A Centralized Multi-Agent Negotiation Approach To Collaborative Air Traffic Resource Management Planning

Peter A. Jarvis[*], Shawn Wolfe, Francis Enomoto, Robert Nado[**], and Maarten Sierhuis[*+]**

NASA Ames Research Center Moffett Field, CA 94035
[*]Craig Technologies, [**]Stinger Ghaffarian Technologies, [***]Carnegie Mellon University, [+]Delft University of Technology
Peter.Jarvis@nasa.gov, Shawn.Wolfe@nasa.gov, Francis.Enomoto@nasa.gov, Robert.Nado@nasa.gov, Maarten.Sierhuis@nasa.gov

Abstract

Demand and capacity imbalances in the US national airspace are resolved using traffic management initiatives designed, in current operations, with little collaboration with the airspace users. NASA and its partners have developed a new collaborative concept of operations that requires the users and airspace service provider to work together to choose initiatives that better satisfy the business needs of the users while also ensuring safety to the same standard as today. In this paper, we describe an approach to implementing this concept through a software negotiation framework underpinned by technology developed in the artificial intelligence community. We describe our exploration of peer-to-peer negotiation and how the number of conversation threads and the time sensitivity of offer acceptance led us to a centralized approach. The centralized approach uses hill climbing to evaluate airport slot allocations from a user perspective and a linear programming solver to seek solutions compatible across the user community. Our experiments with full sized problems identify the potential operational benefits as well as limitations, and where future research needs to be focused.

Keywords
Air Traffic Management, Multi-Agent Systems, Negotiation, Hill Climbing, and Linear Programming.

Introduction

Air traffic demand in the US national airspace (NAS) frequently exceeds the available capacity and such imbalances are projected to increase over the next 10 years (FAA 2006). In current operations, the Air Traffic Service Provider (ATSP) implements air traffic management initiatives (TMIs), such as ground delay programs, with minimal interaction with the airspace users. NASA and its partners have developed a new collaborative concept of operations (Idris 2005) for managing this resource problem. The concept poses substantial implementation challenges, as it requires a large number of self-interested organizations with complex utility functions to quickly find solutions within a safety critical and competitive commercial environment.

In this paper, we describe a framework for implementing the collaborative concept of operation's resource planning function that is based on prior multi-agent negotiation and intelligent search technologies (Ito et. al. 2007). Our experiments with full sized problems show the potential for significant benefits if the approach went operational today. We also identify a number of critical areas where future research should be focused to obtain more benefits.

The collaborative concept of operations covers the entire traffic management process from airspace problem identification, problem impact assessment, resource management planning, and flight implementation. We focus in this paper only on the resource management-planning phase. We describe the other processes and the artificial intelligence (AI) technologies applicable to them elsewhere (Wolfe et. al. 2008; 2009).

The organization of this paper is as follows. We first introduce the air traffic resource management problem and the motivation for taking an automated approach. We explain why we focused on negotiation in lieu of other agreement frameworks such as auctions. We then describe our exploration of both peer-to-peer and centralized negotiation approaches, justify why we settled on a centralized approach, and then describe that approach in detail. We document our full-scale experiments and conclude by describing the future research we are planning.

Airspace Resource Management Planning Problem

A large number of factors can reduce the capacity of the US national airspace including bad weather, air traffic control equipment failures, airport problems, security problems, and military training. As a result, the scheduled demand frequently exceeds the reduced capacity and TMIs need to be designed to address imbalances by throttling demand to enable flights to be safely operated (FAA 2007). The primary control points are the departure rates from airports and the rates on traffic flows between the airports.

Traffic flows together form a network of "highways in the air," within which major commercial air traffic is typically constrained to fly.

In this paper, we will consider only restrictions to airport departure rates, with the infrastructure for evaluating flow constraints under active development. The two control point types are interrelated. A flight must obtain both a departure slot and access to the necessary flows to get to its destination. The checking of flow constraints demands that flights be simulated to ensure that at no point the number on any given flow exceeds the flow rate. To check airport departure constraints, one needs to only sum the number of departures allocated to a time window. This is a limitation of our work reported in this paper. However, we argue the system that we have built can be readily extended within minimal impact on run-time.

Airport Departure Capacity Problem

We define the airport capacity problem as follows.

- the NAS has a set of airports A from which flights may depart (and arrive)
- each airport $a \in A$ has an departure capacity cap_{aw} during a time window w
- the set of flights F contains the flights scheduled to operate in the NAS during the day of concern
- the set of flights $F_{aw} \subseteq F$, where each $f_{aw} \in F_{aw}$ is scheduled to depart from airport a during time window w
- the total number of flights scheduled to depart from an airport a during a time window w, $|F_{aw}|$, must be less than or equal to the capacity of a during w - $|F_{aw}| \leq cap_{aw}$

In current practice, when $|F_{aw}| > cap_{aw}$, flights are delayed at a by adjusting their scheduled departure time to be after the time window w. This increases the demand in the following windows at a, often causing a propagation of delays during the day at an airport to ensure there is no window of time when demand exceeds capacity.

The ATSP normally plans in two-hour windows starting two hours after the current time and continuing up to twelve hours into the future. For example, if the current time is 0600 EST, the ATSP's first planning window will run 0800-0959 EST and second from 1000-1159 EST. Issues arising within the next two hours are not handled as part of the strategic planning process, but rather handled tactically. As the windows extend out in time, the fidelity and certainty of the data products (such as weather forecasts) used in the planning is reduced.

Variation in Flight Value and Opportunities for Collaboration

Commercial airspace users are heterogeneous and range from major passenger carriers with hub and spoke networks, to point-to-point regional carriers, to business jets flying small numbers of people, and to package haulers flying freight.

Interviews with airspace users identified that even within a category, individual users have very different business models and associated utility functions for evaluating the value of a given flight to their business. Many airport capacity problems occur in the NAS in a typical day and the value of a user's flights impacted within each can vary considerably, leading to opportunities for collaboration.

In the current practice in the United States, the ATSP applies a *ration-by-schedule* scheme to allocate reduced airport resources among the users. We use a coarser scheme derived from it that we call *ration-by-demand*, which allocates the reduced capacity to each user in proportion to the percentage of the scheduled nominal capacity that user had. For example, if a user had 50% of the 100 flights in the nominal capacity and the capacity was reduced to 10 flights, that user would retain its percentage and be allocated 5 flights. Our scheme does not specify the actual flights that will fly, as is the case in *ration-by-schedule,* but rather leaves that decision to the users.

While ration-by-demand allocations ensure an equitable reduction in capacity across the airlines, the scheme does not account for the distribution of flight values across the problems. Typically, the airspace users would like the opportunity to gain additional slots at airports where they have high value flights while accepting that they may have to relinquish slots at other airports where they have low value flights. If several users have complementary trades of this form, quid pro quo solutions can be established. The challenge we address in the remainder of this paper is to provide a framework in which the users can find these solutions in a reasonable time, without forcing another user to accept a solution that is worse for it than the ATSP's ration-by-demand allocation. Castro (2009) provides an excellent overview of airline scheduling and flight impact.

Selecting an Agreement Approach

The airport departure capacity problem cannot be solved centrally as it is impractical for the airspace users to share their sensitive, complex, and constantly evolving business utility models. The models contain sensitive commercial information such as the operations that are most profitable. These properties necessitate a distributed approach where each airspace user can keep its utility function private.

Sandholm (1999) and Wooldrige (2002) provide comprehensive surveys of the agreement technologies developed or extended within the AI multi-agent community. Our selection process went as follows:

- *Argumentation* technologies are not applicable because the airspace users have no interest in the reasons of competitors for wanting departure slots at specific airports. They operate within a highly competitive environment where there is no motivation to understand a competitor's needs and seek solutions to help satisfy them. Argumentation excels in a collegial environment.

- *Auction* approaches are not applicable as they are designed to maximize the price (green (actual) money or blue (synthetic) money) paid for a set of resources. We are not seeking solutions to determine the market value for a resource where the user who bids the most secures it, but rather solutions that guarantee no user receives a lower overall utility than in an initial allocation. It is also difficult to prevent a large user massing its resources in a local market to outbid a small regional user. Sheth and Gutierrez-Nolasco (2008) show how such auctions can be applied to airspace resource allocation.
- *Negotiation* frameworks offer the promise of the win-win solutions that we are seeking among airspace users.

We briefly explored a peer-to-peer negotiation approach based upon Fatima et al.'s (2006) work that was designed for negotiating between two agents. The airport problem, in contrast, involves tens of users. At the early prototyping stage we realized agents would have to engage in a large number of conversations on multiple issues (Jarvis et al. 2009). This led to a complex offer management requirement. A user may have a number of offers out that total more than the number of slots the user has available for trading, on the assumption that many offers will be rejected by other users. The distributed approach also permitted a pair of users to enter into an agreement that ruled out solutions for a third user. If the discussion had occurred in a different order, all users could have been satisfied.

These limitations led us to the centralized negotiation approach of Ito et al. (2007) designed specifically for domains with the properties of our airspace resource-planning domain where there are large numbers of agents and each has a non-linear utility function.

Centralized Negotiation Approach

The centralized negotiation approach has each agent develop and rank a number of more beneficial allocations from its perspective that are then passed to an independent arbiter agent who identifies the most mutually beneficial solution that is also consistent with the global constraints on the problem. We describe each of the three phases in turn. The entire process must be completed within a two-hour time frame to fit within the ATSP's planning process.

ATSP - Establishes Airport Problems and Slot Allocations

The ATSP publishes the set of airport problems, P_w, within the NAS for its current planning window w.

- for each airport problem $p_{aw} \in P_w$, the capacity constraint $|F_{aw}| \leq cap_{aw}$ is *violated* for the airport a during time window w
- the ATSP then allocates each airspace user, $u \in U$, that has flights in F_{aw} an allocation $allocation_{uaw}$ for each problem airport a using the *ration*-by-demand scheme, ensuring that $\sum_{u \in U} allocation_{uaw} \leq cap_{aw}$ at each problem airport.

With the problems defined and allocations made, the ATSP informs each $u \in U$ of the set P_w together with the cap_{aw}, and the $allocation_{uaw}$ at each airport. The users are not informed of other involved users or the allocation that each has been given.

Impacted Airspace Users - Identify Beneficial Slot Allocations

Each impacted airspace user (involved in at least one airport problem) performs this phase in parallel with the other impacted users. Each is instructed to have its proposals returned by a deadline to ensure inclusion in the search for compatible allocations. If a user does not return answers in time, its ration-by-demand allocation is used as its only proposal, ensuring that under our negotiation scheme, the user's utility will not decrease and that a tardy user does not delay the process.

Each user needs to evaluate the set of airport problems in which it is involved and identify alternative capacity distributions that are more or equally beneficial to it than the one proposed by the ATSP. Operationally, the users will develop their own methods for evaluating alternative allocations. Our task here is to develop an implementation to determine if it is feasible for an airline to perform this task and to experiment to understand the amount of time the process is likely to take.

Figure 1 presents the pseudocode for this phase. The substantial methods are the generation of alternative allocations and the evaluation of a given allocation's cost.

```
GetBetterCandiates($P$, $s$)
Input: A set of airport problems $P$
       The ATSP slot allocation $s$
Return: A set of better slot allocations $B$ for $P$ than $s$
  $B \leftarrow \{\}$
  $Q_s \leftarrow$ EvaluateAllocationCost($P$, $s$)
  $C \leftarrow$ GenerateAlternativeAllocations($P$, $s$)
  for $c \in C$
    if EvaluateAllocationCost(P, c) $\leq Q_s$ then
      $B \leftarrow B \cup \{c\}$
return B
```

Figure 1: Pseudo code for generating better candidate allocations

Generating Alternative Allocations

For a given user $u \in U$ and airport problem $p_{aw} \in P_w$, u will have $F_{uaw} \subseteq F_{aw}$ (possibly zero) flights scheduled at airport a during time window w.

- the *ration-by-demand* allocation of flights is denoted as $allocation_{uaw}$, where $allocation_{uaw} \leq |F_{uaw}|$
- the set of possible slot allocations L_{uaw} for a user u for a given airport problem $p_{aw} \in P_w$ is the set $\{0,.., |F_{uaw}|\}$

- the set of possible slot allocations Z_{uw} for u across all the airport problems $\{p_{a1w},\ldots, p_{anw}\}$ in P_w is $L_{ua1w} \times, \ldots, \times L_{uanw}$. The number of possible slot allocations is therefore $O(|P_w|^{\max|Faw|})$

We are only interested evaluating a subset of Z_{uw} that satisfy the following constraints:

- The number of flights at an airport a is $\leq cap_{aw}$
- The total number of flights in the candidate allocation is equal to the number of flights in the ATSP's ration-by-demand allocation. We refer to this as the *preservation of mass constraint* as users (at least in initial negotiations) do not want to consider the option of giving away any of their total allocation, just redistributing it.

In the problems we have encountered there has been an order of eleven airport problems in a planning window with the larger airlines having an average of ten flights impacted at each problem. This leads to 11^{10} candidate allocations, of which only approximately 0 to 200 satisfy both the capacity and preservation of mass constraints.

We have implemented a simple Cartesian product enumeration algorithm based on design guidelines in Skiena's manual (1997) to generate the candidate set, which we then prune with the constraints. This approach works within the computational demands resulting from a single planning window and the capability of modern hardware, but is an obvious candidate for improvement. We revisit this issue in the further work section.

Evaluate Allocation Cost

This method allows a user u to evaluate specific slot allocations, $z_{uw} \in Z_{uw}$, to determine its quality. At each airport problem, a user has $|F_{uaw}|$ flights scheduled and an allocation l_{uaw} from z_{uw}. When $l_{uaw} < |F_{uaw}|$ the user has a choice of which flights it delays and which it allows to fly. The user must therefore evaluate all k-subsets of the n-set of F_{uaw} where $k = l_{uaw}$ and $n = |F_{uaw}|$, which has $C(n,k)$ elements.

There are too many k-subsets to evaluate exhaustively. We applied *a hill-climbing* algorithm (Russell and Norvig 2003) that starts with a randomly generated k-subset and evaluates its quality. It then searches the neighboring k-subsets and either moves to one if it is of better quality, or terminates if a better neighbor cannot be found. We describe our evaluation of the performance of the hill-climbing algorithm in the experimental work section.

We implemented a schedule evaluation function based upon our interviews with domain experts in the airspace user community. The evaluation function measures the perturbation of the user's schedule of delaying a specific set of flights. Flights are connected by a number of dependency relationships including the aircraft, the cabin and flight crews, and the passengers and cargo. We record the set of constraints violated by a delay and use that as a guide to how hard the schedule will be to repair. We took this approach, as we need to evaluate many schedule impacts quickly. Constraint propagation can be completed quickly while schedule repair is most time consuming. We accounted for the different business models by allowing each user to weight the different impact factors differently.

Identify Consistent Set of More Beneficial Allocations: Trusted Third Party or ATSP

Each user passes a set of slot allocations B_{uw} that is at least as or more beneficial to it that the ATSP's *ration-by-demand* allocation to an independent third party agent (possibly the ATSP). The independent agent's task is to select a single $b_{uw} \in B_{uw}$ for each user that satisfies the capacity constraints at each airport problem. We cast this as a linear programming program that can be solved by software tools such as LPSolve (Berjekaarm & Van Eijk 2001), taking into account the following factors:

- a variable associated with each $b_{uw} \in B_w$, setting the linear programmer's task as to minimize the total cost of the sum of those assignments;
- a constraint for each airport with an airport problem that the total flights allocated during a time window must not exceed the capacity of each airport during that window;
- the capacity of each airport;
- the cost to the airline of each assignment together with the number of flights each assignment requires at each airport problem;
- one and only one assignment may be selected from each user.

Airspace User Schedule Data Setup

The airspace user schedules are central to determining the quality of slot allocations. We obtained the schedule for all domestic flights in the United States for a day in August 2008. The database contained a little over thirty thousand flights consisting of the departure and arrival airports, the scheduled departure and arrival times, and the type of aircraft servicing the flight.

We needed additional information on the actual aircraft, flights and cabin crews, and passenger connections flowing between flights to compute flight values. The airspace users would be reluctant to share this information with our current collaborative relationship. We were able to obtain a subset (three thousand) flights from the On Time Performance (OTP) database that the FAA recorded for that day, together with ticket data that captured the number of passengers connecting at airports. We used this data to generate the dependency information we needed. Our concern is that we do not have the same dependency structure as the actual schedules and this is an issue for future work. Despite this concern, we feel that when coupled with actual, full-sized schedules there is adequate fidelity to our experimental scenario that our results have operational significance.

	1000-1159 EST *4 Airport Problems*			1200-1359 EST *6 Airport Problems*			1400-1559 EST *8 Airport Problems*			1600-1759 EST *11 Airport Problems*		
Airline	-10%	-50%	-75%	-10%	-50%	-75%	-10%	-50%	-75%	-10%	-50%	-75%
L1	**12**	**65**	0	**23**	**23**	0	0	0	**1**	0	0	0
S1	**1**	0	0	0	0	0	**5**	0	0	0	0	0
L2	**2**	**14**	0	**30**	**16**	0	**2**	0	0	0	0	0
L3	**4**	**4**	**2**	**15**	**18**	0	0	0	0	0	0	0
L4	0	0	0	0	0	0	0	0	0	0	0	0
M1	0	0	0				0	0	0	0	**13**	0
S2	0	**8**	**8**	0	0	0	0	0	0	0	0	0
L5	0	0	0	0	0	0	0	0	0	0	0	**1**
S3	0	0	0									
M2	0	0	0	**1**	0	0	0	0	0	0	0	**3**
L6	**5**	**5**	0	0	0	0	0	0	**4**	0	0	**3**
L7	0	0	0	0	0	0				0	**3**	0
S4				0	0	0	**8**	0	0	0	0	0
S5				0	0	0				**31**	**2**	0
S6							0	0	0	**29**	**17**	0
M3										0	0	0
S7										0	0	0
M4										0	0	0
M5										0	0	0
M6										0	0	0
S8										0	0	0
S9										0	0	0

Table 1: Improvement percentage by user of the negotiated allocation over the ration-by-demand allocation

Experimentation

To determine the overall performance of the system, we took the weather events for a day that would impact the departure capacity of airports in the NAS and then artificially varied their intensity. The result was three different cases where capacity is reduced by 10%, 50%, and 75%. We then applied these reductions to the first four planning windows in the day. The scope of the weather increased over the day to provide a structure where initially only 4 airports and 12 users were impacted while by the fourth planning period 11 airports and 22 users were disrupted. The user names have been anonymized and replaced with the labels *L* for large, *M* for medium, and *S* for small users.

Table 1 shows the percentage improvements of the negotiated solutions that were found over the ATSP's ration-by-demand allocation across the scenarios we setup. Even small percentage improvements equate to tens of flights and hundreds of passengers arriving on time rather than being delayed. Grey indicates that an airline had no scheduled flights at a problem airport during the time window.

The system was able to explore all the options in the first two planning windows within a few minutes (well within the hour limit). In the latter two, the number of alternative slot allocations reached as high as 400 for some users, and the options took as much as 19 hours to evaluate. The options are independent and can therefore be explored concurrently on different CPUs, with around 20 CPUs being required for an airline to meet the one-hour deadline that we have imposed on the process. Users can also return the options they have had time to evaluate by the deadline.

At the component level, the LP solver quickly solved the problems, returning results within ten seconds. The hill-climber took the majority of the time, and was especially sensitive to the number of flights as that increased the number of neighbors that had to be evaluated for a given search node. We carefully evaluated the number of times the hill-climber would restart. Allowing it to run for several days on problems revealed that improvements were rarely found after the fifth restart and none were found after the twelfth. We therefore allowed the hill-climber 50 restarts when assessing the quality of the ATSP's ration-by-demand solution and 10 when considering candidate user allocations. It is essential that the quality of the

ATSP's allocation be not overestimated as it is used to filter all the succeeding candidates. This 50/10 balance provides the best trade-off between speed and ensuring a thorough analysis of the ATSP's solution.

Conclusions and Further Work

We have implemented and evaluated a software framework for realizing NASA's collaborative concept of operations resource planning process. Our implementation is based upon a centralized multi-agent negotiation framework developed in the artificial intelligence community. Our evaluation of full sized problems has revealed significant benefits that can be obtained.

Future work is needed to address a number of issues. The Cartesian product algorithm used to create candidate slot allocations takes a generate-and-test approach. An approach that generates only the subset that satisfies the constraints we post on alternatives will allow the possibility of trades to be considered across planning windows. Currently, the number of alternatives generated across more than one planning window exceeds the memory addressing space of a 32-bit Java virtual machine.

The hill-climbing local search used to evaluate slot allocations is consuming the majority of the system's runtime. One possibility is to apply the approach of Hattori et al (2007), using clustering techniques to allow the allocations to be examined at a coarser level and requiring that extensive search only be conducted within clusters likely to contain good solutions.

To address scenario fidelity, we plan to integrate with the flow rate test-bed under development by our collaborators that will allow us to use the FACET (Bilimoria et al 2001) airspace simulator to check flow rate constraints. We would like to use actual flight interdependency data instead of our current synthesized data. This will necessitate an information sharing agreement with the airspace users that ensures the protection of their commercially sensitive information.

References

Berjekaarm, M., and van Eijk, K. 2001. Efficient and Effective Redundancy Removal for Million-Gate Circuits. In *Proceedings of the international workshop on Logic Synthesis*. Lake Tahoe, CA.

Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., and Grabbe, S., 2001. FACET: Future ATM Concepts Evaluation Tool. *Air Traffic Control Quarterly*, 9(1) pp.1-20.

Castro, A., 2009. *Disruption Management In Airline Operations Control: Designing a Multi-Agent System with a GAIA Based Methodology*. VDM Verlag.

Fatima. S., Wooldridge. M., Jennings. R., 2006. Multi-Issue Negotiation With Deadlines. In *Journal of Artificial Intelligence Research*. 27:381-417.

Federal Aviation Administration. 2006. FAA Long-Range Aerospace Forecasts Fiscal Years 2015, 2020 and 2025.

Federal Aviation Administration, 2007. Traffic Flow Management in the National Airspace. FAA National Headquarters, 800 Independence Avenue, Washington DC.

Hattori, H., Klein, M., and Ito, T. 2007. Using Iterative Narrowing to Enable Multi-Party negotiations on Multiple Interdependent Issues. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent System*. Hawaii, USA.

Idris, H., Vivona, R., Penny, S., Krozel, J., & Bilimoria, K. (2005). Operational Concept for Collaborative Traffic Flow Management based on Field Observations. In *proceedings of the AIAA 5th Aviation, Technology, Integration, and Operations Conference (ATIO).*

Ito, T., Hattori, h., and Klein, M., 2007, Multi-issue negotiation protocol for Agents: Exploring nonlinear utility Spaces. In *Proceedings of the Twentieth International Jon Conference on Artificial Intelligence*. Hyderabad, India.

Jarvis, P., Wolfe, S., Sierhuis, M., Nado, R., Enomoto, F. 2009. Agent-Based modeling and Simulation of Collaborative Air Traffic Flow management using Brahms. In *Proceedings of the SAE 2009 AeroTech Congress*. Seattle WA, USA. SAE International.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Sandholm T., 1999. *Distributed rational decision making*. *Multi Agent Systems*, pp.201-258, MIT Press.

Skiena., S, 1997. *The Algorithm Design Manual*. Springer Verlag, ISBN 0-387-94860-0.

Sheth., K., Gutierrez-Nolasco., S., 2008. Incorporating User Preferences in Collaborative Traffic Flow Management. In *proceedings of the American institute of Aeronautics and Astronautics (AIAA) Guidance Navigation and Control (GNC) Conference*, Honolulu, HI.

Wolfe, S., Sierhuis, M., and Jarvis, P. 2008. To BDI or Not to BDI. In *Proceedings of the 2008 Agent-Directed Simulation Symposium (ADS'08)*, Ottawa Canada

Wolfe, S., Jarvis, P., Enomoto, F., Sierhuis, M., Putten, B., 2009. A Multiagent Simulation of Collaborative Air Traffic Flow Management. In *Edited Collection on Multi-Agent Systems for Traffic and Transportation*. MAS-TT Book

Wooldridge, M. 2002. *An Introduction to Multi Agent Systems*. Wiley.

Acknowledgements: The authors would like to thank Karl Bilimoria, Antonio Castro, Jeremy Frank, Maria Fox, Jimmy Krozel, Kapil Sheth, American Airlines, Business Jets, Delta Airlines, TAP Portugal, UPS, Virgin America, and employees of the Federal Aviation Administration for providing us with the domain knowledge we needed to support this research.