Addressing Hard Constraints in the Air Traffic Problem through Partitioning and Difference Rewards

William Curran
Oregon State University
Corvallis, Oregon
curranw@onid.orst.edu

Adrian Agogino
NASA Ames Research Center
Moffet Field, California
adrian.k.agogino@nasa.gov

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

ABSTRACT

In the US alone, weather hazards and airport congestion cause thousands of hours of delay, costing billions of dollars annually. The task of managing delay may be modeled as a multiagent congestion problem with tightly coupled agents who collectively impact the system. Reward shaping has been effective at reducing noise caused by agent interaction and improving learning in soft constraint problems. We extend those results to hard constraints that cannot be easily learned, and must be algorithmically enforced. We present an agent partitioning algorithm in conjunction with reward shaping to simplify the learning domain. Our results show that a partitioning of the agents using system features leads to up to a 1000x speed up over the straight reward shaping approach, as well as up to a 30% improvement in performance over a greedy scheduling solution, corresponding to hundreds of hours of delay saved in a single day.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

Keywords

Multiagent Reinforcement Learning, Air Traffic

1. INTRODUCTION

A primary concern facing the aerospace industry today is the efficient, safe and reliable management of our ever-increasing air traffic. In 2011, weather, routing decisions and airport conditions caused 330,063 delays, accounting for 266,999 hours of delay [1]. The rate of flights being scheduled is much faster than that of airports being built, making effective traffic control algorithms essential. We refer to the task of managing delay in the system by coordinating aircraft as the Air Traffic Flow Management Problem (ATFMP).

The national airspace (NAS) has many connections from one airport to another, therefore any congestion and associated delay can propagate throughout the system. Delays may be imposed to better coordinate aircraft and mitigate the propagation of congestion and the associated delay, but which aircraft should be delayed? The search space in such a problem is huge, as there are tens of thousands of flights every day within the United States.

We propose an a solution which blends multiagent coordination, reward shaping, automated agent partitioning, and hard constraint optimization. Multiagent coordination and reward shaping give us the ability to perform an intelligent guided search over tens of thousands of aircraft actions, while the hard constraint allows us to completely remove congestion. In the ATFMP, multiagent coordination with reward shaping becomes a computationally intractable task, therefore we used automated agent partitioning to reduce the overhead associated with the hard constraint.

2. RELATED WORK

The ATFMP addresses the congestion in the NAS by controlling ground delay, en route speed or changing separation between aircraft. The NAS is divided into many sectors, each with a restriction on the number of aircraft that may fly through it at a given time. This number is formulated by the FAA and is calculated from the number of air traffic controllers in the sector, the weather, and the geographic location. Additionally, each airport in the NAS has an arrival and departure capacity that cannot be exceeded. Eliminating the congestion in the system while keeping the amount of delay each aircraft small is the fundamental goal of ATFMP.

Difference rewards [3] function such that each agent's reward is related to the individual's contribution to team performance, leading to better policies at an accelerated convergence rate. The difference reward is defined as: $D_i(z) = G(z) - G(z - z_i + c)$, where z is the system state, z_i is the system state with agent i, and c is a counterfactual replacing agent i. This counterfactual offsets the artificial impact of removing an agent from the system.

3. HARD CONSTRAINT OPTIMIZATION

Our approach to traffic flow management involved three main concepts: formulating a multiagent congestion problem by defining agents, formulating the appropriate system rewards and reward shaping while blending the heuristic greedy scheduler with the multiagent learning, and performing hard constraint optimization using agent partitions.

Agent Definition: Agents were assigned to cooperating aircraft because they benefit from learning advantages. One of which is that each aircraft has its own learned policy, eliminating the need for a centralized controller. Another is that agents can be easily partitioned into independent groups, simplifying the learning problem. Agents have no

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013),* Ito, Jonker, Gini, and Shehory (eds.), May 6–10, 2013, Saint Paul, Minnesota, USA.

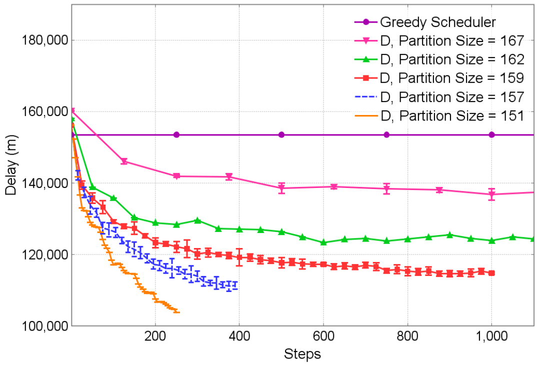

Figure 1: The difference reward performance using the smaller number of partitions shows a 30% improvement over the greedy scheduling solution.

state, and select a ground delay from 0 to 10 minutes. Agents learned using Action-Value Learning.

Reward Structures: Learning algorithms will not sacrifice delay to optimize congestion. To solve this problem we introduce a greedy scheduler. The greedy scheduler will convert sector congestion into a hard constraint, causing any amount of delay to achieve the goal. If any aircraft's flight plan violates the capacity constraint of any sector, it is forced to ground delay for 1 additional minute. Using the greedy scheduler forces congestion to become zero and therefore our system-level reward is: $G(z) = \sum_{a \in A} \delta_{a,g}(z) + \delta_{a,s}(z)$, where $\delta_{a,g}(z)$ is the ground delay incurred and $\delta_{a,s}(z)$ is the scheduler delay incurred.

The ATFMP has been previously analyzed using only a small time window. We wanted to approach this problem with a 14-hour window. This dramatically increases the number of agents from thousands to 35,844. Here it is difficult to achieve high performance without ensuring the agent's reward fully encapsulates the impact it had on the system. A difference reward reduces this noise, and is easily derived from the global reward: $D(z) = (-\delta(z)) - (-\delta(z - z_j + c_j)))$, where $\delta(z)$ is the cumulative delay in the system and $\delta(z - z_j + c_j)$ is the cumulative delay of with agent j replaced with counterfactual c_j.

Although the greedy scheduler is useful in removing congestion, it cannot optimize delay. Due to the congestion in the system, a delay of 0 for each aircraft would be suboptimal. The greedy scheduler will assign delay without taking into account agent coordination. Reinforcement Learning discovers a good ground delay for each aircraft, preventing the need to perform an exhaustive search.

This greedy scheduler can easily be combined with Reinforcement Learning. Agents can take an action, all agents actions can be modified using the greedy scheduler, reducing congestion to zero, and then agents will be rewarded based on the system after the greedy scheduler. When combining the difference reward with the greedy scheduler, there are severe computational issues. The difference reward requires an agent to be removed from the system, the greedy scheduler to reschedule all aircraft back into the system, and then compute the difference in delays. Rescheduling all 35,000 aircraft during each difference calculation makes this a computationally intractable solution.

Agent Partitioning: To reduce the computational overhead while computing the difference reward we reduced the number of aircraft the greedy scheduler had to reschedule by partitioning the aircraft into independent groups. The ATFMP has a clear partitioning of agents. Agents that do not go through the same sectors do not impact each other at all, and therefore can be treated as a smaller, more easily manageable, learning problem. We applied agglomerative hierarchical clustering [2] to partition similar agents together, resulting in a new reward for each partition i: $D_i(z) = (-\delta_i(z)) - (-\delta_i(z_i - z_{i_j} + c_j)))$, where $\delta_i(z)$ is the cumulative delay of partition i and $\delta_i(z_i - z_{i_j} + c_j)$ is the cumulative delay of partition i with agent j replaced with counterfactual c_j.

4. EXPERIMENTAL RESULTS

By adding the greedy scheduler we were able to eliminate congestion by sacrificing delay. Although high, this delay is required to guarantee a safe environment for all aircraft. As mentioned earlier, the greedy scheduler gives a good, but not optimal scheduling policy. This leads us to use the greedy scheduling policy as a good place to start searching. Bootstrapping each agent to choose 0 delay reduces the overall amount of computation time needed to compute D by giving the agents a good policy from the start. Agents can then explore other actions with a frequency based on ϵ and can discover potentially better actions.

Figure 1 shows the magnitude of performance gain when using the difference reward and partitioning. Since partitions had some overlap, actions in one partition may affect the agents in another partition, meaning that the higher the number of partitions the less information the agents receive in the difference reward. Smaller numbers of partitions end up leading to better overall performance at the cost of computation time.

5. DISCUSSION

The main contribution of this paper is to present a distributed adaptive air traffic flow management algorithm with implementable results. The method introduced is based on agents representing aircraft within the NAS choosing their own ground delay with the intent of minimizing delay within the system. It uses multiagent reinforcement learning in combination with the difference reward and hard constraints on congestion. This is typically an impossible problem, but we introduce agent partitions to dramatically reduce the time complexity by up to 1000x, leading to a 30% increase in performance over the greedy solution. Different sized partitions also allowed the implementation to be dynamic to the situation. The ease of adding ground delays in combination with the increase in performance over currently used approaches makes this approach easily deployable and effective.

Acknowledgements This work was partially supported by NSF grant CNS-0931591 .

6. REFERENCES

[1] FAA OPSNET data Jan-Dec 2011. US Department of Transportation website. (http://www.faa.gov/data_statistics/), 2011.

[2] William H.E. Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. In *Journal of Classification*, number 1 in 7-24, 1984.

[3] D. H. Wolpert and K. Tumer. Collective intelligence, data routing and Braess' paradox. *Journal of Artificial Intelligence Research*, 16:359–387, 2002.