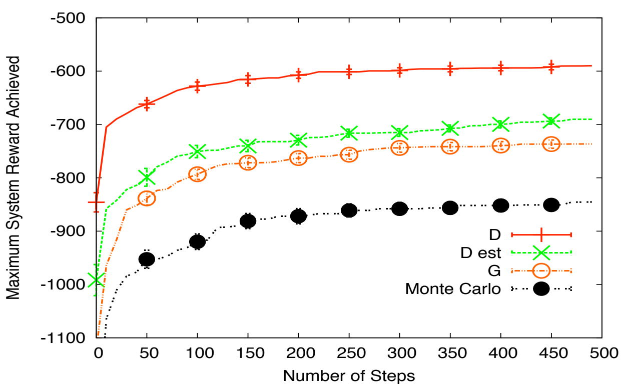

Figure 4: Performance for agents controlling rerouting. 300 Aircraft, 40 Agents.

Discussion

The main contribution of this paper is to present a distributed adaptive air traffic flow management algorithm that can be readily implemented and to test that algorithm using FACET, a simulation tool widely used by the FAA, NASA and industry. Air traffic management is a complex problem and requires new solutions that integrate policies with time horizons ranging from minutes up to a year. Our solution is based on agents representing fixes and having each agent taking one of three actions (setting miles in trail, ground delays or reroutes) for aircraft approaching its fix. It offers the significant benefit of not requiring radical changes to the current air flow management structure and is therefore readily deployable. The agents use reinforcement learning to learn control policies and we explore different agent reward functions and different ways of estimating those functions.

We are currently extending this work in four directions. First, we are exploring new methods of estimating agent rewards, to further speed up the simulations. Second, we are exploring the impact of agent coupling on system performance, where the actions of one agent restrict the actions of another agent (for example, setting ground delays can impact a reroute, or a reroute can impact miles in trail). Third we are investigating deployment strategies and looking for modifications that would have larger impact. One such modification is to extend the definition of agents from fixes to sectors, giving agents more opportunity to control the traffic flow, and allow them to be more efficient in eliminating congestion. Finally, in cooperation with domain experts, we are investigating different system evaluation functions, above and beyond the delay and congestion dependent G presented in this paper.

Acknowledgments: The authors thank Banavar Sridhar for sharing his insights on air traffic flow management, and Shon Grabbe for his help with FACET.

References

2006 NASA Software of the Year Award. 2006. FACET: Future ATM concepts evaluation tool. No. ARC-14653-1.

Agogino, A., and Tumer, K. 2008. Regulating air traffic flow with coupled agents. In *Proc. of the Seventh Joint Conf. on Autonomous Agents and Multi-Agent Systems*.

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

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

FAA OPSNET data Jan-Dec 2005. 2005. US Dept of Transportation. (http://www.faa.gov/data_statistics/).

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

Pechoucek, M.; Sislak, D.; Pavlicek, D.; and Uller, M. 2006. Autonomous agents for air-traffic deconfliction. In *Conf. on Autonomous Agents and Multi-Agent Systems*.

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

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

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

Tumer, K., and Agogino, A. 2007. Distributed agent-based air traffic flow management. In *The Conf. on Autonomous Agents and Multi-Agent Systems*, 330–337. **Best paper**.

Tumer, K., and Wolpert, D., eds. 2004. *Collectives and the Design of Complex Systems*. New York: Springer.