

AIR TRAFFIC CONFLICT NEGOTIATION AND RESOLUTION USING AN ONBOARD MULTI-AGENT SYSTEM

Jie Rong, John Valasek, Shijian Geng, and Thomas R. Ioerger
Aerospace Engineering Department, and Computer Science Department
Texas A&M University, College Station, TX 77843-3141
valasek@aero.tamu.edu, ioerger@cs.tamu.edu

Abstract

Conflict detection and resolution is a critical capacity for realizing free flight. In this paper, an onboard multi-agent system algorithm is proposed to solve the air traffic conflict detection and resolution problem involving multiple aircraft, where each aircraft may be regarded as an intelligent agent. Multilaterally acceptable resolutions to conflicts are solved as a constraint satisfaction problem, using a pair wise, argument-based negotiation approach. Ground based air traffic control is included in this system, acting as a high-level supervisor and coordinator. It has authority to approve or override proposals from any aircraft; force an aircraft to accept a proposal; and to open or close a negotiation. It may also join the collaboration to represent those aircraft which lack the capacity to negotiate for themselves. Real-time constraints such as response time and negotiation deadline are included in the proposed algorithm. Results for a head-on conflict scenario demonstrate that the proposed negotiation system and algorithm is capable of providing mutually acceptable solutions.

Introduction

Free Flight, also called user preferred traffic trajectories, is an innovative concept designed to enhance the safety and efficiency of the National Airspace System (NAS). It moves the NAS from a centralized command-and-control system between pilots and air traffic controllers, to a distributed system that allows pilots, whenever practical, to choose their own route and file a flight plan that follows the most efficient and economical route [1]. Implementation of Free Flight, which offers benefits in system safety, capacity, and efficiency, is critical to advancing aviation by accommodating

the nation's growing airspace needs. Free Flight is potentially feasible due to the contemporarily advanced technologies like Global Positioning System (GPS), datalink communications such as Automatic Dependence Surveillance-Broadcast (ADS-B), and enhanced computational capacity in cockpits.

Practical realization of free flight relies greatly on an effective, efficient, and feasible Conflict Detection and Resolution (CD&R) algorithm. Ideally, all aircraft occupying a given airspace would share current information on their intent, the overall air traffic situation, weather conditions, and terrain variations. The current information would be sent to a CD&R algorithm embedded in the Flight Management System (FMS) of each aircraft. It will first determine whether or not potential conflicts exist. Potential conflicts may be classified into three primary categories: weather, traffic and terrain. If certain conflicts are detected, the CD&R algorithm is responsible for computing and providing optimal and conflict-free flight path guidance after processing all of the relevant information.

CD&R has been an active research topic in the aviation community in recent years, e.g. [2,3,4]. However, very little of this contemporary research makes use of intelligent Multi-Agent Systems (MAS). Wangermann and Stengel introduce an Aircraft/Airspace System (ASS) as a MAS, which includes several agent types: aircraft; operators; and traffic management [5]. Principle negotiation between these agents is proposed as the coordination method to avoid conflicts and ensure safety within the ASS. The agents are also provided with greater freedom to optimize and overcome problems. It is assumed aircraft only negotiate with ground controllers, and not each other. Within this arrangement, multiple aircraft

encountering one or more conflicts may be regarded as a Multi-Aircraft Agent System (MAAS), in which each aircraft is itself an independent agent [6]. The environment of each aircraft includes the adjacent weather conditions, other aircraft in its neighboring airspace, air traffic ground controllers, etc., as shown in Figure 1. The conflict detection and resolution problem may be solved as a constraint satisfaction problem. An aircraft should always fly along a path that satisfies all of the constraints posed by its environment. For instance, avoiding the protected zones of other aircraft, the dangerous areas of squall lines, special usage airspace, and terrain. Among all possible flight paths, an optimal one is desired that produces the lowest cost in terms of time of flight, fuel consumption, passenger cost, etc. As the overall

safety of the airspace is a global concern, collaboration among aircraft is required to search for appropriate solutions to conflicts. Thus negotiation among aircraft in the airspace becomes a critical factor for multilateral agreement of the ultimate conflict resolution. Of course, negotiation between aircraft only occurs when traffic conflicts exist, since weather and terrain do not negotiate. Toward this goal, Harper et. al. obtained some preliminary results for negotiation among aircraft using the principle negotiation method [7]. In this paper, a pair wise, argument-based negotiation approach is established to search for multilateral acceptable resolutions to conflicts. In addition, real-time constraints such as response time and negotiation deadline are considered.

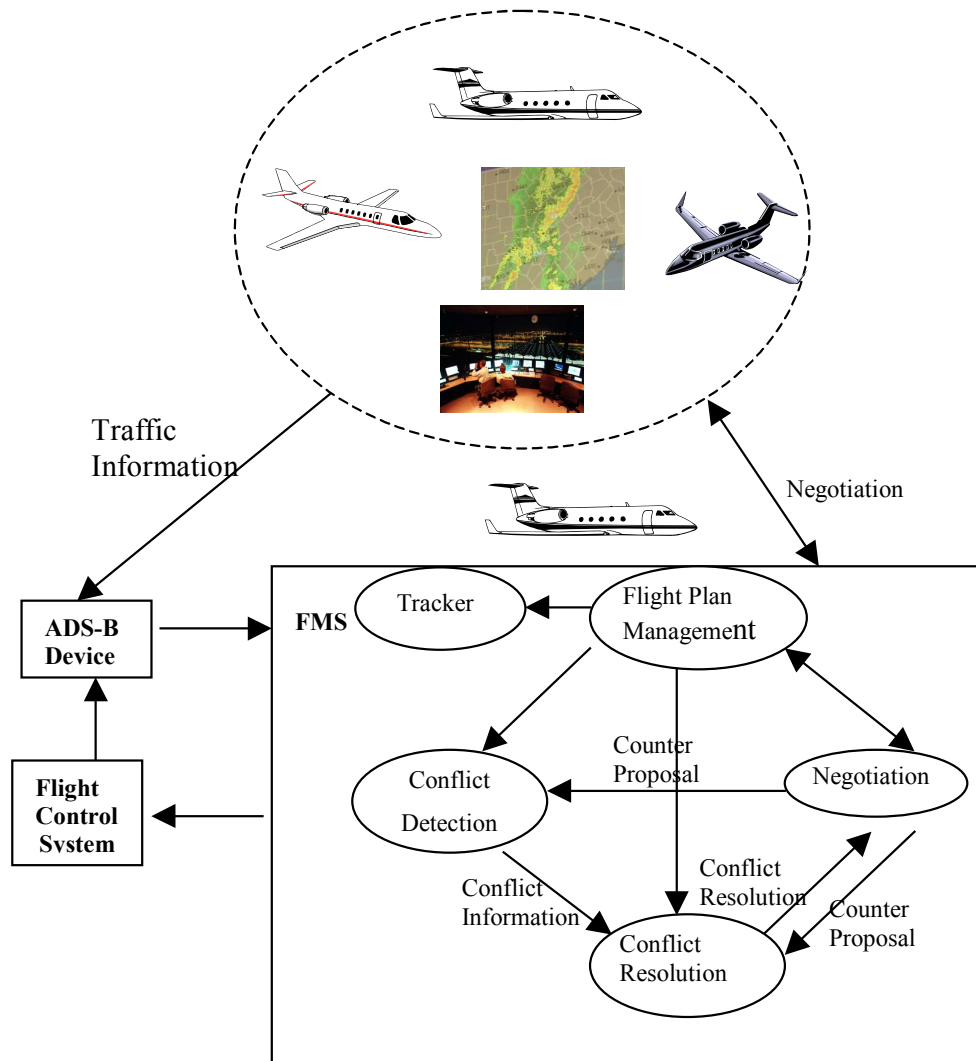


Figure 1. Overall Structure for Multi-Agent System

Overall System Design

Definitions

The following definitions are used in the Multi-Aircraft Agent System (MAAS):

- A flight plan is the sequence of several segments.
- Each segment is composed of starting waypoint, ending waypoint, altitude, and airspeed.
- A proposal is some kind of solution to the problem that the agents face. In the MAAS, it implies an alternative flight plan to an existing flight plan that avoids all the potential conflicts.

Assumptions

The following assumptions are used in the design of the MAAS:

Aircraft Class:

The system developed in this paper is initially designed and implemented for FAR 23 Normal Category General Aviation (GA) aircraft under Free Flight conditions. However, the MAAS is not restricted only to GA aircraft. It can also be extended without significant modification to high performance FAR 23 Class aircraft such as business jets, and FAR 25 Class aircraft such as commuters and commercial air transports. Currently, most FAR 23 Class Normal Category aircraft are not equipped with the navigation and communication devices commonly found on FAR 25 Class aircraft. As the agent system developed in this paper is designed for an onboard, future implementation of free flight, it is assumed that this advanced equipment will be available on GA aircraft at that future time. This research therefore assumes each aircraft to be equipped with the following onboard devices: ADS-B device, to permit assessment of the current traffic situation; a Flight Management System (FMS), for performing conflict detection and resolution, and executing the current flight plan; and a communications suite capable of negotiating with other aircraft and the ground based aircraft controller. Weather or terrain data is assumed to be provided by data link from ground services, or an onboard radar.

Degree of Pilot Autonomy:

Harper, Mulfund, et. al. [7] present definitions of degrees of pilot autonomy in Free Flight, as shown in Table 1. In this paper, only aircraft with Autonomy Level 3 are considered, and Air Traffic Control (ATC) is excluded from the negotiation process, since it is considered a special agent which is different from the others. However, ATC is always included in the system, acting as a high-level supervisor and coordinator with the authority to approve or override any proposal from any aircraft. It can also force an aircraft to accept an ATC proposal, and open or close a negotiation.

Table 1. Degree Of Pilot Autonomy

Degree	Level of Autonomy
1	Standard ATC. Pilots act as instructed by ATC
2	Pilot is free to search for and negotiate potential solutions with other pilots of level 2 or higher and with ATC, and implement the resulting globally approved actions
3	Pilot is free to search for and negotiate potential solutions with other pilots of level 2 or higher, and pose solutions and ATC for approval before implementation

Mixed-Fleet Environment:

Realistically, not every aircraft in the airspace is guaranteed to be able to negotiate for itself. In these cases, ATC should join the collaboration as the representative for these aircraft, and obey the same game rules as other aircraft attending the negotiation. Aircraft which lack the capacity to collaborate fly with Autonomy Level 1. In addition, some subset of the aircraft may be equipped with more advanced onboard devices such as onboard weather radar and/or Traffic Alert and Collision System (TCAS). Thus the capacities of obtaining information about the air traffic environment can be different among aircraft. Since the current implementation of the system is designed primarily for GA aircraft, scenarios such as conflicts between a GA and a commercial air transport are beyond the scope of the current paper. All aircraft are given the same responsibility for assuring the airborne safety in their neighboring airspaces, and no aircraft is permitted to have a

higher priority than any other aircraft. This means emergency modes are not considered.

Agent Properties:

- The MAAS is a non-deterministic, mixed event, dynamic system, with N agents. No agent has perfect knowledge of the system.
- Agents in the MAAS are maximizing agents, searching for flight plans not only acceptable with respect to some constraints, but also optimal with respect to some cost function [7].
- Agents in the MAAS behave rationally, only submitting proposals which satisfy all constraints and decrease cost.
- Agents in the MASS behave cooperatively, only submitting proposals which satisfy all of the known constraints of other agents. They minimize their own cost under the condition that the cost of the part of the system they are involved with is a minimum. They never cheat other agents for their own interest.

Conflict Detection And Resolution Module

For each agent in the MAAS, its conflict detection and resolution problem is solved as a constraint satisfaction problem. In a multi-agent system, negotiation is an essential form of interaction that enables groups of agents to arrive a mutual agreement. A pair-wise, argumentation-based negotiation approach is initially established for the aircraft to search for multilaterally acceptable conflict resolutions. Admittedly, it is not certain that particular multi-way conflicts could be resolved by composing a series of pair-wise negotiations since there are some 3-way conflicts that won't be resolved, e.g. a circularity: A&B, B&C, C&D, D&A. However, multiple pair-wise negotiations may be interleaved to guarantee jointly acceptable solutions among situations involving more than two aircraft. Consider that random groups of multiple aircraft fleetingly come into and go out of potential conflict with each other. Any solution should consider them all, or at least all aircraft must come to a mutually acceptable agreement that is safe (and efficient) for them, even if they "chain" together. For example, if Aircraft A

conflicts with Aircraft B, and Aircraft B with Aircraft C (but not Aircraft A with Aircraft C), and then Aircraft A and Aircraft B determine a "solution" between them, the authors do not consider it a solution unless it is also jointly a solution with Aircraft C. Often, this may involve subsequent and interleaved negotiations with multiple partners (e.g. Aircraft B to Aircraft A, and Aircraft B to Aircraft C). This would be a classic DCSP, but state-of-the-art DCSP algorithms are not really designed for dynamic joining and dropping of participants (and constraints), and therein lies the research challenge.

The argumentation-based protocol is designed for agents to incrementally reveal the parts of their constraints/desires that become relevant to finding a mutually acceptable solution. A priori, the aircraft involved in a conflict do not know each other's desires/constraints, and may not be able to exchange all of this information all the time, with every other aircraft. Hence, when Aircraft 1 proposes a solution, and Aircraft 2 cannot accept it, it does not just send a rejection to Aircraft 1, but rather sends a revised solution annotated with whatever private constraint was violated. Now Aircraft 1 has gained information, so the process (usually) does not stalemate. Figure 2 illustrates the logic flow for a conflict detection and resolution module in an agent. Based on the inputted environment information, the agent first checks to confirm that its current flight plan satisfies all of the constraints. If non-traffic conflicts are detected, a new conflict free flight plan is searched for. If traffic related conflicts are detected, negotiation between aircraft is necessary to search for mutually acceptable flight plans for each aircraft involved. If no solution is found after the searching process, the agent turns to ATC for help, asking for a new flight plan. Otherwise, the new solution requires approval from ATC. Should ATC reject the solution, it provides the reasons along with the rejection message so the agent may add the new constraints to its constraint set. It then starts searching for a new plan with respect to its new constraints. If the agent is not able to find a solution before it reaches the deadline, ATC provides one solution and forces the agent to accept it. Finally, the aircraft executes the new flight plan. The entire iterative process continues throughout the flight.

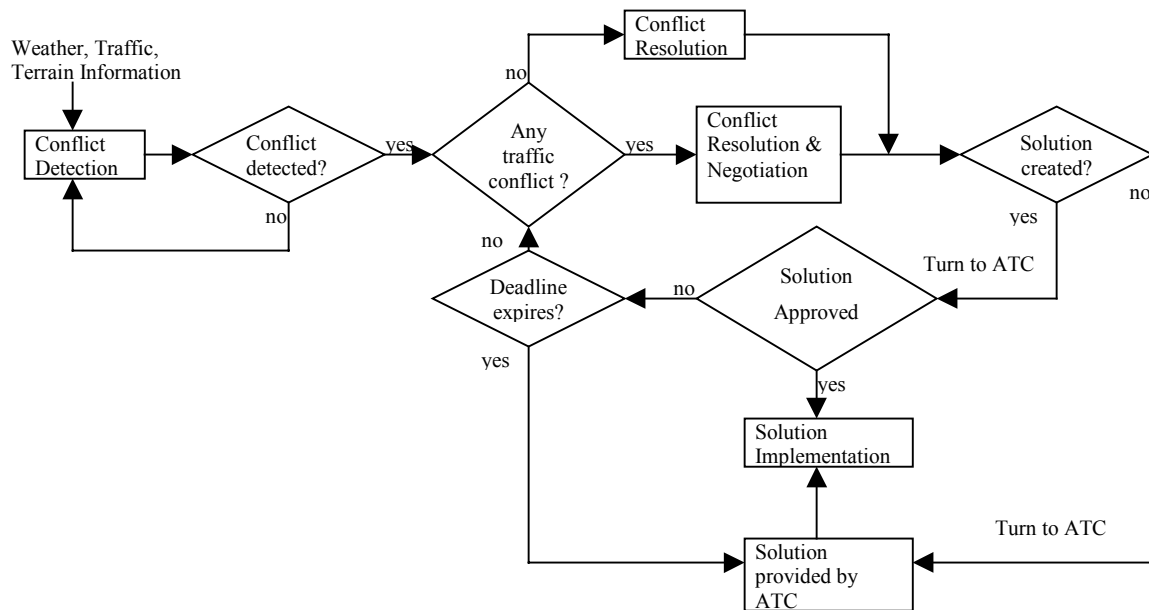


Figure 2. Logic Flow Control for Conflict Detection and Resolution

Negotiation Protocol

These are the rules the agent adheres to in a negotiation process. A state-based negotiation protocol is created for the MAAS. For one particular agent, it uses the constraint satisfaction problem algorithm to seek conflict resolution. In the free-flight application domain, one aircraft may confront more than one traffic conflict, which may be involve two or more aircraft. A pair-wise negotiation strategy is employed for such situations, which implies that negotiation is only conducted between two agents. This strategy is different from the principle negotiation method presented in [5,7].

Conflict Detection

Weather conflict refers to any destructive weather phenomenon that may pose danger to the aircraft safety, such as thunderstorms, tornadoes or squall lines. Terrain conflict refers to ground obstructions of any type that may lead to Controlled Flight Into Terrain (CFIT). These two conflicts along with Special Usage Airspace (SUA) may be represented as restricted areas in a three dimensional airspace. If the flight plan of an aircraft leads it into the above forbidden areas, a potential conflict is considered to occur and a new flight plan is required to avoid the conflict.

The traffic conflict detection takes ADS-B state vectors of other aircraft in the immediate airspace as input, which includes the current states of an aircraft like its position, airspeed, altitude, etc. The ADS-B also contains the intent information of the aircraft, such as its flight plan. Safe airborne separation between aircraft is maintained using protected and alert zones, as illustrated in Figure 3. The protected zone of an aircraft surrounds the aircraft and should never overlay with another aircraft's protected zone. The alert zone surrounds a larger area, and the aircraft can maneuver freely until its alert zone contacts another alert zone [1]. The size of the zones is determined by the velocity, performance, communication, navigation and telecommunication equipment of the aircraft. The agent performs a pair-wise search between current states and flight plans of its own aircraft and any other bogey aircraft in its alert zone. A traffic conflict is assumed to take place when the estimated flight trajectory of an aircraft passes through the protected zone of another aircraft.

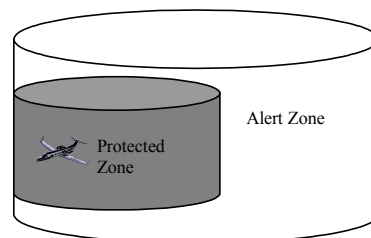


Figure 3. Protected Zone and Alert Zone

Resolution Searching Algorithm As Constraint Satisfaction Problem:

Yokoo et. al. develop and formalize the Distributed Constraint Satisfaction Problem (DCSP) in [8]. DCSP is a general framework that can formalize various application problems into Artificial Intelligence problems. Formally, a CSP consists of the following two components:

- N variables x_1, x_2, \dots, x_n , whose values are taken from finite, discrete domains D_1, D_2, \dots, D_n respectively
- A set of constraints on their values, which a constraint is defined by a predicate. That is, the constraint $P_k(x_{k1}, x_{k2}, \dots, x_{kn})$ is a predicate that is defined on the Cartesian product $D_{k1} \times \dots \times D_{kn}$. This predicate is true (satisfied) if and only if the value assignment of these variables satisfies this constraint. Solving a CSP is equivalent to finding an assignment of values to all variables, such that all constraints are satisfied.

In free-flight, the goal of the conflict resolution process is determining a new flight plan which avoids any potential conflict. Therefore, the flight plan is the only variable defined for the conflict resolution. All of the severe weather conditions, obstructing terrain, special usage airspace, and the flight plans of other aircraft are defined as several constraint predicates of the flight plan. Instrument Flight Rules (IFR) maneuvering is assumed in this paper, so the following constraints are imposed: heading change less than 30 degrees; velocity increases of less than 3% per minute, and velocity decreases of less than 10 % per minute. Other applicable constraints are related to aircraft configuration and performance, such as the dive speed V_D . The potential resolution is a flight plan that satisfies all of the constraints in its constraint set.

The resolution space is a set consisting of all the resolution candidates for a particular CSP. The original resolution space, the set of all potential flight plans for an aircraft, is a continuous, infinite one. To convert the free-flight application into a CSP, this continuous and infinite set must be formalized into a discrete, limited domain. The

basic idea of the formalization is to divide the continuous space into several intervals according to a particular unit. Generally, three kinds of conflict resolution maneuvers are employed – turning right or left, accelerating or decelerating, and ascending or descending. For example, the turning angle of an aircraft may be divided into discrete increments, so that a limited, discrete set for the angle factor is obtained. For speed and altitude modification, the units may be set as every one-percent of the current airspeed, and every 100 feet. A discrete time set (ten seconds) is required to represent the starting and ending moments of the maneuvers. A flight plan may be defined as some combination of the three maneuvers, with the starting and ending times specified. Finally, a discrete, limited domain is obtained for the variable after the formalization. The resolution searching is executed in this domain.

An agent should be a maximizer, selecting the optimal flight plans among all flight plans satisfying the constraints. For an aircraft, several factors are considered in defining a cost function: arrival time, fuel consumption, direct operating cost, and passenger comfort [7].

Pair-Wise Strategy For Group Conflicts:

Determining which agents should attend which negotiations is often difficult and sometimes impossible to determine. For example, if agents A, B, and C are involved in one traffic conflict, they may form one negotiation. However, if there is one conflict between A and B, and another conflict between A and C but no conflict between B and C, is it necessary for C to join the negotiation between A and B? What about the situation when C is also involved with another agent, D, that has no conflict with either A or B. Are one, two, or three negotiations necessary? For simplicity, a pair-wise strategy is applied in this paper, which means that negotiation only occurs between two agents involved in the same conflict. If an agent confronts conflicts with more than one other agent, it takes part in several negotiations simultaneously.

Negotiation Protocol:

Negotiation is a process that takes place between two or more agents who are attempting to achieve goals that they cannot, or prefer not to, achieve on their own. These goals may conflict, in which case the agents have to bargain about which agent achieves which goal, or the agents may

depend upon one another to achieve the goals, in which case they only have to discuss how to go about achieving the goals. In either case, the process of negotiation proceeds by the exchange of proposals, critiques, explanations, and meta-information.

In the argument-based negotiation, agents try to persuade their conflict acquaintances to agree with their proposals [9]. The argumentation consists of the following four elements:

- Claim: a conclusion that an agent establishes and tries to persuade others to agree with
- Support data: the facts that are a foundation for the claim
- Warrants: the reasoning mechanism from the data to the claim
- Qualification: the degree of acceptance of the claim, based on the data and warrants

Considering two aircraft, a negotiation occurs only when a traffic conflict between them is detected. Before negotiation, the agent who first detected the conflict sends a greeting message to notify the agent in the other aircraft of the conflict. The content of the greeting message includes the aircraft's identification, information from various onboard devices, etc. The greeting process confirms the traffic conflict and determines which one of the two should be the initiator, the first agent sending the proposal. Generally, the agent first detecting the conflict is the initiator. However, sometimes the aircraft with more advanced onboard devices act as the initiator, so that the negotiation process becomes more efficient. When the initiator receives a counter-greeting message from the other agent, the negotiation starts and the initiator calls the conflict resolution module and sends its first proposal.

The negotiation protocol is the rules set by which agents precede their negotiations. In the MAAS, the negotiation protocol is in the form of a state transition diagram that shows the various legal states that an agent may be in during a negotiation, and thus the legal transitions between states that an agent is allowed to undertake. Details of these states are described in the next section. Figure 4

illustrates the logic flow of the negotiation carried on between two agents: an initiator, and a counterpart. The negotiation process starts when the initiator generates a proposal. The proposal includes new flight plans for both aircraft, including a flight plan that satisfies all of the initiator's constraints. At this stage, the initiator may not have complete knowledge of the constraints of the counterpart, so it attempts to satisfy the constraints it does know. To increase acceptance of the proposal by the counterpart, the initiator searches for a solution with maximum mutual gain with respect to its own overall cost function, and that of the counterpart.

Generally, the cost of a solution may be divided into three categories. The first is a general cost, such as arrival time, trajectory deviation from the original flight path, etc. It may be computed based on airspeed, heading, and flight plan information contained in ADS-B messages. Therefore, the aircraft may determine the general cost of other aircraft in its neighboring airspace as long as those aircraft have ADS-B messages being transmitted among them. The second cost is related to fuel consumption, and thus configuration and aerodynamically dependent. It is also different among different aircraft types. The third cost is related to pilot preference, such as passenger comfort, and is different from pilot to pilot. The overall cost function is a linear combination of the three costs, and each has a different priority under different conditions. Since the initiator is not able to know the exact cost function of its counterpart, it is only concerned with general cost.

The sum and the difference of the two flight plans finally selected are each the minimum among all potential solutions. The proposal is now sent to the responder by the initiator, along with its constraints set, since argumentation-based negotiation requires sending not only the proposal (claim), but also the reasons (support data) for selecting the specific proposal. Since all of the agents in this paper use the constraint satisfaction problem as their reasoning mechanism, they do not need to send their warrants and qualification. The responder receives the proposal and the constraints set, adds the constraints to its own constraints set, and checks to see if the proposal satisfies all of the constraints. If it does not, the responder retains the

unchanged flight plan of the initiator, and searches for a new flight plan for itself that may increase the utility with respect to its cost function. If a better flight plan is not found, it accepts the current flight plan and sends back an acceptance message. Otherwise, it sends back a counter proposal that includes its new flight plan. If the current proposal does not satisfy all of the constraints, it searches for a solution that provides maximum mutual gain for the initiator. The responder sends back a counter proposal, along with its constraints set if a new

solution is found. Otherwise, it sends back a rejection message. Returning to the initiator, if it receives a counter-proposal, it performs the assessment in the same way as described for the responder. This process iterates until one of the two agents sends an acceptance or rejection message. If an acceptance message is sent, both of the agents contact ATC to ask for approval of the new flight plans. Otherwise, both of the agents turn to ATC for help.

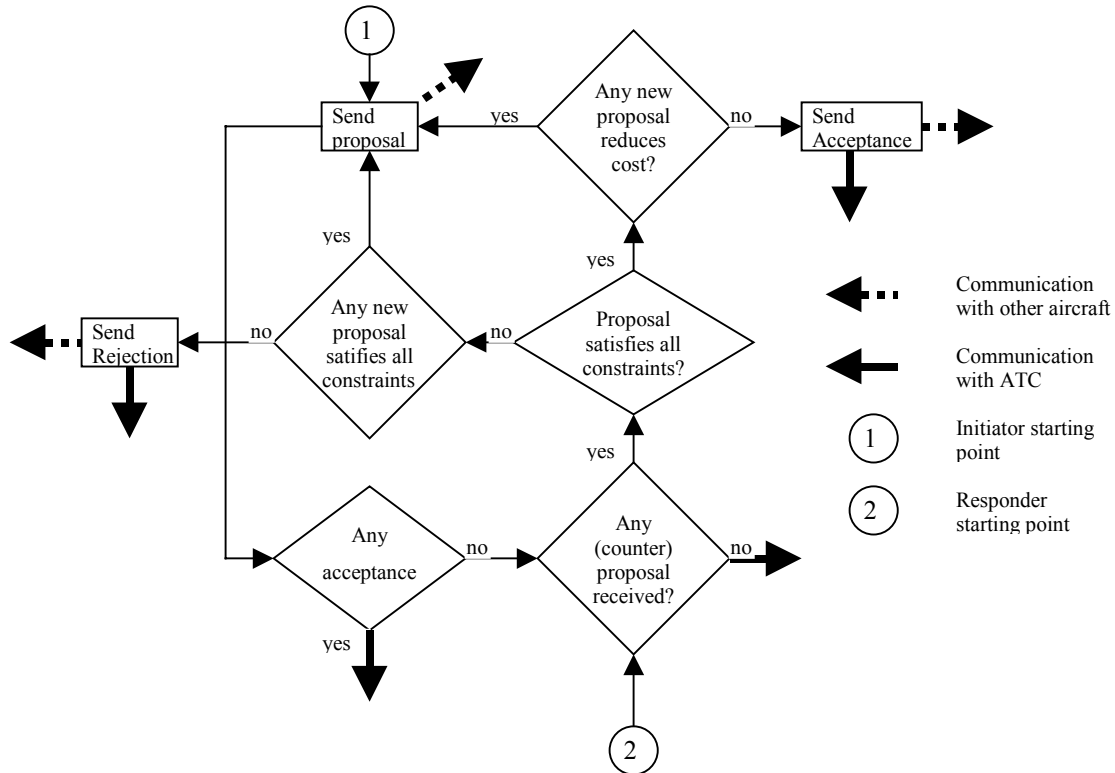


Figure 4. Logic Flow of the Negotiation Between Two Agents

Implementation Of Individual Aircraft Agent

A simple multi-agent simulation system is created for the validation and evaluation of the MAAS. Each aircraft in this multi-agent system is regarded as an independent intelligent agent. It is further divided into three main modules: an ADS-B device; a flight control system, and a Flight Management System (FMS). The overall structure of a multi-agent system involving two or more aircraft is illustrated in Figure 1. Details of each

module are described in the following sections. The environment of the MAAS is also created in the simulation system. For example, in simulating a radar image a 2-dimensional Gaussian function is chosen to give the weather intensity due to a thunderstorm at any longitude and latitude. To obtain more realistic images, the intensity computations are carried out for a series of thunderstorms [10].

ADS-B Device

Each aircraft is equipped with an ADS-B device, which requires an aircraft to periodically broadcast its state vector and other information in an information packet, as well as receive the same information packets from other aircraft. Currently, the content of an ADS-B message includes current latitude, longitude, airspeed, heading angle, altitude, rate of climb, and latitude and longitude of the previous and current waypoints in the flight plan.

Flight Control System

This module contains a basic three-axis autopilot, and navigation routines to calculate the current position of the aircraft after each execution cycle of the simulation program.

Flight Management System

Conceptually the "brain" of the aircraft, it is the primary and most important module for use with the agent system. It is further subdivided into several sub-modules as follows:

Flight Plan Management:

This module stores, edits, or erases the current flight plan. If a traffic conflict exists and two aircraft involved in the conflict agree on avoidance flight plans, this module will receive the new flight plan from the negotiation module, and modify the old one.

Tracker:

During flight, an aircraft can deviate from its designated trajectory and require correction to return to the proper course. This module calculates

the correction needed for the task and sends it to the flight control system module.

Conflict Detection Module:

This is a rule-based system that converts the inputted environment information into corresponding constraint functions, updates the constraints set of the agent, and checks to see if the current flight plan violates any of the present constraints.

Conflict Resolution Module:

Provides a candidate conflict solution if conflicts are detected. It is composed of a constraint set, a discrete and finite resolution set, a cost function set, and a search engine.

Negotiation Module:

This state-based module handles negotiations with other aircraft according to the pre-defined negotiation protocol shown in Figure 5. Table 2 lists eight legal states of the negotiation module, along with the actions the agent may perform in each state.

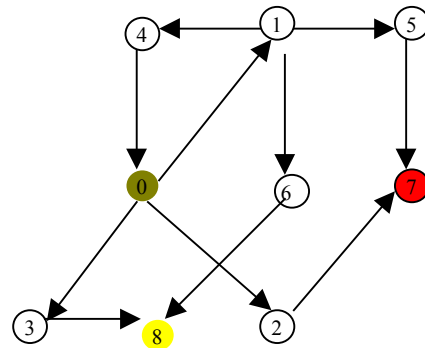


Figure 5. Agent Negotiation Legal States

Table 2. State-Based Navigation Module

State	Description	Actions
0	Waiting	Check the Inbox for any new incoming message: if the new message is a proposal, move to state 1; if the new message is an acceptance, move to state 2; if the new message is a rejection, move to state 3
1	Receiving a proposal	Evaluate incoming proposal: if the proposal is ok, move to state 5; if the proposal is not ok and the agent may create a counter-proposal, move to state 4; if the proposal is not ok and the agent may not create a counter-proposal, move to state 6
2	Receiving an acceptance	Move to state 7
3	Receiving a rejection	Move to state 8

4	Ready to send a proposal	Create and send a new proposal, move to state 0
5	Ready to send an acceptance	Send an acceptance, move to state 7
6	Ready to send a rejection	Send a rejection, move to state 8
7	Reaching a mutual agreement	Send the new flight plan to ATC for approval, terminate the current negotiation and remove it from the list
8	Unable to find a resolution to the conflict	Turn to ATC, terminate the current negotiation and remove it from the list

Numerical Example

The following example demonstrates how the multi-agent system detects and resolves a head-on traffic conflict between two aircraft.

Test Case: Head-On Conflict Detection and Resolution

Figure 6a shows two aircraft, labeled sim A and sim B, initially converging along opposite headings (6a). Aircraft sim B has a conflict detection capability, but not a resolution capability. Aircraft sim A has both capabilities, and both aircraft are capable of negotiation. Sim B detects the conflict first, and initiates the greeting process with sim A. The greeting process is very important

as it decides whether or not to start the primary negotiation. The greeting process may be extended to include more information such as an emergency mode of the aircraft, and data and information from other devices such as a weather radar, FMS, etc. Thus it can facilitate the negotiation process and make it more efficient. In addition, sim B informs sim A that it does not have a resolution capability, so sim A assumes responsibility for resolving the conflict. Sim A initiates the negotiation process by creating an avoidance flight plan, and proposes it to sim B. Sim B evaluates the proposal, accepts it, and notifies sim A. Sim A agrees, and then both aircraft execute the mutually acceptable flight plan to avoid the conflict (Figures 6b, 6c, 6d). Table 3 contains the full and detailed sequence of the negotiation process.

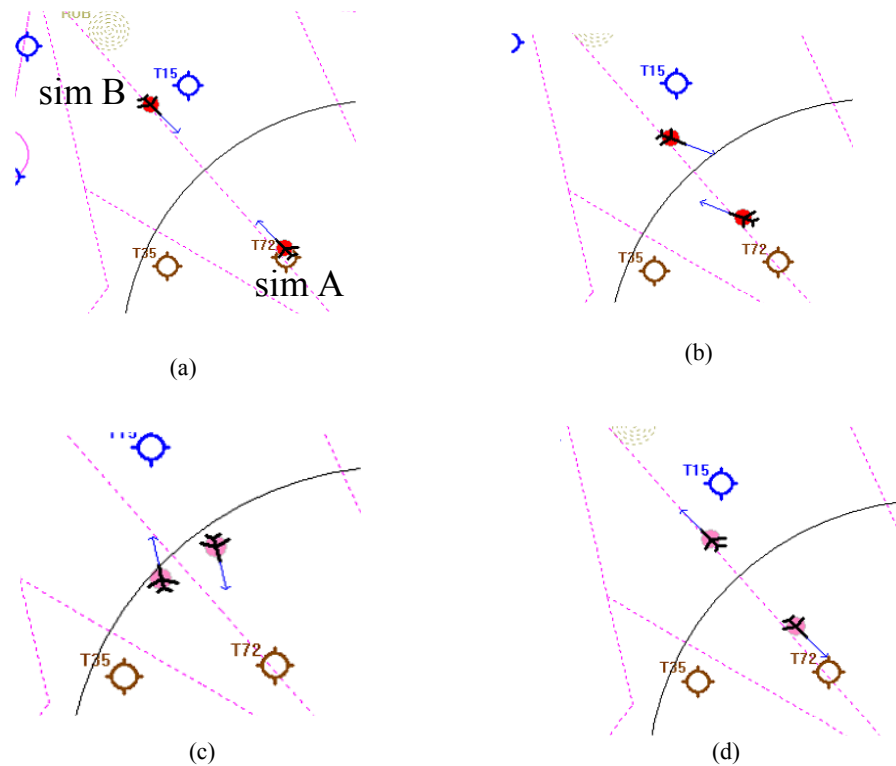


Figure 6. (a) Before Conflict (b) Starting Avoidance (c) Conflict Resolved (d) Back to Original Path

Table 3. Head-On Conflict Negotiation Process

sim_B: I suggested a conflict exists between sim_A and me
sim_A: sim_B suggested a conflict exists
sim_A: sim_B has no conflict resolution module
sim_A: I do have conflict resolution module, I may propose first
sim_A: I suggested a conflict exists between sim_B and me
sim_B: sim_A suggested a conflict exists
sim_B: sim_A will propose first
sim_B: I confirmed the conflict with sim_A, and sim_A will go on proposing a avoidance plan
sim_A: sim_B confirmed the conflict, and I will go on proposing a avoidance plan
sim_A: I propose a conflict avoidance plan to sim_B
sim_B: sim_A proposed an conflict avoidance plan
sim_B: I accept sim_A's proposed plan
sim_A: sim_B accepted my proposed plan
sim_A: I confirmed sim_B's acception of the plan
sim_B: sim_A confirmed my acception of the plan

The negotiation process is different if neither aircraft has a conflict resolution capability. In this case, the negotiation is terminated immediately to avoid wastage of communication or computing resources, and ATC is contacted to provide the resolution (Table 4). Other types of conflicts were tested to evaluate the MAAS, including pure crossing, general crossing, and follow-on cases. In each case, aircraft in the MAAS demonstrated the capacity to detect conflicts, interact and exchange

information with other aircraft, collaborate with a counterpart in searching for mutually acceptable conflict resolutions, and execute the avoidance flight plan. This limited evaluation concerned only two aircraft, and did not consider other constraints such as severe weather conditions and SUA. Future work will focus on more realistic and complex test cases that evaluate the advanced features of the MAAS.

Table 4. Greeting Process Between Aircraft Without Conflict Resolution Capability

sim_B: I suggested a conflict exists between sim_A and me
sim_A: sim_B suggested a conflict exists
sim_A: sim_B has no conflict resolution module
sim_A: I do not have conflict resolution module either,turn to ATC
sim_A: I confirmed the conflict with sim_B, but I am not able to propose a plan either
sim_B: sim_A confirmed the conflict, but it is not able to propose a plan either, turn to ATC

Conclusions

This paper proposed a multi aircraft agent system for the free flight environment. The conflict detection and resolution problem for each agent is solved as a constraint satisfaction problem. A pairwise, argument-based negotiation approach is established as a coordination method for multilaterally acceptable conflict resolutions under free flight. A simple multi-aircraft simulation

system is developed to demonstrate that appropriate conflict resolution may be produced via the proposed inter-agent negotiation method, which appears to be a promising candidate for the conflict detection and resolution algorithm for a multi-aircraft system.

This paper presented only conceptual and preliminary efforts of the authors' current research. Future work will focus on expanding the current algorithm to more advanced air traffic scenarios,

including the resolution of multiple aircraft conflicts. In addition, the negotiation protocol will address a more complicated mixed fleet environment. For example, different aircraft types with different priorities, such as commuters, business jets, and commercial air transports. Finally, the inter-negotiation model proposed in this paper can be applied to the management of other types of vehicle systems, such as autonomous ground vehicles, UAVs, submarines, etc.

Acknowledgment

This research is funded by the State of Texas Advanced Technology Program under grant number 000512-0301-1999. The authors gratefully acknowledge this support.

References

- [1] Federal Aviation Administration, "FREE FLIGHT-Introduction"
http://www.faa.gov/freeflight/ff_ov.htm
- [2] Shandy, Surya U., and John Valasek, "Intelligent Agent for Aircraft Collision Avoidance," AIAA-2001-4055, AIAA Guidance, Navigation and Control Conference, Montreal, Canada, 6-9 August 2001.
- [3] Tomlin, Claire, G. Pappas, and S. Sastry, "Conflict Resolution For Air Traffic Management: A Study In Multi-Agent Hybrid Systems," IEEE Transactions On Automatic Control, Vol. 43, No. 4, pp. 509-521, 1998.
- [4] Vilaplana, Miguel A., Colin Goodchild, and Stefano Elefante, "Co-operative Optimal Conflict Avoidance in Free Flight Airspace," 3rd USA/Europe Air Traffic Management R&D Seminar, Paper 23, 2000.
- [5] Wangermann, J. P., Robert F. Stengel, "Principled Negotiation Between Intelligent Agents: A Model For Air Traffic Management", Artificial Intelligence in Engineering, December, 1997, pp. 177-187.
- [6] Rong, Jie, Sangeeta Bokadia, Surya U. Shandy, and John Valasek, "Hierarchical Agent Based System for General Aviation CD&R Under Free Flight," AIAA-2002-4553, AIAA Guidance, Navigation and Control Conference, Monterey, CA, 5-8 August 2002.
- [7] Harper, K., S. Mulfund, S. Guarino, A. Mehta, G. Zacharias, "Air Traffic Controller Agent Model For Free Flight", AIAA-99-3987.
- [8] Yokoo, M., E.H. Durfee, T. Ishida, and K. Kuwabara, "The Distributed Constraint Satisfaction Problem: Formalization and Algorithms," IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 5, 1998, pp. 673-685.
- [9] Parsons, Simon, Carles Sierra, and Nick Jennings, "Agents That Reason and Negotiate by Arguing", Journal of Logic and Computation, Vol.8, No.3, 1998, pp. 261-292.
- [10] Bokadia, Sangeeta, and John Valasek, "Severe Weather Avoidance Using Informed Heuristic Search," AIAA-2001-4232, AIAA Guidance, Navigation and Control Conference, Montreal, Canada, 6-9 August 2001.