

# Partitioning Agents and Shaping Their Evaluation Functions in Air Traffic Problems with Hard Constraints

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

Hundreds of thousands of hours of delay, costing millions of dollars annually, are reported by US airports. The task of managing delay may be modeled as a multiagent congestion problem with agents who collectively impact the system. In this domain, agents are tightly coupled, and the environment can quickly change, making it difficult for agents to assess how they impact the system. We combine the noise reduction of fitness function shaping, the robustness of cooperative coevolutionary algorithms, and agent partitioning to perform hard constraint optimization on the congestion and reduce the delay throughout the National Air Space (NAS). Our results show that an autonomous partitioning of the agents using system features leads to up to 540x speed over simple hard constraint enforcement, as well as up to a 21% 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

Aerospace Industry, Multiagent systems

## 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 [2]. Typical methods to alleviate delay involve imposing ground delay on aircraft, changing the speed of en route aircraft, and changing separation between aircraft. Because the airspace has many connections from one airport to another, the 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.

Our previous approach [1] blended multiagent coordination, fitness function shaping, automated agent partitioning, and hard constraint optimization. We extend that approach to use cooperative coevolutionary algorithms. We succeed in finding a solution which is robust to changes within the NAS while decreasing delay incurred by aircraft.

## 2. RELATED WORK

The air traffic flow management problem (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 congestion in the system while minimizing delay each aircraft incurs is the fundamental goal of ATFMP.

The difference evaluation function reduces noise from agents in the system [3]. This leads to final better policies at an accelerated converge rate. The difference evaluation function 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. APPROACH TO HARD CONSTRAINT OPTIMIZATION

Our approach to traffic flow management involved four main concepts: formulating a multiagent congestion problem by defining agents, formulating the appropriate system fitness function and fitness function shaping, and performing hard constraint optimization through the use of agent partitioning.

**Agent Definition:** Agents were modeled as aircraft with cooperation enforced by airport terminals. This decentralized solution benefits from many advantages. One of which is that each aircraft has its own 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 state, but may select a certain amount of ground delay from 0 to 10 minutes in the beginning of every simulation.

Agents evolved their policies using traditional cooperative coevolutionary algorithms (CCEAs). CCEAs are well

suited for multiagent domains where robustness is a desirable quality, and agents need to succeed, or fail, as a team. The random population member assignment generates a robust solution which is able to withstand sudden schedule changes that are frequent in air traffic.

**Fitness Evaluation:** The system-level evaluation focuses on the cumulative delay ( $\delta$ ) throughout the system. The greedy scheduler eliminates congestion by analyzing a planes flight plan and assigning extra ground delay if it causes congestion. A team of population members take an action, all actions are then modified using the greedy scheduler, and then each population member is assigned a fitness based on the joint actions. The delay is a linear combination of the amount of ground delay and the amount of scheduler delay an agent incurred:  $\delta(z) = \sum_{a \in A} \delta_{g,a}(z) + \delta_{s,a}(z)$ , where  $\delta_{g,a}(z)$  is the ground delay the aircraft incurred and  $\delta_{s,a}(z)$  is the scheduler delay the aircraft incurred.

We wanted to approach this problem with a 28-hour time frame. This increases the number of agents from thousands to 35,844. In this complex coordination domain, it is important to reduce the noise in the reward signal. A difference evaluation is easily derived from the system-level evaluation function:  $D(z) = (-\delta(z)) - (-\delta(z - z_j + c_j))$ , 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  $c_j$ .

**Agent Partitioning:** Combining the difference evaluation with the greedy scheduler increases computational complexity. The difference evaluation requires the greedy scheduler to reschedule all aircraft back into the system, and then compute the difference in delays. To reduce the computational overhead we reduced the number of aircraft the greedy scheduler had to reschedule by partitioning the aircraft into independent groups. Agents were partitioned together using hierarchical agglomerative clustering based on the number of similar sectors they had within their flight plan. This dramatically reduces the time complexity and allows the difference evaluation to be efficiently combined with the greedy scheduler. This results in a new fitness for each partition  $i$ :  $D_i(z) = (-\delta_i(z_i)) - (-\delta_i(z_i - z_{i,j} + c_j))$  where  $\delta_i(z_i)$  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

The experimental results in this paper analyzes the performance of the accumulation of different approaches. Since this is an extension of previous work, please see Curran et. al [1] for a more in-depth analysis on performance. In this paper, we analyze the quality of the partitioning based on size, time per simulation step and performance.

Speed and performance of partitions were negatively correlated. As the number of partitions were reduced, the greedy scheduler was required to schedule more planes per difference calculation. This greatly increased the amount of time per evaluation step. Additionally, with a smaller number of partitions, planes that slightly affected each other were partitioned together, leading to higher performance. This speed and performance correlation also occurs within the same partition. As the delay decreases, more agents choose a non-zero delay allowing better scheduling. This means that the agent no longer equals the counterfactual,

and the difference evaluation calculation cannot be skipped. Table 1 shows this in more detail.

The agent partitioning allows applications to become very situation dependent. If results need to be found very quickly, a larger number of partitions could be used, and this approach will find a policy still better than using the greedy scheduler. On the other hand if time spent is not important, the smaller number of partitions will result in a very good policy 21% better than the greedy scheduler.

Partitions	Time/run (s)			Steps	Final Delay (min)
	First	Avg	Last		
Greedy	5	5	5	*	169,750
1	3hr	*	*	*	*
21	116	364	402	2,000	134,050
24	64.8	155	173	2,000	139,630
29	37.1	89.2	100.2	2,000	145,630
38	23.3	46.9	52.38	2,000	154,770
56	16.6	26.6	29.2	2,000	162,710
91	13.5	20.3	22.6	2,000	167,990

**Table 1: With the greater number of partitions, the performance decreases and speed increases.**

## 5. CONCLUSION AND FUTURE WORK

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 cooperating aircraft within the NAS choosing ground delay with the intent of minimizing delay within the system. It uses CCEAs in combination with the difference evaluation 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 540x per evaluation step and a 21% increase in performance over the greedy solution. The different sized partitions also allowed the implementation to be dynamic to the situation. If results need to be computed quickly, a large number of partitions could be used, where a smaller number of partitions could be used if performance was more important than speed. The ease of adding simple ground delays in combination with the large increase in performance over currently used approaches makes this approach easily deployable and effective.

**Acknowledgments:** This work was partially supported by the National Science Foundation under Grant No. CNS-0931591

## 6. REFERENCES

- [1] W. Curran, A. K. Agogino, and K. Tumer. Addressing hard constraints in the air traffic problem through partitioning and difference rewards. In *Proc. of 12th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, May 2013.
- [2] FAA OPSNET data Jan-Dec 2011. US Department of Transportation website. (<http://www.faa.gov/data-statistics/>), 2011.
- [3] D. H. Wolpert and K. Tumer. Collective intelligence, data routing and Braess' paradox. *Journal of Artificial Intelligence Research*, 16:359–387, 2002.