Free Flight in Commercial Aircraft: A New Algorithm for Automatic Conflict Avoidance

Thesis

Presented to the Department of Electronics and Computer Engineering of Technical University of Crete, GREECE

in partial fulfilment of the requirements for the degree of Master in Electronic Engineering

Constantinos Costoulakis

September 2003

Contents

List of Figures

Chapter 1

Introduction to Air Traffic Management

1.1 What is Air Traffic Management?

1.1.1 A Brief History

In the earliest days of aviation, so few aircraft were in the skies that there was little need for ground-based control of aircraft. In Europe, though, aircraft often travelled in different countries, and it soon became apparent that some kind of standard rules were needed. In 1919, the International Commission for Air Navigation (ICAN) was created to develop "General Rules for Air Traffic". Its rules and procedures were applied in most countries where aircraft operated [1].

The United States did not sign the ICAN Convention, but later developed its own set of air traffic rules after passage of the Air Commerce Act of 1926 (see Appendix B.1). This legislation authorized the Department of Commerce to establish air traffic rules for the navigation, protection, and identification of aircraft, including rules as to safe altitudes of flight and rules for the prevention of collisions between vessels and aircraft. The first rules were brief and basic. For example, pilots were told not to begin their takeoff until "there is no risk of collision with landing aircraft and until preceding aircraft are clear of the field."

As traffic increased, some airport operators realized that such general rules were not enough to prevent collisions. They began to provide a form of air traffic control (ATC) based on visual signals. The early controllers stood on the field, waving flags to communicate with pilots. Archie League was one of the system's first flagmen, beginning in the late 1920s at the airfield in St. Louis, Missouri.

As more aircraft were fitted for radio communication, radio-equipped airport traffic control towers began to replace the flagmen. In 1930, the first radio-equipped control tower in the United States began operating at the Cleveland

Municipal Airport. By 1932, almost all airline aircraft were being equipped for radio-telephone communication, and about 20 radio control towers were operating by 1935.

Further increases in flights created a need for ATC that was not just confined to airport areas but also extended out along the airways. In 1935, the principal airlines using the Chicago, Cleveland, and Newark airports agreed to coordinate the handling of airline traffic between those cities. In December, the first Airway Traffic Control Center opened at Newark, New Jersey. Additional centers at Chicago and Cleveland followed in 1936.

The early en route controllers tracked the position of planes using maps and blackboards and little boat-shaped weights that came to be called "shrimp boats." They had no direct radio link with aircraft but used telephones to stay in touch with airline dispatchers, airway radio operators, and airport traffic controllers. These individuals fed information to the en route controllers and also relayed their instructions to pilots.

In July 1936, en route ATC became a federal responsibility, and the first appropriation of 175.000 dollars was made. The Federal Government provided "airway" (see Appendix B.2) traffic control service, but local government authorities where the towers were located continued to operate those facilities.

In August 1941, Congress appropriated funds for the Civil Aeronautics Administration (CAA) to construct and operate ATC towers, and soon the CAA began taking over operations at the first of these towers, with their number growing to 115 by 1944. In the postwar era, ATC at most airports was eventually to become a permanent federal responsibility. In response to wartime needs, the CAA also greatly expanded its en route air traffic control system. Women, too, for the first time were trained as controllers during the war, and, at their peak, represented well over 40 percent of the controller workforce.

The postwar years saw the beginning of a revolutionary development in ATC, the introduction of radar (see Appendix B.2), a system that uses radio waves to detect distant objects. Originally developed by the British for military defense, this new technology allowed controllers to "see" the position of aircraft tracked on video displays. In 1946, the CAA unveiled an experimental radar-equipped tower for control of civil flights. By 1952, the agency had begun its first routine use of radar for approach and departure control. Four years later, it placed a large order for long-range radars for use in en route ATC.

Beginning in 1950, the CAA began consolidating some airport traffic control towers at smaller airports with airway communication stations, the forerunners of today's flight service stations. By 1958, it ran 84 of these combined station-towers, the last of which closed in 1981.

In 1960, the FAA [2] began successful testing of a system under which flights in certain "positive control" areas were required to carry a radar beacon, called a transponder, that identified the aircraft and helped to improve radar performance. Pilots in this airspace were also required to fly on instruments regardless of the weather and to remain in contact with controllers. Under these conditions, controllers were able to reduce the separation between aircraft by as much as half the standard distance.

For many years, pilots had negotiated a complicated maze of airways. In September 1964, the FAA instituted two layers of airways, one from 1000 to 18.000 feet (305 to 5.486 meters) above ground and the second from 18.000 to 45.000 feet (13.716 meters). It also standardized aircraft instrument settings and navigation checkpoints to reduce the controllers' workload.

Although experimental use of computers in ATC had begun as early as 1956, a determined drive to apply this technology began in the 1960s. To modernize the National Airspace System (see Appendix B.2), the FAA developed complex computer systems that would replace the plastic markers for tracking aircraft. Instead, controllers viewed information sent by aircraft transponders to form alphanumeric symbols on a simulated three-dimensional radar screen. By automating some routine tasks, the system allowed controllers to focus on providing separation. These capabilities were introduced into the ATC system during the ten years that began in 1965.

The FAA established a Central Flow Control Facility in April 1970, to prevent clusters of congestion from disrupting the nationwide air traffic flow. This type of ATC became increasingly sophisticated and important, and in 1994, the FAA opened a new Air Traffic Control System Command Center with advanced equipment.

In January 1982, the FAA unveiled the National Airspace System (NAS) Plan. The plan called for modernized flight service stations, more advanced systems for ATC, and improvements in ground-to-air surveillance and communication. Better computers and software were developed, air route traffic control centers were consolidated, and the number of flight service stations reduced. New Doppler radars and better transponders complemented automatic, radio broadcasts of surface and flight conditions.

The FAA recognized the need for further modernization of air traffic control, and in July 1988, selected IBM to develop the new multi-billion-dollar Advanced Automation System (AAS) for the Nation's en route ATC centers. AAS would include controller workstations, called "sector suites," that would incorporate new display, communications and processing capabilities. The system would also include new computer hardware and software to bring the air traffic control system to higher levels of automation.

In December 1993, the FAA reviewed its order for the planned AAS. IBM was far behind schedule and had major cost overruns. In 1994 the FAA simplified its needs and picked new contractors. The revised modernization program continued under various project names. Some elements met further delays. In 1999, controllers began their first use of an early version of the Standard Terminal Automation Replacement System, which included new displays and capabilities for approach control facilities. During the following year, FAA completed deployment of the Display System Replacement, providing more efficient workstations for en route controllers.

In 1994, the concept of Free Flight was introduced. It might eventually allow pilots to use onboard instruments and electronics to maintain a safe distance between planes and to reduce their reliance on ground controllers. Full implementation of this concept would involve technology that made use of the Global Positioning System (see Appendix B.2) to help track the position of aircraft. In 1998, the FAA and industry began applying some of the early capabilities developed by the Free Flight program.

Current studies to upgrade ATC include the Communication, Navigation and Surveillance for Air Traffic Management system that relies on the most advanced aircraft transponder, a global navigation satellite system, and ultra-precise radar. Tests are underway to design new cockpit displays that will allow pilots to better control their aircraft by combining as many as 32 types of information about traffic, weather, and hazards.

1.1.2 Definition, Objectives and Structure

Air traffic control involves monitoring the movements of all aircraft, both in the air and on the ground, in the vicinity of an airport. Its main purpose is to keep aircraft safely separated to prevent accidents. Air traffic control is needed so that the risk of collision becomes extremely low. This can be achieved only by strictly following procedures that are set out and monitored by air traffic controllers, individuals who direct air traffic within assigned airspace and control moving aircraft and service vehicles at airports [3].

In flight, an aircraft follows en route air traffic control instructions as it flies through successive flight information regions. When it approaches an airport for landing, the aircraft enters the terminal control area where it is monitored by controllers using radar and who constantly tell pilots how to navigate within the area. Controllers also monitor the aircraft all the way to the ground and tell the pilot how to maneuver on the ground to avoid collisions on the ground of the airfield and how to reach its final location where passengers can disembark. Departing aircraft go through a reverse procedure. Overall, the degree of control depends greatly on the weather conditions. In general, the better the weather, then the less the control.

The International Civil Aviation Organization (ICAO), defines the objectives of air traffic control as:

• Preventing collisions between aircraft in flight

• Preventing collisions between aircraft on the maneuvering area of an airport and obstructions on that area

• Expediting and maintaining an orderly flow of air traffic

• Providing advice and information useful for the safe and efficient conduct of flights

• Notifying appropriate organizations regarding aircraft in need of search and rescue aid, and assisting such organizations as required

The main goal of ATC is to guarantee security and to give aircraft optimal trajectories to fly from one airport to an other.

There is a simple definition of security in ATC systems: two aircraft can never be closer than one standard separation. A standard separation is a distance usually given in nautical miles. It depends on the equipment available to control aircraft. It is usually 8 or 5 Nautical Miles (NM) in the horizontal plane and 1000 or 2000 feet in the vertical plane. Two aircraft are in conflict when both standard separation are violated.

It is useful to try to understand how an Air Traffic Control and an Air Traffic Management system are built. They can be represented by an assembly of filters, or shells. A classical view of the shells in an ATC system could be:

• Airspace design (airways, control sectors, etc): when joining two airports, an aircraft must follow routes and beacons; these beacons are necessary for pilots to know their position during navigation and help controllers to visualize the traffic. As there are many aircraft simultaneously present in the sky, a single controller is not able to manage all of them. So, airspace is partitioned into different sectors, each of them being assigned to a controller. This task aims at designing the air network and the associated sectoring.

• Air Traffic Flow Management (ATFM) (strategic planning, a few hours ahead): with the increasing traffic, many pilots choose the same routes, generating many conflicts on the beacons inducing overloaded sectors. Traffic assignment aims at changing aircraft routes to reduce sector congestion, conflicts and coordinations. It also aims at computing arrival times for aircraft at airports. Airport capacity is often the bottleneck of the system, especially in the USA, and an efficient sequencing is one key of Airspace capacity.

• Coordination planning (a few minutes ahead): this task guarantees that new aircraft entering sectors do not overload the sector.

• Classical control in ATC centers (up to 20 mn ahead) at this level, controllers solve conflicts between aircraft.

• Collision avoidance systems (a few minutes ahead): this level is activated only when the previous one has failed. This level is supposed to be activated only in emergency situations.

Each level has to limit and organize the traffic it passes to the next level, so that this one will never be overloaded.

However, it is still difficult to have aircraft separated. The reasons for that are different regarding the country. For instance, in the USA, airport capacity is the main problem, while in Europe, and mainly in France, En Route capacity is the critical point.

As Air Traffic keeps increasing, the Air Traffic Control overload is now a serious concern. For the last twenty years, different approaches have been tried, and different solutions have been proposed. To be short, all theses solutions fall in the range delimited by the two following extreme positions:

• On the one hand, it could be possible to imagine an ATC system where every trajectory would be planned and where each aircraft would follow its trajectory with a perfect accuracy. With such a system, no reactive system would be needed, as no conflict between aircraft would ever occur. This solution is close to the ARC-2000 hypothesis, which has been investigated by the Eurocontrol Experimental Center.

• On the other hand, it could also be possible to imagine an ATC system where no trajectories are planned. Each aircraft would fly its own way, and all collisions would have to be avoided by reactive systems. Each aircraft would be in charge of its own security. This could be called a completely free flight system. The free flight hypothesis is currently seriously considered for all aircraft flying "high enough" in a quite near future.

Of course, no ATC system will ever totally rely on only one of these two hypothesis. It is quite easy to understand why. A completely planned ATC is impossible, as no one can guarantee that each and every trajectory would be perfectly followed; there are too many parameters that can not be perfectly controlled: meteorological conditions (storms, winds, etc.), but also breakdowns in aircraft (motor, flaps, etc) or other problems (closing of landing runaway on airports, etc). On the other hand, a completely reactive system looks difficult to handle; it would only perform local optimizations for trajectories. Moreover, in the vicinity of departing and landing areas, the density of aircraft is so high that trajectories generated by this system could soon look like Brownian movements.

1.2 Automation Issues in Air Traffic Management

The term automation has been defined in a number of ways in the technical literature. It is defined by some as any introduction of computer technology where it did not exist before. Other definitions restrict the term to computer systems that possess some degree of autonomy. Here, we will define automation as: "a device or system that accomplishes (partially or fully) a function that was previously carried out (partially or fully) by a human operator" and we will retain that definition throughout this work.

The pressures for automation of the air traffic control system originate from three primary sources: the needs for improved safety, and efficiency (which may include flexibility, potential cost savings, and reductions in staffing); the availability of the technology; and the desire to support the controller. Even given the current very low accident rate in commercial and private aviation, the need remains to strive for even greater safety levels: this is a clearly articulated implication of the "zero accident" philosophy of the Federal Aviation Administration (FAA) and of current research programs of the National Aeronautics and Space Administration (NASA). Naturally, solutions for improved air traffic safety do not need to be found only in automation; changing procedures, improving training and selection of staff, and introducing technological modernization programs that do not involve automation per se, may be alternative ways of approaching the goal. Yet increased automation is one viable approach in the array of possibilities. The need for improvement is perhaps more strongly driven by the desire to improve efficiency without sacrificing current levels of safety. Efficiency pressures are particularly strong from the commercial air carriers, which operate with very thin profit margins, and for which relatively short delays can translate into very large financial losses. For them it is desirable to substantially increase the existing capacity of the airspace (including its runways) and to minimize disruptions that can be caused by poor weather, inadequate air traffic control equipment, and inefficient air routes. The forecast for the increasing traffic demands over the next several decades exacerbates these pressures. Of course, as with safety, so with efficiency: advanced air traffic control automation is not the only solution. In particular, the concept of free flight is a solution that allocates greater responsibility for flight path choice and traffic separation to pilots (i.e., between human elements), rather than necessarily allocating more responsibility to automation. Automation is viewed as a viable alternative solution to solve the demands for increased efficiency. Furthermore, it should be noted that free flight does depend to some extent on advanced automation and also that, from the controller's point of view, the perceived loss of authority whether it is lost to pilots (via free flight) or to automation, may have equivalent human factors implications for design of the controller's workstation.

It is, of course, the case that automation is made possible by the existence of technology. It is also true that, in some domains, automation is driven by the availability of technology; the thinking is, "the automated tools are developed, so they should be used." Developments in sensor technology and artificial intelligence have enabled computers to become better sensors and pattern recognizers, as well as better decision makers, optimizers, and problem solvers. The extent to which computer skills reach or exceed human capabilities in these endeavors is subject to debate and is certainly quite dependent on context. However, more and more experts agree that the availability of computer technology should be a reason for automation in and of itself. It should be considered only if such technology has the capability of supporting legitimate system or human operator needs. Automation has the capability both to compensate for human vulnerabilities and to better support and exploit human strengths.

Current system needs and the availability of some technology provide adequate justification to continue the development and implementation of some forms of air traffic control automation. However, it is strongly argued that this continuation should be driven by the philosophy of *human-centered automation*, whose main characteristics are summarized as follows:

The choice of what to automate should be guided by the need to compensate for human vulnerabilities and to exploit human strengths. The development of the automated tools should proceed with the active involvement of both users and trained human factors practitioners. The evaluation of such tools should be carried out with human-in-the-loop simulation and careful experimental design. The introduction of these tools into the workplace should proceed gradually, with adequate attention given to user training, to facility differences, and to user requirements. The operational experience from initial introduction should be very carefully monitored, with mechanisms in place to respond rapidly to the lessons learned from the experiences.

Chapter 2

The Concept of Free Flight

2.1 Free Flight in General

2.1.1 Definition

Today's ATC concept as described in the previous chapter is often not the most optimal way of flying from an airline point of view. Many companies have become frustrated by what they view as inefficiencies in the national airspace. Such inefficiencies are viewed to result, in part, from three factors:

(1) Standard linear airways that rarely allow the most direct flight between two points (e.g., a great circle route),

(2) Strict adherence to air traffic control procedures for route changes, which sometimes imposes delays, inefficiencies, or denial of requests that in fact might be entirely safe, and

(3) Dependence on radar for separation standards, which are therefore constrained by the resolution of radar in estimating position.

This situation translates into flight delays, occasionally missed connections, passenger complaints, excess fuel consumption, excess crew time, and, ultimately, loss of revenue, for companies that already have a very thin profit margin. Airlines would prefer a more optimal way of flying with respect to fuel and time within the safety margins if possible. Assuming the aircrew is able to perform the conflict resolution task, they might be able to fly more optimal routes. Self-optimization therefore could provide a more efficient, while still safe, and apparently more complex traffic pattern. This idea of self-optimization forms the basis of Free Flight.

In response to the above concerns, since 1994 an effort triggered by the airline industry has begun to examine the concept of user-preferred routing or free flight, a concept in which pilots are better able to select their preferred routes, unconstrained by air traffic control. This system is designed to allow pilots to take better advantage of local information (e.g., weather) that may not be available to air traffic control and, most important, will allow pilots to rely on the global positioning system (GPS) for navigation and separation that is far more precise than the radar-based guidance available from air traffic control. The concept has been developed by a working committee on free flight sponsored by the RTCA [4], who propose the following definition:

"A safe and efficient flight operating capability under instrument flight rules (IFR) in which the operators have the freedom to select their path and speed in real time. Air traffic restrictions are only imposed to ensure separation, to preclude exceeding airport capacity, to prevent unauthorized flight through special use airspace, and to ensure safety of flight. Restrictions are limited in extent and duration to correct the identified problem. Any activity which removes restrictions represents a move toward free flight."

Summing up, free flight is a new way of managing air traffic that was originally designed to enhance the safety, capacity, and efficiency of the U.S. NAS (National Aerospace System). Under this new management system, air traffic control is expected to move gradually from a highly structured system based on elaborate rules and procedures to a more flexible system that allows pilots, within limits, to change their route, speed, and altitude, notifying the air traffic controller of the new route. In contrast, under the present system, while flight plans are developed in conjunction with air traffic control personnel, aircraft are required to fly along specific routes with minimal deviation. When deviations from designated routes are allowed (to, for example, avoid severe weather) they must be pre-approved by an air traffic controller. Under free flight, despite the availability of flexibilities to pilots, the ultimate decision-making authority for air traffic operations will continue to reside with controllers.

While FAA and the aviation community have recently increased their efforts to implement free flight, the concept of free flight (allowing pilots to fly more optimal routes) is not new. In fact, the idea has been around for decades. With the development of navigation technology in the 1970s that allowed aircraft to fly directly from origin to destination without following fixed air routes (highways in the sky), the possibility of providing pilots with flexibility in choosing routes became viable. However, until recently, movement to develop the procedures and decision support systems needed to fully use this type of point-to-point navigation has been slow. In the last several years, because of the need to meet demands for increasing the systems capacity and efficiency, FAA and aviation system users and their major trade organizations, representatives of air traffic control personnel, equipment manufacturers, the U.S. Department of Defense (DOD), and others (collectively referred to as stakeholders) have been working on plans to accelerate the implementation of free flight.

2.1.2 Reasons for Free Flight

There are three drivers for free flight. Two are economic and the third is related jointly to comfort and safety:

1. Horizontal free flight results in fuel saving by allowing the flying of shorter, more direct routes, ideally following great circle paths, avoiding head winds or capitalizing on tailwinds.
2. Vertical free flight results in fuel saving by allowing flying at altitudes that have the most favorable winds.
3. Flying around bad weather and clear air turbulence (both horizontally and vertically) results in passenger comfort and safety.

It is important to realize that the concept of free flight is not defined by a universally accepted set of procedures. Different players have very different notions of what it should be, how free it will be, and over what domains of the airspace it will apply (e.g., en route versus TRACON, high altitude versus all altitudes, continental versus oceanic). However, an important distinction contrasts *strategic* free flight, in which route planning is done in a manner that is unconstrained by air traffic control (i.e., free scheduling and free routing), with *tactical* free flight, in which executions of flight path changes, including maneuvers to avoid conflicts, are carried out without air traffic control guidance or instructions. A continuum of levels exists between strategic and tactical maneuvering.

A large number of issues must be addressed and resolved before determining if a free flight system is feasible in an airspace whose regulators and occupants are committed to safety as a primary goal. We discuss these issues below in two categories, those pertaining to the airspace system as a whole, and those focusing more directly on human factors.

2.1.3 System-level Issues

Air Traffic Control Role

The role of air traffic control in a free flight regime will continue to remain a critical and controversial issue. Indeed, one of the thornier issues concerns the appropriate level of authority that should be maintained by air traffic control. On one extreme is a system in which aircraft maneuver as they choose, allowing air traffic control to be only a passive monitor of the changing trajectories, until or unless these lead to danger, and then intervening with control. A more conservative system will require pilots to inform air traffic control of their maneuvers but proceed to carry them out unless vetoed by air traffic control. Still more conservative is a system not unlike that in existence today, in which pilots request deviations and air traffic control approves. However, under a free flight regime, such requests would be far more frequent (as would approvals), given that pilots would have the equipment (GPS, cockpit display of traffic information) and training to carry them out safely.

Pilot's and the Airline Operations Center's Roles

Up to this point, it has been implicitly assumed that the pilot is the one calling the shots in a free flight regime. However, from the standpoint of commercial aviation, the pilot is not necessarily the best originator of unconstrained maneuver plans. Instead, the airline operations center, and its agent the aircraft dispatcher, will probably have far better global knowledge of weather patterns, winds, traffic scheduling, and regional traffic density, in order to make more nearly optimal decisions on route and trajectory changes. Hence, although the pilot may become free from air traffic control constraints, these may be replaced by constraints from the dispatcher.

System-Wide Efficiency

On paper, convincing cases can be made for the cost savings of direct routings and other free flight concepts. However, in practice, savings that appear in one place may be lost in others. For example, complex simulation runs have revealed that free flight can considerably lessen the cruise flight time en route between TRACONs (see Appendix B.2). But much of the time saved may then be lost, as a large stack of rapidly arriving aircraft must now wait at the feeder gate to a TRACON (constrained airspace), in order to be handled in a less efficient, more sequential fashion by air traffic control. The analysis revealed a phenomenon whereby several aircraft, all requesting the same preferred routing, created a bunching on that preferred route that ultimately slowed their flight, and in some cases required redirection back to the earlier non-preferred route, now with a considerable loss of time. In this case, flight time is not saved, nor is any workload reduced for the controller. It may well be difficult or impossible to predict other such system-wide effects until or unless a full operational test of the system is in place.

Safety Versus Efficiency

The pressure toward free flight is primarily efficiency driven. Lee et al. (1997) simulated flying on a set of cross-country routes and estimated a 6 percent fuel savings and, with equal fuel burn between preferred and non-preferred routing, found an average 15-minute time savings. The FAA has rightfully maintained a conservative stance, driven by safety, in responding to pressures to move toward free flight. But given the recent commitment to reduce accident rates by a factor of five over the next decade (White House Commission on Aviation Safety and Security, 1997), it can be argued that any radical change to an already safe system will at least have the possibility of being safety-compromising. And given the complexity of the free flight concept, accurate assessment of its safety benefits may not be achievable for several years after its implementation.

2.2 Controlled Flight vs. Free Flight

As we have seen, up to now, the separation task has been carried out by an air traffic controller at an Area Control Center (controlled flight) whereas now an effort is being made to move this task to the cockpit. This move is really more radical than it sounds. The task is not simply moved, it is rather distributed among the aircrews in the sector, which means a fundamental change is going to happen in the structure of the system. From a centrally controlled system, the ATC system becomes a distributed system with inter-acting elements [5].

There are some general aspects to decentralization of a system. For the centrally organized ATC system, the addition of another aircraft puts more strain on the central node and its capacity, since the controller is the only active factor and aircraft are passive elements increasing the dimension of the problem, often in a dramatic way. In a distributed system the addition of another aircraft adds an extra potential problem solver to the sector, since the aircraft are no longer passive elements.

Another effect is the data flow: in a centrally controlled system all data of all aircraft has to be available tot he central node for a good global optimization. In the case of self-optimization these data are already locally available allowing more specific (but local nonetheless) optimization. On the other hand, instability can theoretically lead to catastrophic situations in a distributed system, something that could have been prevented in an orderly, centrally controlled system.

We saw the differences between the two approaches in a conceptual level and now we will proceed to describe them in more detail by looking at how the various components of the system interact in both situations.

2.2.1 Controlled Flight

In Controlled Flight, the pilot does not fully control the flown trajectory. He merely follows the route for which he has received a clearance. The controller therefore knows this route and decides the speed and altitude the aircraft will follow while flying. The route has been entered into the Flight Management System (FMS) by the pilot(s). The FMS is connected to the autopilot and autothrottle. When the autopilot's Lateral NAVigation (LNAV) mode is enabled, the heading is controlled by the FMS ensuring the aircraft will fly over the waypoints that form the FMS route. Similarly, when the Vertical NAVigation (VNAV) mode has been selected, the altitude, speed and vertical speed will be controlled by the FMS. Most of the time, especially during the cruise phase, LNAV and VNAV are enabled.

In manual flight the autothrottle and autopilot are disconnected and the pilot also closes the flight control loop by controlling the speed vector himself. For instance, the take-off is always performed manually and often the approach and landing are also flown manually.

From the ground the radar monitors the aircraft's position during the complete flight as long as the aircraft is under radar coverage. Remote and less developed areas as well as the ocean lack en-route radar coverage although the aircraft remain under ground control. The radar determines the aircraft's position every rotation (usually every 4 seconds). The aircraft's lateral position is determined by the radar independent of the aircraft's navigation system ("independent surveillance"). The altitude as well as a four-digit ID code ("squawk") are transmitted by the aircraft's transponder ("dependent surveillance"). Since all aircraft use the same reference atmospheric pressure to determine the altitude, the relative altitude is determined quite accurately. This ensures two aircraft that are transmitting a sufficiently different altitude, will indeed be separated vertically (i.e. there will be no danger of collision).

These data as received by the radar are fed into a filter progam called "tracker". The tracker can receive information from several radars and combines this data into one traffic picture which is shown to the controller or his display together with labels indicating the altitude, identification and other data available on the specific aircraft.

The controller separates the aircraft by assigning different altitudes as well as speed and heading directions. In this way conflicts are prevented long before they could occur. Several tools are used by the controller to aid him in his decisions. However, if the controller fails to prevent a conflict or a pilot fails to obey his directions, there is a potential loss of separation. In that case a Short Term Conflict Alert (STCA) will alert the controller of a predicted loss of separation, approximately 5 minutes before it actually happens. If the conflict is not prevented in this way, then there is also another independent safety net called TCAS (Traffic Collision Avoidance System) onboard the aircraft. This system will alert the crew of an aircraft about 45 seconds before a predicted collision, allowing them to maneuver before it could result in an actual collision.

The directions of the controller are transmitted to the pilot by voice radiotelephony and the pilot has to read back a clearance to confirm that the directions have been received and understood. This communication process takes a certain time and this is the cause of the sequential nature of the controller's actions. This time management is an essential part of the "art of air traffic control".

Finally, a controller can only handle a limited number of aircraft at the same time. Therefore the airspace is divided in sectors whose size and shape are determined by the traffic flow. Typically, a sector will contain anything from 5 to 20 aircraft. When an aircraft leaves a sector, it is handed off to the next controller, a task that also takes some time to perform.

2.2.2 Free Flight

In Free Flight there is no longer a need to provide a controller with an orderly traffic picture. All of the other information previously routed via ATC now goes directly to the crew, including conflict alerts. The pilot does not need the overall picture as long as the separation of his own aircraft can be maintained. This could allow more optimal routing. In this case, the route in the FMS will be more direct and typically consist of longer legs between waypoints. Depending on constraints of weather and the inhibited SUA (Special Use Airspace - typically reserved for military flights) it could consist of one leg following the direct route ("great circle") between the entry point and the exit point of the Free Flight sector. The aircraft will probably still use the LNAV and VNAV mode to follow this route and will gradually climb during cruise to optimize fuel consumption, instead of the discrete step climb during controlled flight, which was required for the traffic picture of the controller. Also, there will be fewer bends in the route both horizontally and vertically.

The aircraft's position, as determined by its navigational systems, is broadcast by the ADS-B transmitter (see Appendix B.2). These position data are received by all aircraft (and ground stations) within range which can vary from 80 to 200 nm. They receive messages containing identification, position, velocity and maybe even information on the intended route ("dependent surveillance"). The update rate can vary from once per second to once per 25 seconds depending on the ADS-B system and possibly the range. These data are filtered (similar to a radar tracker) and presented to the pilot on the traffic display, which in modern aircraft will be integrated in the navigation display. The tracker will be part of a new collision avoidance system which will contain a module called Conflict Detection.

The Conflict Detection module predicts the future trajectory of all the aircraft in the sector, using the received data on position, velocity and possibly the intended route. As soon as a future loss of separation, a so-called conflict, has been detected within the lookahead time, the pilot will be alerted aurally and visually on the traffic display.

The module will calculate one or more advised maneuvers to avoid the loss of separation or "resolve" the conflict. This advisory is shown on the navigation (and traffic) display and primary flight display. In some proposed implementations the advisory consists of a route change which is transferred to the flight management system to be activated by the crew. The crew must then select one of the proposed resolutions, or create their own and resolve the conflict. If required, TCAS might still be present to provide an independent safety net. Voice radiotelephony can also be regarded as an independent safety net, allowing the crew to report a failure and their relative position in the vicinity.

Chapter 3

Proposed Approach

3.1 Framework

We consider the problem of resolving conflicts arising among many aircraft following a cooperative approach, i.e. all aircraft involved in a conflict collaborate to its resolution. This approach is based on the following central assumptions:

• Aircraft are assumed to cruise within a fixed altitude layer (the layer structure is the same as the one described in [6]). Aircraft can thus be modelled in a purely kinematic way, as points in a plane with an associated fore axis, that indicates the direction of motion, and conflict envelope radius. The task of each aircraft is to reach a given goal configuration from a given start configuration (start and goal configurations may represent waypoints planned for the aircraft by the higher level planner).

• All interacting aircraft cooperate towards optimization of a common goal, as agents in the same team. The common goal is to reach the final configuration avoiding all possible conflicts in the minimum possible time. This applies to all aircraft in the same airspace, defined as a zone in which they can exchange information on positions, velocities and goals.

• In ATC literature [6-14], two different cases have been considered up to now: in the first case aircraft maneuvers are studied that consist of instantaneous velocity changes and in the second case only heading angle changes are allowed. We propose a combinatory approach that allows aircraft to change simultaneously both their velocities and heading angles. The efficiency of this approach will be studied and compared to those used in previous approaches.

The problem of finding the shortest conflict-free paths, can be modelled as a non-linear mixed-integer programming (MIP) problem, which may be solved using optimization tools such as GAMS[15]. The resulting algorithm can be rerun at regular sample times to generate a feedback control law. This will lead to a piecewise linear overall trajectory for the aircraft.

3.2 Problem Statement

We consider a finite number *n* of aircraft sharing the same airspace; each aircraft is an autonomous vehicle that flies on a horizontal plane. Each aircraft has an initial and a final, desired configuration (position, heading angle) and the same goal which is to reach the final configuration in minimum time while avoiding conflicts with other aircraft. A conflict between two aircraft occurs if they are closer than a given distance *d* (current enroute air traffic control rules often consider this distance to be 5 nautical miles [2]).

Aircraft are identified by points in the plane (position) and angles (heading angle, direction) and thus by a point $(x,y,\theta) \in R \times R \times S^1$. Let $(x_i(t),y_i(t),\theta_i(t))$ denote the configuration of the *i*-th aircraft at time *t*; a conflict occurs when the distance between two aircraft is less than *d*, i.e. a conflict between aircraft *i* and *j* occurs if for some value of *t*,

$$\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} < d. \tag{3.1}$$

Considering the aircraft as discs of radius $d/2$, the condition of non conflict between aircraft is equivalent to the condition of non intersection of the discs. In the following we refer to those as the *safety discs* of the aircraft. Later on, we will detail the construction of non-linear conflict avoidance constraints that are equivalent to (3.1).

Our method is based mainly on the approach of [7]. In this approach, two different cases are considered in order to avoid possible conflicts:

1. Aircraft are allowed to change the velocity of flight but the direction of motion remains fixed. This case is referred to as the Velocity Change problem (VC problem).

2. Aircraft fly at the same velocity v and are only allowed to change instantaneously the direction of flight. This case is referred to as the Heading Angle Change problem (HAC problem).

In both cases each aircraft is allowed to make a maneuver, at time t=0, to avoid all possible conflicts with other aircraft and it is assumed that no conflict occurs at time t=0.

These approaches are formulated as linear MIP problems which then can readily be solved using various computational tools. However, they are not very realistic, as aircraft can only make a specific type of maneuver (velocity or heading angle). We propose a variation of the above methods, which will allow aircraft to change simultaneously both their velocities and their heading angles, thus making it easier to avoid possible conflicts with less overall deviation from their nominal paths. It has to be noted, that the resulting problem will be non-linear and thus, more computationally expensive than before, but as it turns out, the difference in algorithm execution times is not so big and moreover the accuracy of the problem modelling is increased dramatically.

We proceed to define by q_i the velocity change and by p_i the heading angle deviation of the *i*-th aircraft respectively. The problem consists of finding a velocity change q_i and a heading angle deviation p_i, for each aircraft, to avoid any possible conflict while deviating as little as possible from the original flight plan. The deviation here is viewed as the difference between the time it would take for the aircraft to reach its final configuration if there weren't any conflicts (and thus, if it didn't need to maneuver at all) and the actual time it takes it with the changes of velocity and/or heading angle. This time difference is summed up for all the aircraft and the result is the objective function which we will try to minimize. The problem is formulated as a mixed non-linear optimization problem with non-linear constraints and some boolean variables.

3.3 Problem Formulation

In this section we briefly discuss the VC and HAC problems and then proceed to formulate the combined approach.

3.3.1 The VC Problem

The VC problem consists of aircraft that fly along a given fixed direction and can maneuver only once with a velocity variation. The *i*-th aircraft changes its velocity by a quantity of q_i that can be positive (acceleration), negative (deceleration) or null (no velocity variation). Each aircraft has upper and lower bounds on the velocity υ_i: $\upsilon_{i,min} \leq \upsilon_i \leq \upsilon_{i,max}$. For commercial flights, we usually have $\frac{\upsilon_{i,max}-\upsilon_{i,min}}{\upsilon_{i,min}} \leq 0.1$. The problem then is to find an acceptable value of q_i, for each aircraft, such that all conflicts are avoided and such that the new velocity satisfies the upper and lower bounds. Hence, given the initial velocity υ_i, after a velocity variation of amount q_i the following inequalities must be satisfied:

$$\upsilon_{i,min} \leq \upsilon_i + q_i \leq \upsilon_{i,max}. \tag{3.2}$$

We will restrict to the case of two aircraft to obtain conflict avoidance constraints for simplicity reasons. The general case of n aircraft can then easily be addressed to. Consider two aircraft denoted 1 and 2, respectively and let (x_i, y_i, θ_i), i=1,2 be the aircraft position and direction of motion and υ_i be the initial velocity. Referring to Figure 3.1, we consider the following velocity vectors:

$$\hat{\upsilon}_1 = \begin{bmatrix} (\upsilon_1 + q_1)cos\theta_1 \\ (\upsilon_1 + q_1)sin\theta_1 \end{bmatrix}; \tag{3.3}$$

$$\hat{\upsilon}_2 = \begin{bmatrix} (\upsilon_2 + q_2)cos\theta_2 \\ (\upsilon_2 + q_1)sin\theta_2 \end{bmatrix}; \tag{3.4}$$

$$\hat{\upsilon}_1 - \hat{\upsilon}_2 = \begin{bmatrix} (\upsilon_1 + q_1)cos\theta_1 - (\upsilon_2 + q_2)cos\theta_2 \\ (\upsilon_1 + q_1)sin\theta_1 - (\upsilon_2 + q_1)sin\theta_2 \end{bmatrix}; \tag{3.5}$$

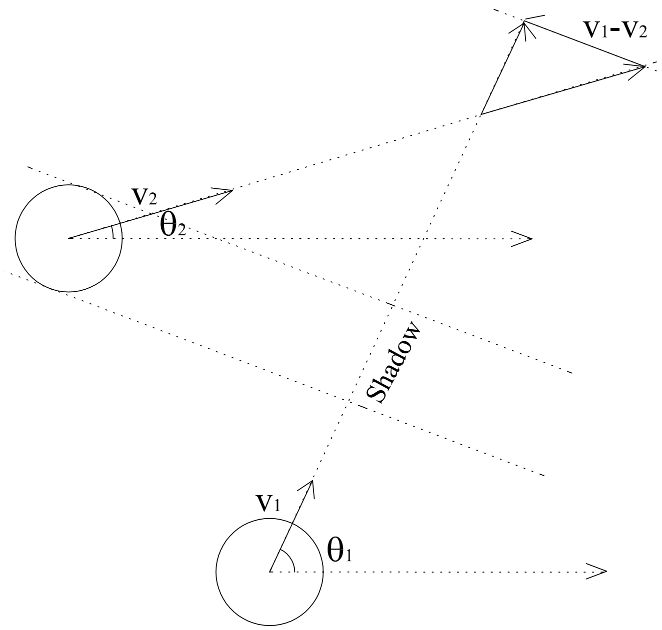

Figure 3.1: Geometric construction for conflict avoidance constraints in the case of intersecting trajectories for the VC problem. In this case Aircraft 1 do not intersect the shadow generated by Aircraft 2 then no conflict will occur between the aircraft.

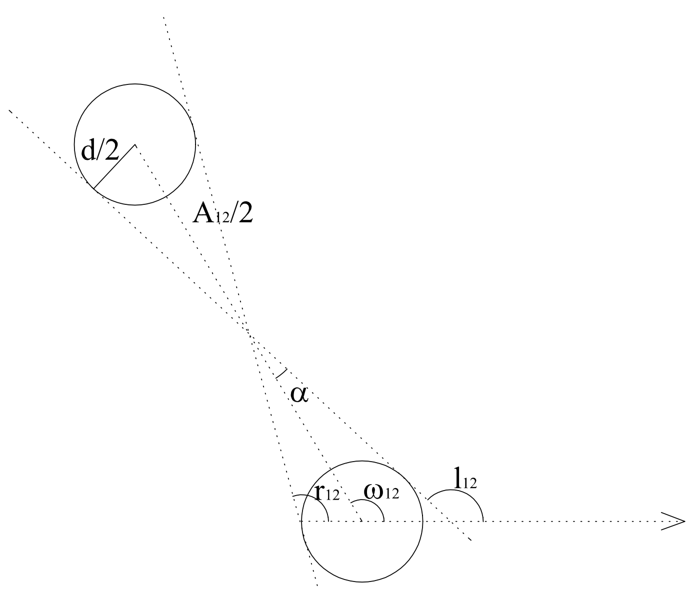

Figure 3.2: The two non parallel straight lines tangent to the safety discs of radius $d/2$ for two aircrafts at distance $A_{12}/2$.

The two lines parallel to $\hat{\upsilon}_1$-$\hat{\upsilon}_2$ that are tangent to aircraft 2, localize a segment on the direction of motion of 1 (see Figure 3.1). We will refer to this segment as the *shadow* of aircraft 2 along the direction of 1. A conflict occurs if the safety disc of aircraft 1 intersects the shadow generated by aircraft 2, or vice-versa since $\hat{\upsilon}_1$-$\hat{\upsilon}_2$ and $\hat{\upsilon}_2$-$\hat{\upsilon}_1$ are parallel.

Consider now the two non-parallel straight lines that are tangent to the discs of both aircraft (see Figure 3.2). Let l_{12}, r_{12} be the angles between these two straight lines and the horizontal axis. We have l_{12}=ω_{12}+a and r_{12}=ω_{12}-a while a=$arcsin(\frac{d}{A_{12}})$ where A_{12} is the distance between the two aircraft and ω_{12} is the angle between the line that joins the aircraft and the horizontal axis. It follows from the above that the conditions for non conflict are:

$$\frac{(\upsilon_1 + q_1)sin\theta_1 - (\upsilon_2 + q_2)sin\theta_2}{(\upsilon_1 + q_1)cos\theta_1 - (\upsilon_2 + q_2)cos\theta_2} \geq \tan(l_{12}) \tag{3.6}$$

or

$$\frac{(\upsilon_1 + q_1)sin\theta_1 - (\upsilon_2 + q_2)sin\theta_2}{(\upsilon_1 + q_1)cos\theta_1 - (\upsilon_2 + q_2)cos\theta_2} \leq \tan(r_{12}) \tag{3.7}$$

In order to obtain non conflict constraints for *n* aircraft we need to consider the conditions described in (6) and (7) for all pairs of aircraft involved in a possible conflict. Let us then consider the general pair of aircraft (i,j). We distinguish two possible cases: 1) $(\upsilon_i$+$q_i)\cos(\theta_i)$- $(\upsilon_j$+$q_j)\cos(\theta_j)$<0 and 2) $(\upsilon_i$+$q_i)\cos(\theta_i)$- $(\upsilon_j$+$q_j)\cos(\theta_j)$>0. If we let h_i=$\tan(l_{ij})\cos(\theta_i)$-$\sin(\theta_i)$, h_j=$\tan(l_{ij})\cos(\theta_j)$-$\sin(\theta_j)$, k_i=$\tan(r_{ij})\cos(\theta_i)$-$\sin(\theta_i)$ and k_j=$\tan(r_{ij})\cos(\theta_j)$-$\sin(\theta_j)$, we obtain the following groups of constraints:

Case 1: $(\upsilon_i$+$q_i)\cos(\theta_i)$- $(\upsilon_j$+$q_j)\cos(\theta_j)$<0

$$\left\{ \begin{array}{c} \cos(\theta_i)q_i - \cos(\theta_j)q_j \leq -\upsilon_i\cos(\theta_i) + \upsilon_j\cos(\theta_j) \\ -h_iq_i + h_jq_j \leq \upsilon_ih_i - \upsilon_jh_j \end{array} \right. \tag{3.8}$$

or

$$\left\{ \begin{array}{c} \cos(\theta_i)q_i - \cos(\theta_j)q_j \leq -\upsilon_i\cos(\theta_i) + \upsilon_j\cos(\theta_j) \\ k_iq_i - k_jq_j \leq -\upsilon_ik_i + \upsilon_jk_j \end{array} \right. \tag{3.9}$$

Case 2: $(\upsilon_i$+$q_i)\cos(\theta_i)$- $(\upsilon_j$+$q_j)\cos(\theta_j)$>0

$$\left\{ \begin{array}{c} -\cos(\theta_i)q_i + \cos(\theta_j)q_j \leq \upsilon_i \cos(\theta_i) - \upsilon_j \cos(\theta_j) \\ h_i q_i - h_j q_j \leq -\upsilon_i h_i + \upsilon_j h_j \end{array} \right. \tag{3.10}$$

or

$$\left\{ \begin{array}{c} -\cos(\theta_i)q_i + \cos(\theta_j)q_j \leq \upsilon_i \cos(\theta_i) - \upsilon_j \cos(\theta_j) \\ -k_i q_i + k_j q_j \leq \upsilon_i k_i - \upsilon_j k_j \end{array} \right. \tag{3.11}$$

These two groups of constraints will be included in the model as *or*-constraints (see Appendix A for more information).

Notice that all constraints obtained are linear in the velocity variation q_i. To conclude the formulation of the VC problem the upper and lower bounds in (3.2) must be considered that are already linear in q_i.

Obviously a solution to the VC problem does not always exist, for example in the case of head-to-head conflict, a change of velocity is not sufficient to avoid the conflict. The inability to handle head-to-head conflicts is one of the major disadvantages of the VC method. However, the HAC method can easily solve these conflicts with a heading angle maneuver.

3.3.2 The HAC Problem

The HAC problem consists of *n* aircraft that fly at the same constant velocity υ and that can maneuver only once with an instantaneous heading angle deviation. The *i*-th aircraft changes its heading angle by a quantity of p_i that can be positive (left turn), negative (right turn) or null (no deviation).

The problem is then to find an acceptable value of p_i for each aircraft such that all conflicts are avoided with the new heading angle (direction of flight), $\theta_i + p_i$. In this section, we will formulate the non-conflict constraints for the HAC problem as inequalities that are linear in the unknowns p_i and that are functions of the aircraft initial configurations (x_i, y_i, θ_i).

As in the previous section we restrict to the case of two aircraft to obtain conflict avoidance conditions and then we will examine the general case of n aircraft. Consider two aircraft denoted 1 and 2, respectively. Let $(x_i, y_i, \theta_i + p_i)$, $i = 1, 2$ be the aircraft's states after the maneuver of amplitude p_i. We will show that it is possible to predict the existence of conflicts between the two aircraft based on their initial configurations. The constraints will be obtained by geometrical construction.

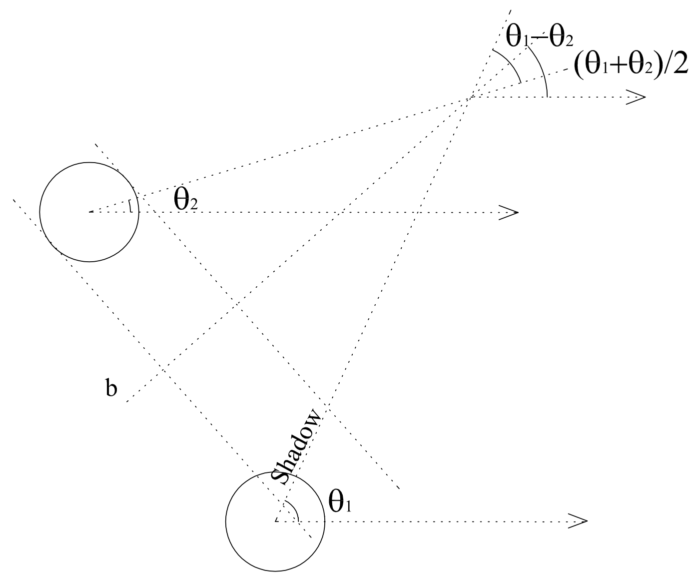

Figure 3.3: Geometric construction for conflict avoidance constraints in the case of intersecting trajectories for the HAC problem. In this case, aircraft 1 intersects the shadow of aircraft 2, and thus a future conflict between the two has been detected.

Referring to Figure 3.3, consider two aircraft (x_1, y_1) and (x_2, y_2) with heading angles θ_1 and θ_2 respectively. Consider for a moment $p_1 = p_2 = 0$ for simplicity (the general equation will be expressed in the next section). Finally, consider the angle of amplitude $(\theta_1 - \theta_2)$ comprised within the aircraft flight directions. The bisector b is then a straight line that forms an angle $(\theta_1 + \theta_2)/2$ with the x-axis, while the orthogonal to the bisector forms an angle of $m_{12} = (\theta_1 + \theta_2 + \pi)/2$ with the x-axis.

The family of straight lines of slope $\tan(m_{12})$, orthogonal to the bisector, represents also the projection of one aircraft along the direction of motion of the other. The two lines in this family that are tangent to aircraft 2 localize a segment on the direction of motion of 1 (see Figure 3.3). We will refer to this segment as the *shadow* of aircraft 2 along the direction of 1. As described earlier, a conflict occurs if aircraft 1 with its safe disc intersects the shadow generated by aircraft 2, or vice-versa since the angle m_{12} is symmetric in θ_1 and θ_2.

Consider again Figure 3.2, where l_{12}=ω_{12}+a, r_{12}=ω_{12}-a and a=$arcsin(\frac{d}{A_{12}})$. A_{12} is the distance between the two aircraft and ω_{12} is the angle between the line that joins the aircraft and the horizontal axis. The condition of non intersection of the shadows is equivalent to the following condition:

$$m_{12} \leq r_{12} \tag{3.12}$$

or

$$m_{12} \geq l_{12} \tag{3.13}$$

where $m_{12} = \frac{\theta_1+\theta_2+\pi}{2}$.

Consider now n aircraft and their initial configurations $(x_i, y_i, \theta_i + p_i), i = 1, 2, ..., n$. We have shown in previous sections that with some geometric considerations it is possible to predict a conflict between pairs of aircraft using only information given by initial states of all n aircraft and the deviations p_i. While the constraints given by and are linear in the heading angle deviation p_i, the constraints expressed above are not explicitly expressed in p_i. We now reformulate them as linear constraints in p_i.

We now consider the general case of n aircraft and deviations p_i. From equations (3.12) and (3.13), no conflict between the aircraft i and aircraft j occurs if

$$\frac{\theta_i + p_i + \theta_j + p_j + \pi}{2} \leq r_{ij} \tag{3.14}$$

or

$$\frac{\theta_i + p_i + \theta_j + p_j + \pi}{2} \geq l_{ij} \tag{3.15}$$

where θ_i has been replaced by the new heading angle $\theta_i + p_i$ after the maneuver of amplitude p_i. Values of l_{ij} and r_{ij} are given by $\omega_{ij} \pm \arcsin(\frac{d}{A_{ij}})$ and $l_{ij} > r_{ij}$ where $A_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ and $\omega_{ij} = \arctan(\frac{y_i - y_j}{x_i - x_j})$.

Let's now define $L_{ij} = l_{ij} - \frac{\theta_i+\theta_j+\pi}{2}$ and $R_{ij} = r_{ij} - \frac{\theta_i+\theta_j+\pi}{2}$. In order to avoid conflicts each pair of aircraft (i, j) with $i < j$ and such that $\frac{\theta_i+p_i+\theta_j+p_j+\pi}{2} \in [-\pi, \pi]$ must satisfy one of the following inequalities:

$$p_i + p_j \leq 2R_{ij} \tag{3.16}$$

or

$$-p_i - p_j \leq -2L_{ij} \tag{3.17}$$

If

$$\frac{\theta_i + p_i + \theta_j + p_j + \pi}{2} \geq \pi \tag{3.18}$$

or if

$$\frac{\theta_i + p_i + \theta_j + p_j + \pi}{2} \leq -\pi \tag{3.19}$$

the quantities R_{ij} and L_{ij} must be shifted by a quantity of π and $-\pi$ respectively, so that we work with angles in $[-\pi, \pi]$. Hence considering all possible cases for the values of $\frac{\theta_i+p_i+\theta_j+p_j+\pi}{2}$ we obtain three groups of constraints:

1. Case $\frac{\theta_i+p_i+\theta_j+p_j+\pi}{2} \in [-\pi, \pi]$:

$$p_i + p_j \leq \pi - \theta_i - \theta_j \tag{3.20}$$
$$-p_i - p_j \leq -3\pi + \theta_i + \theta_j \tag{3.21}$$
$$p_i + p_j \leq 2R_{ij} \tag{3.22}$$

or

$$p_i + p_j \leq \pi - \theta_i - \theta_j \tag{3.23}$$
$$-p_i - p_j \leq -3\pi + \theta_i + \theta_j \tag{3.24}$$
$$-p_i - p_j \leq -2L_{ij} \tag{3.25}$$

2. Case $\frac{\theta_i+p_i+\theta_j+p_j+\pi}{2} > \pi$:

$$-p_i - p_j \leq -\pi + \theta_i + \theta_j \tag{3.26}$$
$$p_i + p_j \leq 2R_{ij} + 2\pi \tag{3.27}$$

or

$$-p_i - p_j \leq -\pi + \theta_i + \theta_j \tag{3.28}$$
$$-p_i - p_j \leq -2L_{ij} - 2\pi \tag{3.29}$$

3. Case $\frac{\theta_i+p_i+\theta_j+p_j+\pi}{2} < -\pi$:

$$p_i + p_j \leq -3\pi - \theta_i - \theta_j \tag{3.30}$$
$$p_i + p_j \leq 2R_{ij} - 2\pi \tag{3.31}$$

or

$$p_i + p_j \leq -3\pi - \theta_i - \theta_j \tag{3.32}$$
$$-p_i - p_j \leq -2L_{ij} + 2\pi \tag{3.33}$$

These three groups of constraints will be included in the model as *or*-constraints.

The model of the HAC problem in now complete. In the case of heading angle maneuvers we can consider also other kinds of constraints. For example we can consider the possible existence of forbidden zones of airspace due to severe weather conditions, overloaded space or even military restricted zones (see Figure 3.4). To model these forbidden zones, it is sufficient to consider bounds of the heading angle deviations p_i.

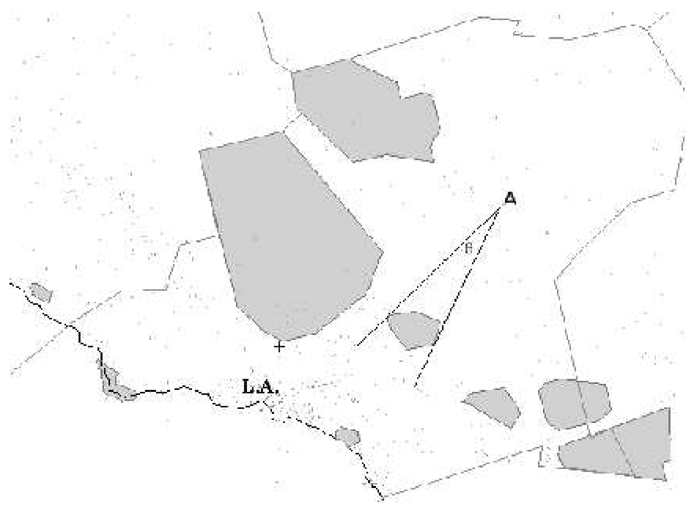

Figure 3.4: Example of forbidden sectors in the Los Angeles control sector. For aircraft A we need to introduce more constraints on the direction of flight due to forbidden zones of airspace.

3.4 The Unified Approach

As mentioned earlier, the method we will use is in effect a combination of the VC and HAC methods that have been previously analyzed. Here, we will present the final configuration and calibration of the unified method, proceed to some qualitative changes in areas such as choosing the cost function and finally comment on the produced results.

3.4.1 The Final Conflict Constraints For The Unified Approach

First of all, we will include in our model the conflict constraints of the VC model, namely equations (3.8-3.11). The difference here is that we replace the initial angle θ_i in the VC problem (remember that the heading angle is constant in VC), with the quantity $\theta_i + p_i$, where p_i is the angle deviation of the $i-th$ aircraft from its nominal path. So, the actual set of constraints used for simulation purposes is this:

Case 1: $(\upsilon_i + q_i)\cos(\theta_i + p_i) - (\upsilon_j + q_j)\cos(\theta_j + p_j) < 0$

$$\left\{ \begin{array}{c} \cos(\theta_i + p_i)q_i - \cos(\theta_j + p_j)q_j \leq -\upsilon_i \cos(\theta_i + p_i) + \upsilon_j \cos(\theta_j + p_j) \\ -h_i q_i + h_j q_j \leq \upsilon_i h_i - \upsilon_j h_j \end{array} \right. \tag{3.34}$$

or

$$\left\{ \begin{array}{c} \cos(\theta_i + p_i)q_i - \cos(\theta_j + p_j)q_j \leq -\upsilon_i \cos(\theta_i + p_i) + \upsilon_j \cos(\theta_j + p_j) \\ k_i q_i - k_j q_j \leq -\upsilon_i k_i + \upsilon_j k_j \end{array} \right. \tag{3.35}$$

Case 2: $(\upsilon_i + q_i)\cos(\theta_i + p_i)$- $(\upsilon_j + q_j)\cos(\theta_j + p_j) > 0$

$$\left\{ \begin{array}{c} -\cos(\theta_i + p_i)q_i + \cos(\theta_j + p_j)q_j \leq \upsilon_i \cos(\theta_i + p_i) - \upsilon_j \cos(\theta_j + p_j) \\ h_i q_i - h_j q_j \leq -\upsilon_i h_i + \upsilon_j h_j \end{array} \right. \tag{3.36}$$

or

$$\left\{ \begin{array}{c} -\cos(\theta_i + p_i)q_i + \cos(\theta_j + p_j)q_j \leq \upsilon_i \cos(\theta_i + p_i) - \upsilon_j \cos(\theta_j + p_j) \\ -k_i q_i + k_j q_j \leq \upsilon_i k_i - \upsilon_j k_j \end{array} \right. \tag{3.37}$$

As noted earlier, only one of these sets of constraints will be used in our model for each instance. Thus, according to which of the two cases (1 or 2) holds true, we will use the first (equations 3.35-3.36) or the second (equations 3.37-3.38) set.

We will also include in our model the constraints of the HAC problem. Note that in this case we don't have to modify them by replacing the constant velocity that all the aircraft have in the HAC problem by a variant one, since the velocity does not appear in these constraints. We will therefore use the equations (3.20-3.33) as described earlier. It is necessary to bear in mind that so far, we have derived the final set of conflict constraints we want to keep for our unified model, but we still lack two important things. The first one is the formulation of these constraints in a suitable manner so that they can be used as input to an optimization software package in order to produce the desired results and the second is the choice of the cost function, namely what metric we want to optimize. We will start with the latter, since it is easier to be addressed first. It is true that there are quite a number of available choices for the cost function. In the ATC literature, there have been metrics such as the values of the velocity changes q_i (with a respective cost function of the form $\sum_{i=1}^{n} -q_i$), the infinity norm of the vector of the heading angle changes p_i, for $i = 1, ..., n$ (with a respective cost function of the form $max(|p_1|, ..., |p_n|)$, or even the 1-norm of the vector $p = (p_1, ..., p_n)$ (with a respective cost function of the form $\sum_{i=1}^{n} |p_i|$). After much consideration we decided to use a more complicated metric. Specifically, we minimize for all the aircraft, the total difference in flight time between the ideal time it would take them to reach their final configuration points if there weren't any conflicts and the actual time it will take them in a scenario where they must be able to avoid all conflicts. If we denote this difference by $d = t_{ideal} - t_{actual}$, we will try to minimize the cost function $\sum_{i=1}^{n} d_i$ for all of the n aircraft. We end this brief discussion by noting that our model is now a non-linear one (in fact as we will see later, it will be a mixed-integer non-linear one) so we must be prepared for larger executional times and generally slower performance, which will have an impact upon the system's potential use in real time conditions.

3.4.2 Development Platform And Reformulation Of The Final Conflict Constraints

At this point, we are faced with a non-linear optimization problem and we wish to solve it computationally as efficiently as possible. Because the problem has a very large number of constraints that increases exponentially with the number of aircraft involved, it is imperative that we use a software optimization package in order to solve it. There are, indeed, many such tools available. We have chosen the GAMS software package (www.gams.de), which is essentially a front end for solvers such as CPLEX, dicopt etc. Its friendly interface and accessibility make it an ideal tool for the user who does not wish the full processing power of professional high-end products.

We will now examine how the final conflict constraints should be reformulated in order for us to be able to use them as input to GAMS. We assume that the reader is familiar with the basics of linear and non-linear optimization problems. If, however, this is not the case, the reader is encouraged to recur to Appendix A, for a brief summary.

GAMS, as any other optimization package, requires that the constraints present for any problem are all satisfied simultaneously (*and*-constraints). In other words it is able to solve optimization problems of the form:

$$\min f(x) \tag{3.38}$$

such that

$$g(x) \leq 0 \tag{3.39}$$

where $f(x)$ is a function of n real variables $x = (x_1, x_2, ..., x_n) \in R^n$ and is subject to a set of inequality constraints $g(x) \leq 0$ $(g_j(x) \leq 0, j = 1, 2, ..., p)$. This means that the constraints $g_j(x)$ must all be valid at the same time (g_1 AND g_2 AND ... AND g_p). Clearly in our case, where we have some *or*-constraints, a reformulation is necessary. We therefore shall have to introduce some boolean variables to convert these *or*-constraints to *and*-constraints. A simple example will be presented for comprehensive purposes.

Let us assume that we have the following sets of constraints:

$$\begin{array}{c} c_1 \leq 0 \\ and \\ c_2 \leq 0 \end{array} \tag{3.40}$$

OR

$$\begin{gathered} c_3 \leq 0 \\ and \\ c_4 \leq 0 \\ and \\ c_5 \leq 0 \end{gathered} \tag{3.41}$$

OR

$$\begin{gathered} c_6 \leq 0 \\ and \\ c_7 \leq 0 \end{gathered} \tag{3.42}$$

where the terms $c_i, i = 1, ..., 7$, are expressions (linear or non-linear) with respect to the decision variables.

The way to transform these *or-constraints* into more convenient *and-constraints* is to introduce boolean variables [16-17]. Let f_k with $k = 1, 2, 3$ be a binary number that becomes zero when one of the respective *or-constraints* is active and 1 otherwise (i.e. $f_1 = 0$ if constraints c_1 and c_2 are active, $f_1 = 1$ otherwise). Letting M be a large arbitrary number, the previous set of constraints is equivalent to:

$$\begin{gathered} c_1 - Mf_1 \leq 0 \\ c_2 - Mf_1 \leq 0 \\ c_3 - Mf_2 \leq 0 \\ c_4 - Mf_2 \leq 0 \\ c_5 - Mf_2 \leq 0 \\ c_6 - Mf_3 \leq 0 \\ c_7 - Mf_3 \leq 0 \\ f_1 + f_2 + f_3 \leq 2 \end{gathered} \tag{3.43}$$

The above constraints are all *and-constraints* so we have overcome the previous difficulty. It is however obvious that now we are faced with a so-called Mixed Integer Programming (MIP) problem, because we have two different kinds of variables: normal variables that can take any value and binary variables $(f_1 - f_3)$ that can only take the values 0 or 1. MIP problems are considerably more complex than both the Pure Integer Programming problems (where the decision variables can **only** take binary values) and the classic LP or NLP problems. Again, for a more detailed explanation of the subject, we refer the reader to Appendix A.

Using the aforementioned notion of binary variables, we are able to transform all of our *or-constraints* into *and-constraints* and therefore to provide the optimization software package with all the data it needs in order to compute the decision variables for each case. In the next section we will present the results produced by the optimization software package for various topology scenarios and we will comment on their characteristics in terms of efficiency, complexity, execution time etc.

3.5 Simulation and Case Studies

The basic topology used for simulation purposes is shown at Figure 3.5. The idea here is that we have a circle of given radius R (referred to as "control circle" from now on) in which we scan for possible conflicts. Aircraft going through the control circle have an initial and a final configuration. The initial configuration of an aircraft consists of its velocity and its heading angle at the point of entry in the circle, while its final configuration consists of its velocity and its heading angle at the point that lies at the original aircraft direction at distance $1.5d_i$ from the entry point, where d_i is the length of the i-th aircraft's trajectory *inside* the circle.

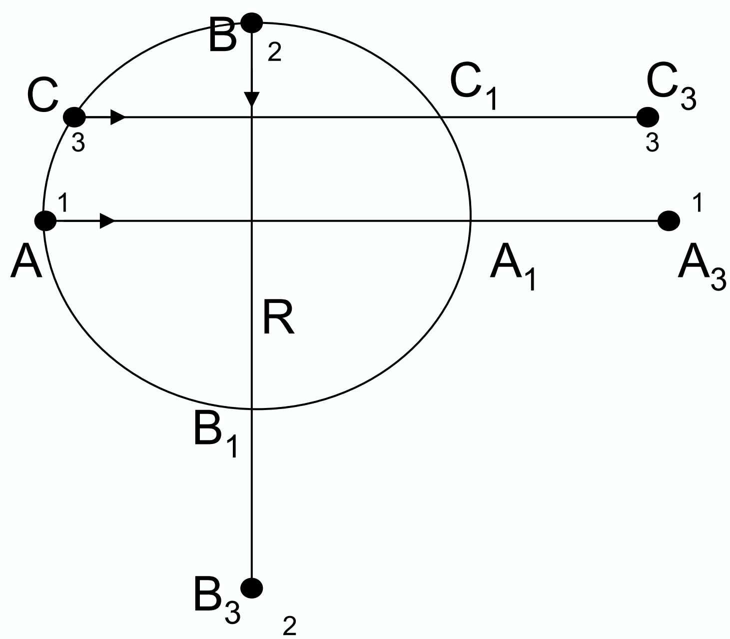

Figure 3.5: Initial and final configuration points for 3 aircraft traversing the control circle.

For example, in Figure 3.5, the initial configuration points for aircraft 1,2 and 3 are A,B and C respectively, while the final configuration points are A_3, B_3 and C_3 respectively (where $AA_3 = 1.5AA_1$, $BB_3 = 1.5BB_1$ and $CC_3 = 1.5CC_1$). Any aircraft that enters the control circle must check for possible conflicts with the other aircraft that are inside the circle at the moment. It will then make an appropriate maneuver according to the results produced by the solution of the mixed-integer NLP problem and will continue to travel with the new velocity and heading angle until it reaches the point of exit from the circle. If a second aircraft enters the circle, while the first one is still in it and there exists a possibility of conflict between these two, then the first aircraft will have to make a new maneuver in order to avoid the new conflict and so on until it reaches the end of the circle. When it finally arrives at the exit point, it will make a "corrective" change of heading angle and velocity in order to reach its final configuration point which lies outside the circle. Consider Figure 3.6 for a comprehensive example:

Aircraft 1 and 2 enter simultaneously the control circle with the same velocity ($v_1 = v_2$) and they follow the trajectories shown, with heading angles of 0 and -$\pi/2$ respectively. Clearly, if no changes are made, a conflict will occur at the center of the circle. However, the aircraft aided by the optimization software make instantaneous changes both to their velocities and to their heading angles and avoid the conflict. Note that this is true in the general case but it does not mean that both attributes **have** to change. Depending on the nature of the problem, one may find that the optimum maneuver consists only of a change in velocity and no heading angle change or vice-versa. It is up to the software to point out such a case, by providing zero values to the respective aircraft attributes. In our example, we assume that changes have been made both to the velocities of the aircraft and to their heading angles. So, after the conflict resolution, they travel respectively with velocities $v_1 + q_1$ and $v_2 + q_2$ and heading angles p_1 and $p_2 - \pi/2$ until they exit the control circle (points A_2 and B_2 respectively). New changes must then be made, in order for the aircraft to reach their final configuration points, since the change in the heading angles has forced them to deviate from their nominal paths. Therefore, the aircraft turn by a factor of magnitude of $p_1^{'}$ and $p_2^{'}$ respectively and change their velocity by a factor of magnitude $q_1^{'}$ and $q_2^{'}$ respectively. When eventually, they reach their final configuration points A_3 and B_3, they must make yet another turn by a factor of magnitude $p_1^{'} - p_1$ and $p_2^{'} - p_2$ respectively, so that their nominal path can be fully resumed.

The unknown variables in this scenario are: the initial heading angle (p_1,p_2) and velocity (q_1, q_2) changes, and the exit points changes ($p_1^{'}$, $p_2^{'}$,$q_1^{'}$, $q_2^{'}$). Overall, there are 8 unknowns for a pair of aircraft.

In the following subsections we will discuss some case studies for various aircraft configuration scenarios and we will present the results obtained from the

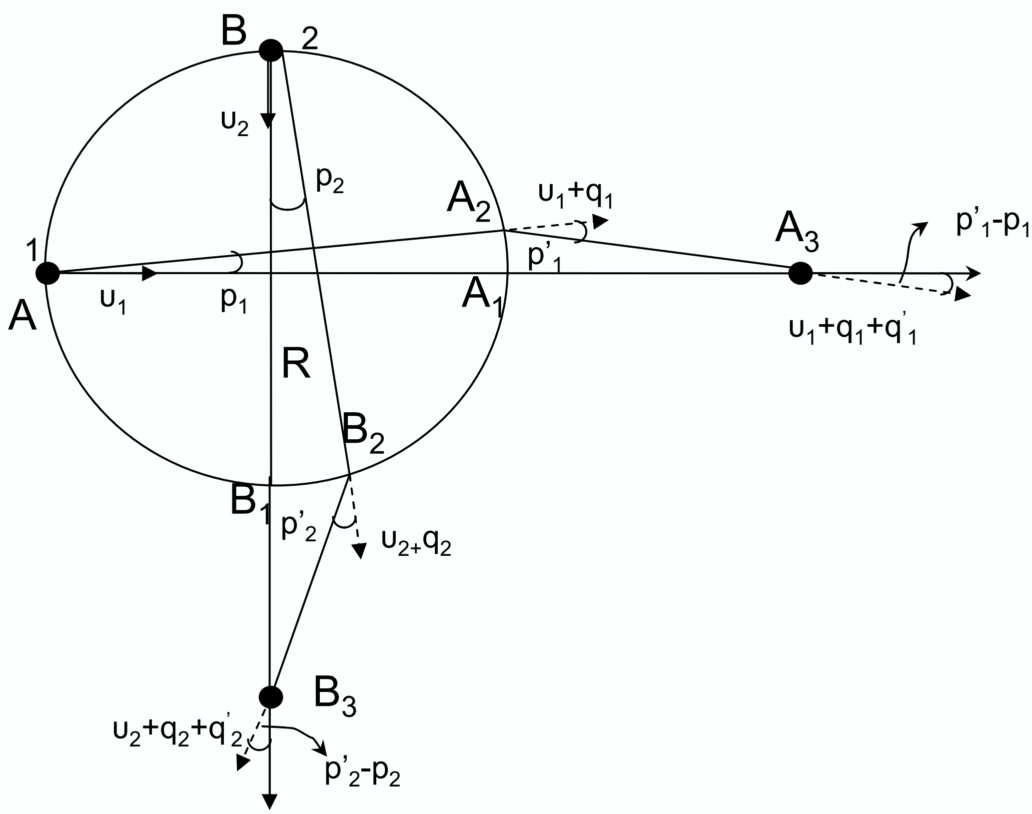

Figure 3.6: Simplified scenario of conflict resolution between 2 aircraft with magnitude of maneuvers provided by the optimization software.

corresponding simulations conducted with the aid of the optimization software package. Four cases are studied where the aircraft involved are symmetrically placed onto the control circle and three cases where the aircraft are positioned randomly. The various plots were created within the MATLAB environment from the data acquired by GAMS.

3.5.1 Case Study 1: Three symmetrically distributed aircraft originally travelling through control circle center

In the following simulations all aircraft are assumed to enter the control circle with the same speed (the maximum speed allowed by the restrictions for passenger comfort/safety and fuel consumption). The control circle has a radius of $60nm$ or $108kM$ and the minimum safety distance has been set to $5nm$ or $9kM$. Two plots are presented for each case study, one that shows the aircraft configuration and their projected trajectories if no maneuvers are made to avoid possible conflicts and one that shows the corresponding situation after the various heading angle and speed maneuvers. The final configuration points are not presented in the plots since we are "visually" interested in the motion of the aircraft only inside the control circle. Each case study is accompanied by a table that shows the velocities and heading angles of the aircraft before and after the conflict resolution, so as to compare the various cases.

In Figure 3.7 we see three symmetrically distributed aircraft which are all headed towards the origin. Since the initial velocities are the same, there will clearly be a conflict between them at the origin if no maneuvers are made.

In Figure 3.8 we see the aircraft and their trajectories after the maneuvers for conflict resolution. It is important to remember that an aircraft need not necessarily change its trajectory in order to avoid a conflict; it can merely change its speed (accelerate or decelerate). This means that in all the following plots, identical trajectories between two consecutive plots do not imply absence of maneuver in general, but rather absence of heading angle change. In the specific example that we study, the conflicts were resolved both by velocity and by heading angle changes. The values of the velocities and heading angles of the aircraft before and after the conflict resolution are shown in Figure 3.9.

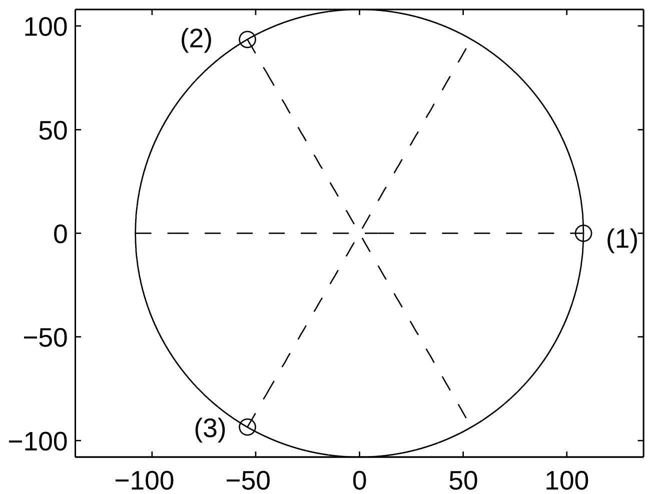

Figure 3.7: Three symmetrically distributed aircraft and their projected trajectories before conflict resolution.

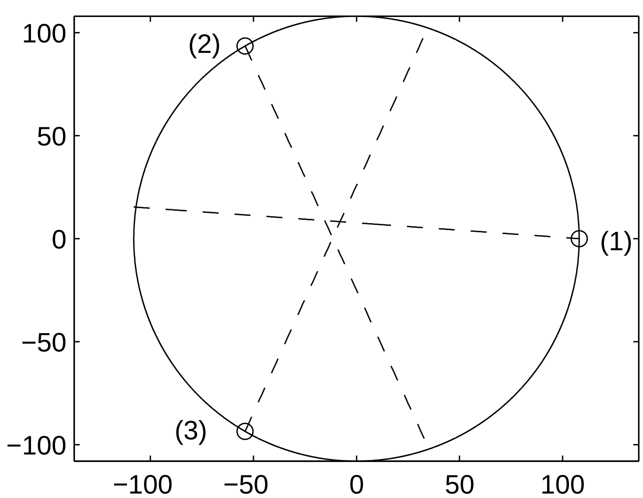

Figure 3.8: Three symmetrically distributed aircraft and their projected trajectories after conflict resolution.

Aircraft / Attributes	Aircraft 1	Aircraft 2	Aircraft 3
Velocity before CR (kM/min)	15	15	15
Change in velocity after CR (kM/min)	-0.21	-0.126	-0.115
HA before CR (rad)	3.14	-1.047	1.047
Change in HA after CR (rad)	-0.071	-0.096	0.099

Figure 3.9: Table showing velocities and heading angles for three symmetrically distributed aircraft. CR stands for Conflict Resolution and HA for Heading Angle.

3.5.2 Case Study 2: Five symmetrically distributed aircraft originally travelling through control circle center

Similarly, in Figure 3.10 we are presented with five aircraft symmetrically distributed on the control circle. Here, all aircraft change both their heading angle and velocities in order to avoid the impending conflict as seen in Figure 3.11. The values of the velocities and heading angles of the aircraft before and after the conflict resolution are shown in Figure 3.12. From this figure, we can conclude that practically aircraft 1 does not change its heading angle (only 0.005 rad) and thus resolves the conflict almost entirely by velocity change. Notice also that in this case, the absolute values of the changes in the aircraft velocities and heading angles are in aggregate larger than in the previous case of three aircraft.

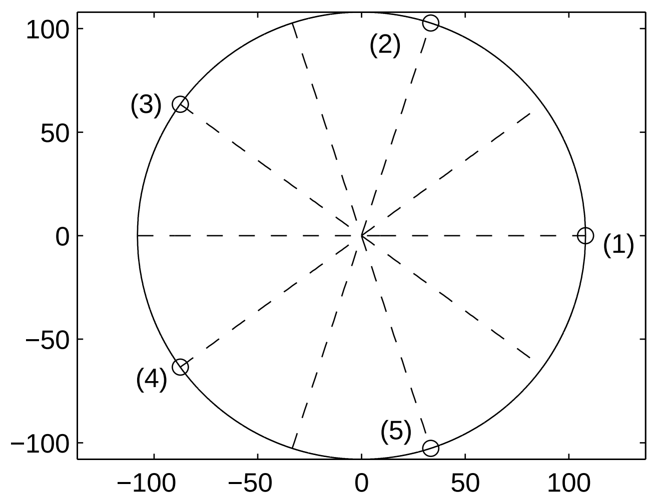

Figure 3.10: Five symmetrically distributed aircraft and their projected trajectories before conflict resolution.

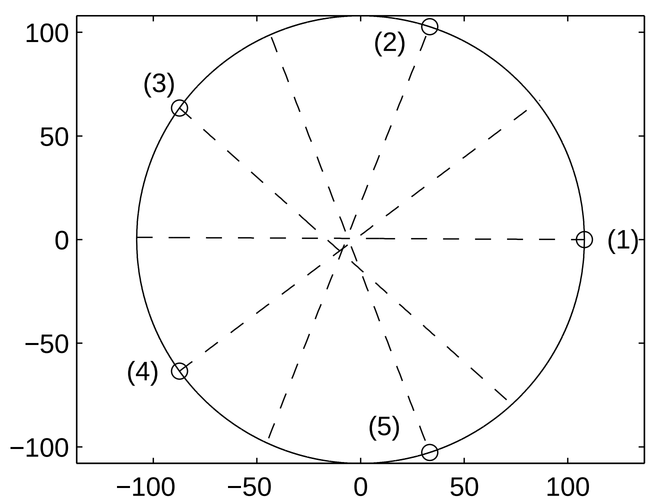

Figure 3.11: Five symmetrically distributed aircraft and their projected trajectories after conflict resolution.

Aircraft / Attributes	Aircraft 1	Aircraft 2	Aircraft 3	Aircraft 4	Aircraft 5
Velocity before CR (kM/min)	15	15	15	15	15
Change in velocity after CR (kM/min)	-0.309	-0.24	-0.11	-0.304	-0.259
Heading angle before CR (rad)	0	-1.885	-0.628	0.628	1.885
Change in heading angle after CR (rad)	-0.005	-0.059	-0.1	0.016	0.05

Figure 3.12: Table showing velocities and heading angles for five symmetrically distributed aircraft. CR stands for Conflict Resolution.

3.5.3 Case Study 3: Seven symmetrically distributed aircraft originally travelling through control circle center

In Figure 3.13 we have seven aircraft symmetrically distributed on the control circle. Again all aircraft change their velocities and heading angles as seen in Figure 3.14, however some of them practically don't change their trajectory since the amount of change in their heading angle is very small. The values of all the velocities and heading angles of the aircraft before and after the conflict resolution are shown in Figure 3.15. Similarly to the previous cases, the absolute values of the changes in the aircraft velocities and heading angles are in aggregate larger than in the previous cases.

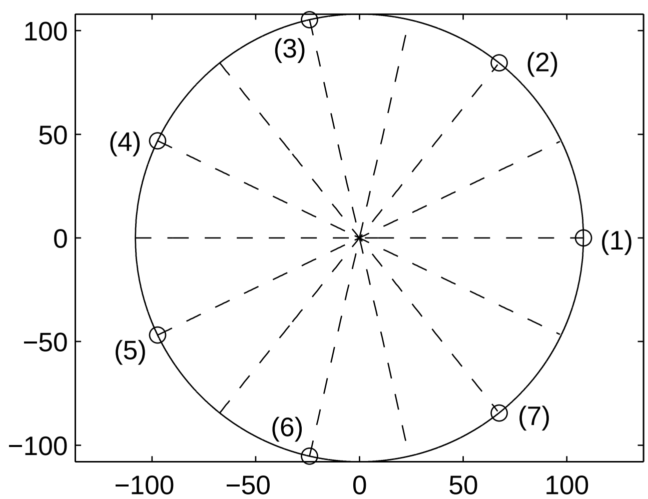

Figure 3.13: Seven symmetrically distributed aircraft and their projected trajectories before conflict resolution.

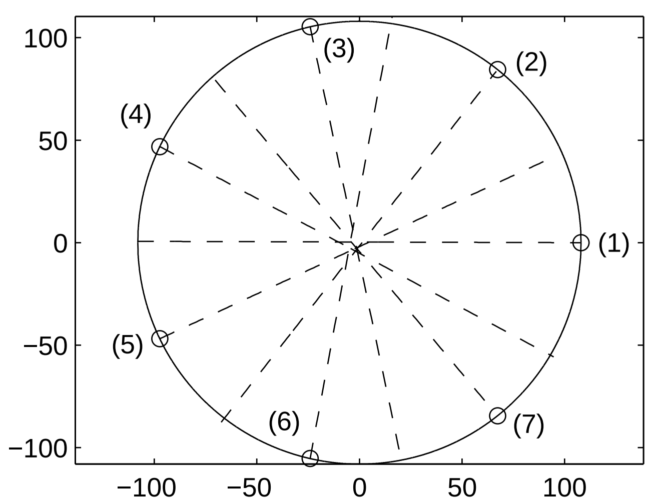

Figure 3.14: Seven symmetrically distributed aircraft and their projected trajectories after conflict resolution.

Attributes \ Aircraft	A.C. 1	A.C. 2	A.C. 3	A.C. 4	A.C. 5	A.C. 6	A.C. 7
Velocity before CR (kM/min)	15	15	15	15	15	15	15
Change in velocity after CR (kM/min)	-0.309	-0.308	-0.304	-0.274	-0.304	-0.276	-0.298
Heading angle before CR (rad)	3.14	-2.244	-1.346	-0.448	0.448	1.346	2.244
Change in heading angle after CR (rad)	-0.003	0.009	-0.017	-0.042	-0.016	0.041	0.026

Figure 3.15: Table showing velocities and heading angles for seven symmetrically distributed aircraft. CR stands for Conflict Resolution.

3.5.4 Case Study 4: Nine symmetrically distributed aircraft originally travelling through control circle center

Finally, in Figure 3.16 we have nine aircraft symmetrically distributed on the control circle. Here, aircraft 2, 3, 4, 6, 7, 8 and 9 are deviating from their paths, while aircraft 1 and 5 resolve the conflict practically only with a change in their speed, maintaining their original trajectories as seen in Figure 3.17. The values of the velocities and heading angles of the aircraft before and after the conflict resolution are shown in Figure 3.18. Similarly to the previous cases, the absolute values of the changes in the aircraft velocities and heading angles are in aggregate larger than in the previous cases, which shows that the greater the number of the aircraft involved in a potential conflict, the larger the magnitude of the aircraft maneuvers necessary for the conflict avoidance. In other words, the aircraft try to use up as much "space" for their maneuvers as possible (subject of course to the limitations imposed for passenger safety and comfort).

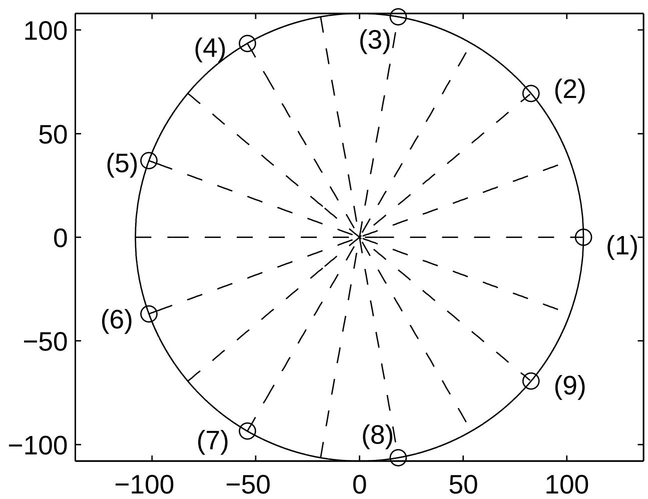

Figure 3.16: Nine symmetrically distributed aircraft and their projected trajectories before conflict resolution.

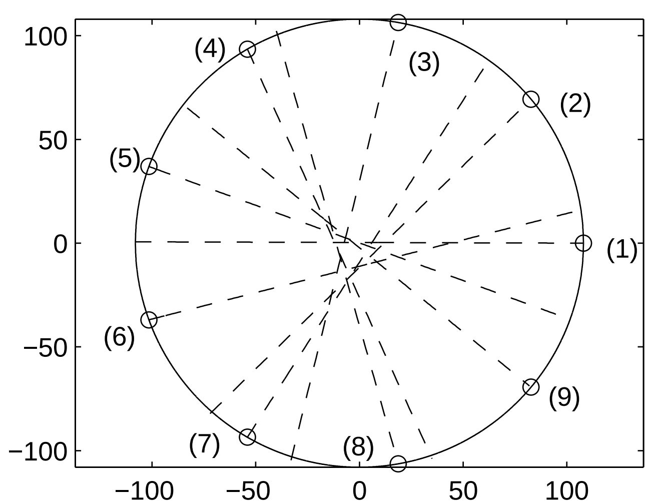

Figure 3.17: Nine symmetrically distributed aircraft and their projected trajectories after conflict resolution.

Attributes \ Aircraft	A.C. 1	A.C. 2	A.C. 3	A.C. 4	A.C. 5	A.C. 6	A.C. 7	A.C. 8	A.C. 9
Velocity before CR (kM/min)	15	15	15	15	15	15	15	15	15
Change in velocity after CR (kM/min)	-0.309	-0.192	-0.222	-0.11	-0.6	0.66	-0.6	0.66	-0.304
HA before CR (rad)	3.14	-2.443	-1.745	-1.047	-0.349	0.349	1.047	1.745	2.443
Change in HA after CR (rad)	-0.003	0.077	-0.066	-0.1	0.001	-0.1	-0.046	0.1	0.016

Figure 3.18: Table showing velocities and heading angles for nine symmetrically distributed aircraft. CR stands for Conflict Resolution and HA for Heading Angle.

3.5.5 Case Study 5: Three randomly distributed aircraft

The next three case studies are conducted with random aircraft configurations. In the first of those, in Figure 3.19 there is one conflict between aircraft 2 and 3 which is resolved with heading angle and velocity maneuvers by both aircraft, while aircraft 1 does not change neither its velocity nor its heading angle, as seen in Figure 3.20. The values of the velocities and heading angles of the aircraft before and after the conflict resolution are shown in Figure 3.21.

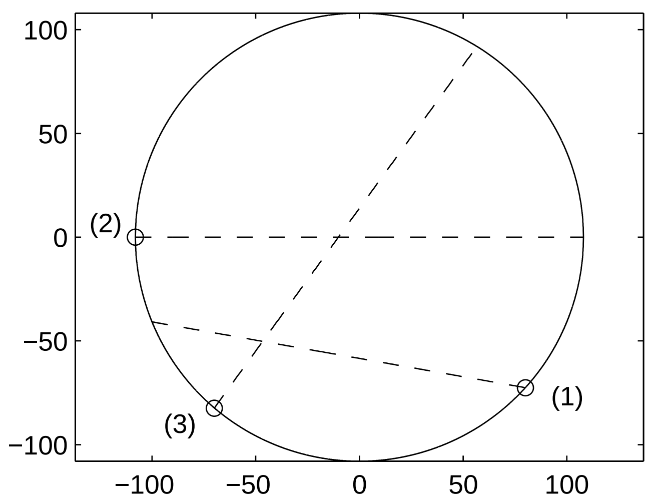

Figure 3.19: Three randomly distributed aircraft and their projected trajectories before conflict resolution.

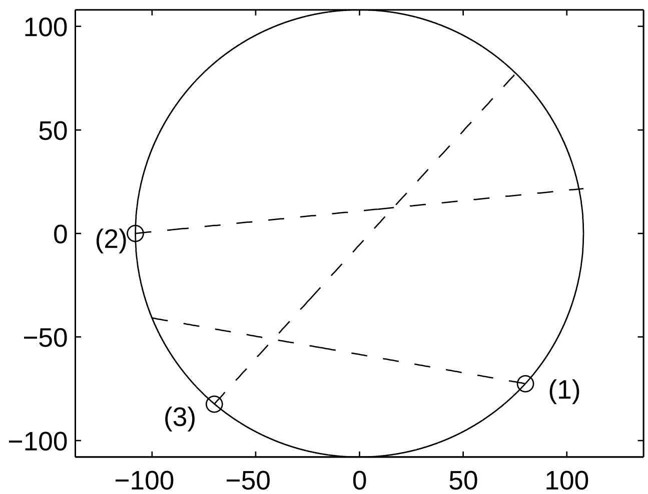

Figure 3.20: Three randomly distributed aircraft and their projected trajectories after conflict resolution.

Attributes \ Aircraft	Aircraft 1	Aircraft 2	Aircraft 3
Velocity before CR (kM/min)	15	15	15
Change in velocity after CR (kM/min)	0	-0.11	-0.11
Heading angle before CR (rad)	2.967	0	0.942
Change in heading angle after CR (rad)	0	0.1	0.1

Figure 3.21: Table showing velocities and heading angles for three randomly distributed aircraft. CR stands for Conflict Resolution.

3.5.6 Case Study 6: Five randomly distributed aircraft

In the random configuration of Figure 3.22 there is one conflict between aircraft 3 and 5 which is resolved with velocity and heading angle maneuvers by both aircraft, while aircraft 1, 2 and 4 do not change neither their velocity nor their heading angle, as seen in Figure 3.23. The values of the velocities and heading angles of the aircraft before and after the conflict resolution are shown in Figure 3.24.

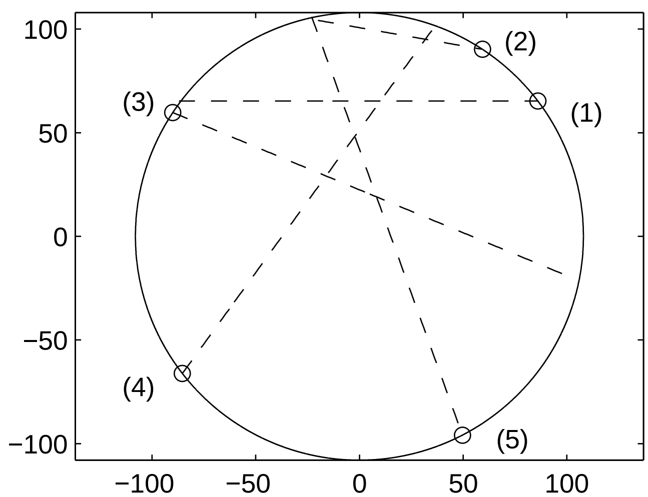

Figure 3.22: Five randomly distributed aircraft and their projected trajectories before conflict resolution.

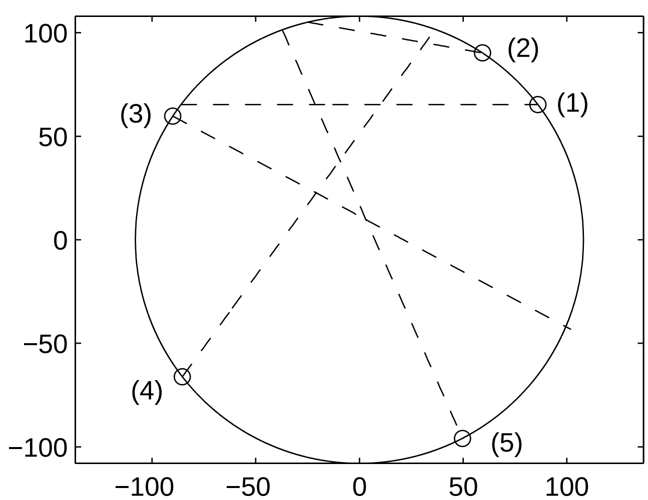

Figure 3.23: Five randomly distributed aircraft and their projected trajectories after conflict resolution.

Attributes \ Aircraft	Aircraft 1	Aircraft 2	Aircraft 3	Aircraft 4	Aircraft 5
Velocity before CR (kM/min)	15	15	15	15	15
Change in velocity after CR (kM/min)	0	0	-0.6	0	-0.212
Heading angle before CR (rad)	3.14	2.967	-0.392	0.942	1.916
Change in heading angle after CR (rad)	0	0	-0.1	0	0.069

Figure 3.24: Table showing velocities and heading angles for five randomly distributed aircraft. CR stands for Conflict Resolution.

3.5.7 Case Study 7: Seven randomly distributed aircraft

In the random configuration of Figure 3.25 there are two conflicts: one between aircraft 2 and 3 which is resolved with velocity and heading angle maneuvers by both aircraft and one between aircraft 5 and 6 which is resolved in a similar manner, as seen in Figure 3.26. Aircraft 1, 4 and 7 do not maneuver at all. Figure 3.27 shows the values of the velocities and heading angles of the aircraft before and after the conflict resolution.

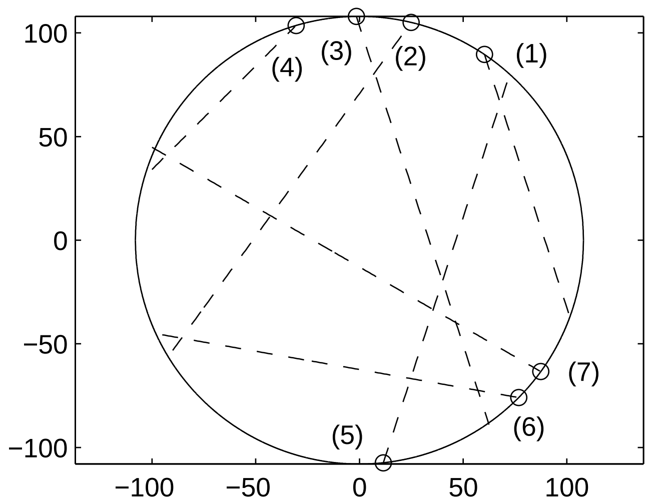

Figure 3.25: Seven randomly distributed aircraft and their projected trajectories before conflict resolution.

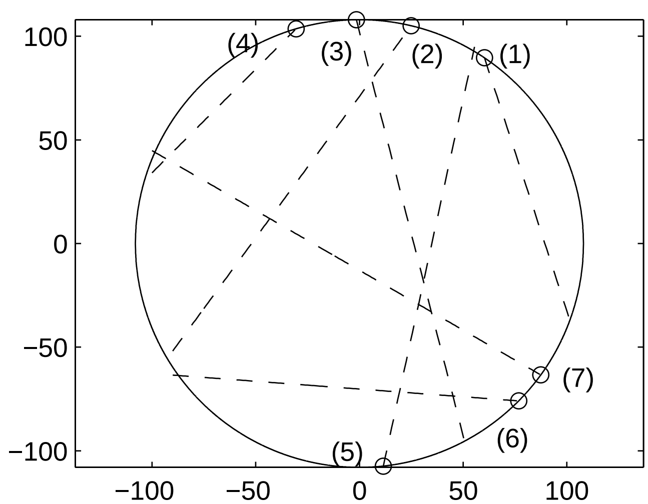

Figure 3.26: Seven randomly distributed aircraft and their projected trajectories after conflict resolution.

Attributes \ Aircraft	A.C. 1	A.C. 2	A.C. 3	A.C. 4	A.C. 5	A.C. 6	A.C. 7
Velocity before CR (kM/min)	15	15	15	15	15	15	15
Change in velocity after CR (kM/min)	0	-0.309	-0.11	0	-0.11	-0.23	0
HA before CR (rad)	-1.256	-2.2	1.256	-2.356	2.617	-1.256	2.967
Change in HA after CR (rad)	0	-0.004	0.1	0	0.1	-0.063	0

Figure 3.27: Table showing velocities and heading angles for seven randomly distributed aircraft. CR stands for Conflict Resolution and HA for Heading Angle.

n	Time (sec)	Δθ (rad)	Δυ (kM/min)	Variables
3	0.5	0.02	0.2	55
3*	1.03	0.04	0.25	55
5	2.81	0.05	0.51	95
5*	4.13	0.06	0.65	95
7	5.74	0.07	0.8	135
7*	8.34	0.07	0.8	135
9	9.89	0.1	1.1	175

Figure 3.28: Table presenting quantitative results of simulations. The star next to the number of aircraft denotes a random configuration.

3.5.8 Quantitative Results

In the previous subsections we presented various plots of aircraft configurations and trajectories before and after conflict resolution, based on data supplied by the GAMS software. In other words, we studied the problem in a qualitative aspect. In this subsection we will briefly present and discuss the quantitative results of the simulations, namely, the magnitudes of the various heading angle and velocity maneuvers, number of total variables for each scenario and computational times required for the problem's solution.

Figure 3.28 provides a handy overview of these quantitative results. We indicate the computational time (in seconds) of GAMS to find the optimal solution to the MIP problem, the number of the aircraft involved in each scenario (denoted by n), the maximum absolute angular deviations ($\Delta\theta$) and velocity changes (Δv) and finally the total number of constraints for each scenario.

Commenting on these results, we would firstly have to note that the computational times are somewhat larger when compared to other similar approaches [7-10]. This, however, is unavoidable since here, we are dealing with a non-linear problem whereas in all the previous work, the assumptions and simplifications produced linear problems which by default have a smaller degree of complexity. However, the other quantitative results, maximum angular and velocity changes, are indeed smaller than those indicated by [7], which is a logical thing, since here we allow for both kinds of maneuvers (heading angle and velocity) thus making it possible to "break down" the magnitude of maneuver into two pieces. All other simulations have been conducted for one kind of maneuver exclusively, ignoring the existence of the other.

3.5.9 Future Work

In spite of the implementation and functionality of our method, challenges remain in the field and there is still a good amount of work to be done in order to develop a fully working framework for aircraft conflict avoidance.

Of great relevance is the work in progress also at the Technical University of Crete, that addresses the problem in the three-dimensional space and takes into account the different altitudes that aircraft travel, thus making it possible to implement potential conflict situations involving the takeoff and landing of aircraft.

In another direction, more case studies should be examined, with more complex configuration patterns and a greater number of aircraft so as to gain more insight in the algorithm and the way it works.

The integration of this method into a software package and its implementation in real-time simulations, would be the logical next step, in order to determine its usefulness and its potential application in commercial flights.

In any case, the field of conflict avoidance and free flight in general, is currently evolving and new material is constantly coming to surface. The near future will indicate the final form of the free flight concept and the tools that will be utilized in its implementation.

Appendix A

Linear And Non-Linear Programming

Before we examine the areas of mixed-integer and non-linear programming, we will present very briefly the basic ideas of linear programming, so that the basis for the more advanced concepts will be understood.

A.1 Linear Programming

Linear programming uses a mathematical model to describe the problem of concern. The adjective *linear* means that all the mathematical functions in this model are required to be *linear functions*. The word *programming* does not refer here to computer programming; rather, it is essentially a synonym for *planning*. Thus linear programming involves the *planning of activities* to obtain an optimal result, i.e., a result that reaches the specified goal best (according to the mathematical model) among all feasible alternatives.

The mathematical model of a linear programming problem is the system of equations and related mathematical expressions that describe the essence of the problem.Thus, if there are n related quantifiable decisions to be made, they are represented as **decision variables** (i.e. $x_1, x_2, ..., x_n$) whose respective values are to be determined. The appropriate measure of performance (i.e. profit) is then expressed as a mathematical function of these decision variables (for example, $P = 3x_1 + 2x_2 + ... + 5x_n$). This function is called the **objective function**. Any restrictions on the values that can be assigned to these decision variables are also expressed mathematically, typically by means of inequalities or equations (for example $x_1 + 3x_1x_2 + 2x_2 \leq 10$). Such mathematical expressions for the restrictions are often called **constraints**. The constants (namely, the coefficients and right-hand sides) in the constraints and the objective function are called the

parameters of the model. The mathematical model might then say that the problem is to choose the values of the decision variables so as to maximize the objective function, subject to the specified constraints.

A.1.1 A Comprehensive Example

A glass factory produces high-quality glass products, including windows and glass doors. It has three plants. Aluminum frames and hardware are made in Plant 1, wood frames are made in Plant 2 and Plant 3 produces the glass and assembles the products.

Because of declining earnings, top management has decided to revamp the company's product line. Unprofitable products are being discontinued, releasing production capacity to launch two new products having large sales potential:

Product 1: An 8-foot glass door with aluminum framing

Product 2: A 4X6 foot double-hung wood-framed window

Product 1 requires some of the production capacity in Plants 1 and 3, but none in Plant 2. Product 2 needs only Plants 2 and 3. The marketing division has concluded that the company could sell as much of either product as could be produced by these plants. However, because both products would be competing for the same production capacity in Plant 3, it is not clear which *mix* of the two products would be *most profitable*. The necessary data are presented in Figure A1. We formulate the mathematical (linear programming) model for this problem as follows:

x_1=number of batches of product 1 produced per week

x_2=number of batches of product 2 produced per week

Z=total profit per week (in thousands of dollars) from producing these two products

Thus x_1 and x_2 are the *decision variables* for the model. Using the bottom row of the table in Figure A1, we obtain $Z = 3x_1 + 5x_2$.

Plant	Production Time per Batch, Hours		Production Time Available per Week, Hours
	Product		
	1	2	
1	1	0	4
2	0	2	12
3	3	2	18
Profit per batch	3000 $	5000 $	

Figure A.1: Data for the glass factory problem.

The objective is to choose the values of x_1 and x_2 so as to maximize $Z = 3x_1 + 5x_2$, subject to the restrictions imposed on their values by the limited production capacities available at the three plants. Figure A1 indicates that each batch of product 1 produced per week uses 1 hour of production time per week in Plant 1, whereas only 4 hours per week is available. This restriction is expressed mathematically by the inequality $x_1 \leq 4$. Similarly, Plant 2 imposes the restriction that $2x_2 \leq 12$. The number of hours of production time used per week in Plant 3 by choosing x_1 and x_2 as the new products' production rates would be $3x_1 + 2x_2$. Therefore, the mathematical statement of the Plant 3 restriction is $3x_1 + 2x_2 \leq 18$. Finally, since production rates cannot be negative, it is necessary to restrict the decision variables to be non-negative: $x_1 \geq 0$ and $x_2 \geq 0$.

To summarize, in the mathematical language of linear programming, the problem is to choose values of x_1 and x_2 so as to

Prototype Example	General Problem
Production Capacities of plants	Resources
3 plants	*m* resources
Production of products	Activities
2 products	*n* activities
Production rate of product j, x_j	Level of activity j, x_j
Profit Z	Overall measure of performance Z

Figure A.2: Common Terminology for Linear Programming.

$$\text{Maximize } Z = 3x_1 + 5x_2$$

subject to the restrictions

$$\begin{aligned} x_1 &\leq 4 \\ 2x_2 &\leq 12 \\ 3x_1 + 2x_2 &\leq 18 \end{aligned}$$

and

$$\begin{aligned} x_1 &\leq 0 \\ x_2 &\leq 0 \end{aligned}$$

A.1.2 Formulation of the Linear Programming Model

The previous example is intended to illustrate a typical linear programming problem (in a small scale). However, linear programming is too versatile to be completely characterized by a single example. Here we will briefly discuss the general characteristics of linear programming problems, including the various legitimate forms of the mathematical model for linear programming.

We will begin with some terminology and notation. The first column of Figure A2 summarizes the components of the glass factory problem. The second column then introduces more general terms for these same components that will fit many linear programming problems. The key terms are *resources* and *activities*, where m denotes the number of different kinds of resources that can be used and n denotes the number of activities being considered. Some typical resources are money and particular kinds of machines, equipment, vehicles and personnel. Examples of activities include investing in particular projects, advertising in particular media and shipping goods from a particular source to a particular destination. In any application of linear programming, all the activities may be of one general kind (such as any one of these three examples), and then the individual activities would be particular alternatives within this general category.

The most common type of application of linear programming involves allocating resources to activities. The amount available of each resource is limited, so a careful allocation of resources to activities must be made. Determining this allocation involves choosing the *levels* of the activities that achieve the best possible value of the *overall measure of performance.*

Certain symbols are commonly used to denote the various components of a linear programming model. These symbols are listed below, along with their interpretation for the general problem of allocating resources to activities.

Z=value of overall measure of performance. x_j=level of activity j (for $j = 1, 2, ..., n$). c_j=increase in Z that would result from each unit increase in level of activity j. b_i=amount of resource i that is available for allocation to activities (for $i = 1, 2, ..., m$). a_{ij}=amount of resource i consumed by each unit of activity j.

The model poses the problem in terms of making decisions about the levels of the activities, so $x_1, x_2, ..., x_n$ are called the **decision variables**. As summarized in Figure A3, the values of c_j, b_i and a_{ij} (for $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$) are the *input constants* for the model. The c_j, b_i and a_{ij} are also referred to as the **parameters** of the model.

A.1.3 A Standard Form of the Model

Proceeding as for the glass factory problem, we can now reformulate the mathematical model for this general problem of allocating resources to activities. In particular, this model is to select the values for $x_1, x_2, ..., x_n$ so as to

A.1.4 Assumptions of Linear Programming

All of the assumptions of linear programming actually are implicit in the model formulation given earlier. However, it is good to highlight theses assumptions so that it can be more easily evaluated how well linear programming applies to any given problem.

Proportionality

Proportionality is an assumption about both the objective function and the functional constraints, as summarized below:

The contribution of each activity to the *value of the objective function* Z is *proportional* to the *level of the activity* x_j, as represented by the c_jx_j term in the objective function. Similarly, the contribution of each activity to the *left-hand side of each functional constraint* is *proportional* to the *level of the activity* x_j,

Resource	Resource Usage per Unit of Activity				Amount of Resource Available
	Activity				
	1	2	...	n	
1	a_{11}	a_{12}	…	a_{1n}	b_1
2	a_{21}	a_{22}	…	a_{2n}	b_2
…	…	…	…	…	…
m	a_{m1}	a_{m2}	…	a_{mn}	b_n
Contribution to Z per unit of activity	c_1	c_2	…	c_n	

Figure A.3: Data needed for a Linear Programming Model Involving the Allocation of Resources to Activities.

as represented by the $a_{ij}x_j$ term in the constraint. Consequently, this assumption rules out any exponent other than 1 for any variable in any term of any function (whether the objective function or the function on the left-hand side of a functional constraint) in a linear programming model.

Additivity

Although the proportionality assumption rules out exponents other than one, it does not prohibit *cross-product terms* (terms involving the product of two or more variables). The additivity assumption does rule out this latter possibility, as summarized below:

Every function in a linear programming model (whether the objective function or the function on the left-hand side of a functional constraint) is the *sum* of the *individual contributions* of the respective activities.

Divisibility

Our next assumption concerns the values allowed for the decision variables:

Decision variables in a linear programming model are allowed to have *any* values, including *non-integer* values, that satisfy the functional and non-negativity constraints. Thus, these variables are *not* restricted to just integer values. Since each division variable represents the level of some activity, it is being assumed that the activities can be run at *fractional levels*.

Certainty

Our last assumption concerns the *parameters* of the model, namely, the coefficients in the objective function c_j, the coefficients in the functional constraints a_{ij}, and the right-hand sides of the functional constraints b_i.

The value assigned to each parameter of a linear programming model is assumed to be a *known constant*.

A.1.5 The Assumptions in Perspective

A mathematical model is intended to be only an idealized representation of the real problem. Approximations and simplifying assumptions generally are required in order for the model to be tractable. Adding too much detail and precision can make the model too unwieldy for useful analysis of the problem. All that is really needed is that there be a reasonably high correlation between the prediction of the model and what would actually happen in the real problem.

This advice is certainly applicable to linear programming. It is very common in real applications of linear programming that almost *none* of the four assumptions hold completely. Except perhaps for the *divisibility assumption*, minor disparities are to be expected. This is especially true for the *certainty assumption*, so sensitivity analysis normally is a must to compensate for the violation of this assumption.

It is important, however, to examine the four assumptions for the problem under study and to analyze just how large the disparities are. If any of the assumptions are violated in a major way , then a number of useful alternative models are available (integer programming (IP), mixed-integer programming (MIP), non-linear programming (NLP)). A disadvantage of these other models is that the algorithms available for solving them are not nearly as powerful as those for linear programming, but this gap has been closing in some cases. For some applications, the powerful linear programming approach is used for the initial analysis , and then a more complicated model is used to refine this analysis.

A.2 Integer Programming

There have been numerous applications of integer programming that involve a direct extension of linear programming where the divisibility assumption must be dropped. However, another area of application may be of even greater importance, namely, problems involving a number of interrelated "yes-or-no decisions". In such decisions, the only two possible choices are *yes* and *no*. For example, should we undertake a particular fixed project? Should we make a particular fixed investment? Should we locate a facility in a particular site?

With just two choices, we can represent such decisions by decision variables that are restricted to just two values, say 0 and 1. Thus the jth yes-or-no decision would be represented by, say x_j, such that x_j=1 if decision j is yes and x_j=0 if decision j is no.

Such variables are called **binary variables** (or 0-1 variables). Consequently, IP problems that contain only binary variables sometimes are called **binary integer programming (BIP)** problems (or 0-1 integer programming problems). In the next subsection we will examine the usage of binary variables in the reformulation of IP problems.

A.2.1 Formulation Possibilities with Binary Variables

Binary variables sometimes enable us to take a problem whose natural formulation is intractable and *reformulate* it as a pure or mixed IP problem.

This kind of situation arises when the original formulation of the problem fits either an IP or a linear programming format *except* for minor disparities involving combinatorial relationships in the model. By expressing these combinatorial relationships in terms of questions that must be answered yes or no, *auxiliary* binary variables can be introduced to the model to represent these yes-or-no decisions. Introducing these variables reduces the problem to an MIP problem (or a *pure* IP problem if all the original variables are also required to have integer values.

Some cases that can be handled by this approach are discussed next, where the x_j denote the *original* variables of the problem (they may be either continuous or integer variables) and the y_i denote the *auxiliary* binary variables that are introduced for the reformulation.

Either-Or Constraints

Consider the important case where a choice can be made between two constraints, so that *only one* must hold (whereas the other one can hold but is not required to do so). For example, there may be a choice as to which of two resources to use for a certain purpose, so that it is necessary for only one of the two resource availability constraints to hold mathematically. To illustrate the approach to such situations, suppose that one of the requirements in the overall problem is that

$$\text{Either } 3x_1 + 2x_2 \leq 18$$

$$\text{or } x_1 + 4x_2 \leq 16$$

i.e., at least one of these two inequalities must hold but not necessarily both. This requirement must be reformulated to fit it into the linear programming format where *all* specified constraints must hold. Let M be a very large positive number. Then this requirement can be rewritten as

$$3x_1 + 2x_2 \leq 18$$

Either

$$x_1 + 4x_2 \leq 16 + M$$

$$3x_1 + 2x_2 \leq 18 + M$$

or

$$x_1 + 4x_2 \leq 16$$

The key is that adding M to the right-hand side of such constraints has the effect of eliminating them, because they would be satisfied automatically by any solutions that satisfy the other constraints of the problem. (This formulation assumes that the set of feasible solutions for the overall problem is a bounded set and that M is large enough that it will not eliminate any feasible solutions. This formulation is equivalent to the set of constraints:

$$
\begin{aligned}
3x_1 + 2x_2 &\leq 18 + My \\
x_1 + 4x_2 &\leq 16 + M(1 - y)
\end{aligned}
$$

Because the *auxiliary variable* y must be either 0 or 1, this formulation guarantees that one of the original constraints must hold while the other is, in effect, eliminated. This new set of constraints would then be appended to the other constraints in the overall model to give a pure or mixed IP problem (depending upon whether the x_j are integer or continuous variables).

This approach is related directly to our earlier discussion about expressing combinatorial relationships in terms of questions that must be answered yes or no. The combinatorial relationship involved concerns the combination of the *other* constraints of the model with the *first* of the two *alternative* constraints and then with the *second*. Which of these combinations of constraints is *better* (in terms of the value of the objective function that then can be achieved)? To rephrase the question in yes-or-no terms, we ask two complementary questions:

1. Should $x_1 + 4x_2 \leq 16$ be selected as the constraint that must hold?
2. Should $3x_1 + 2x_2 \leq 18$ be selected as the constraint that must hold?

Because exactly one of these questions is to be answered affirmatively, we let the binary terms y and $1 - y$, respectively, represent these yes-or-no decisions. Thus, $y = 1$ if the answer is yes to the first question (and no to the second), whereas $1-y = 1$ (that is $y = 0$) if the answer is yes to the second question (and no to the first). Since $y + 1 - y = 1$ (one yes) automatically, there is no need to add another constraint to force these two decisions to be mutually exclusive. (If separate binary variables y_1 and y_2 had been used instead to represent these yes-or-no decisions, then an additional constraint $y_1 + y_2 = 1$ would have been needed to make them mutually exclusive).

A formal presentation of this approach is given next for a more general case.

K out of N constraints must hold

Consider the case where the overall model includes a set of N possible constraints such that only some K of these constraints *must* hold. (Assume that $K < N$). Part of the optimization process is to choose the *combination* of K constraints that permits the objective function to reach its best possible value. The $N - K$ constraints *not* chosen are, in effect, eliminated from the problem, although feasible solutions might coincidentally still satisfy some of them.

This case is a direct generalization of the preceding case, which had $K = 1$ and $N = 2$. Denote the possible constraints by:

$$
\begin{aligned}
f_1(x_1, x_2, ..., x_n) &\leq d_1 \\
f_2(x_1, x_2, ..., x_n) &\leq d_2 \\
&. \\
&. \\
&. \\
f_n(x_1, x_2, ..., x_n) &\leq d_n
\end{aligned}
$$

Then, applying the same logic as for the preceding case, we find that an equivalent formulation of the requirement that some K of these constraints *must* hold is:

$$
\begin{aligned}
f_1(x_1, x_2, ..., x_n) &\leq d_1 + My_1 \\
f_2(x_1, x_2, ..., x_n) &\leq d_2 + My_2 \\
&. \\
&. \\
&. \\
f_n(x_1, x_2, ..., x_n) &\leq d_n + My_n \\
\sum_{i=1}^{N} y_i &= N - K
\end{aligned}
$$

and

$$y_i \text{ is binary, for } i = 1, 2, ..., N,$$

where M is an extremely large positive number. For each binary variable $y_i (i = 1, 2, ..., N)$, note that $y_i = 0$ makes $My_i = 0$, which reduces the new constraint i to the original constraint i. On the other hand, $y_1 = 1$ makes $(d_i + My_i)$ so large that (again assuming a bounded feasible region) the new constraint i is automatically satisfied by any solution that satisfies the other new constraints, which has the effect of eliminating the original constraint i. Therefore, because the constraints on the y_i guarantee that K of these variables will equal 0 and those remaining will equal 1, K of the original constraints will be unchanged and the other $(N-K)$ original constraints will, in effect, be eliminated. The choice of *which* K constraints should be retained is made by applying the appropriate algorithm to the overall problem so it finds an optimal solution for *all* the variables simultaneously.

Functions with N possible values

Consider a situation where a given function is required to take on any one of N given values. Denote this requirement by

$$f(x_1, x_2, ..., x_n) = d_1 \text{ or } d_2,..., \text{ or } d_n.$$

One special case is where this function is

$$f(x_1, x_2, ..., x_n) = \textstyle\sum_{j=1}^{n} a_j x_j,$$

as on the left-hand side of a linear programming constraint. Another special case is where $f(x_1, x_2, ..., x_n) = x_j$ for a given value of j, so the requirement becomes that x_j must take on any one of N given values.

The equivalent IP formulation of this requirement is the following:

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{N} d_i y_i$$

$$\sum_{i=1}^{N} y_i = 1$$

and

$$y_i \text{ is binary, for } i = 1, 2, ..., N,$$

so this new set of constraints would replace this requirement in the statement of the overall problem. This set of constraints provides an *equivalent* formulation because exactly one y_i must equal 1 and the others must equal 0, so exactly one d_i is being chosen as the value of the function. In this case, there are N yes-or-no questions being asked, namely, should d_i be the value chosen ($i = 1, 2, ..., N$)? Because the y_i respectively represent these *yes-or-no decisions*, the second constraint makes them *mutually exclusive alternatives*.

To illustrate how this case can arise, reconsider the glass company problem presented earlier. Eighteen hours of production time per week in Plant 3 currently is unused and available for the two new products *or* for certain future products that will be ready for production soon. In order to leave any remaining capacity in usable blocks for these future products, management now wants to impose the restriction that the production time used by the two current new products be 6 *or* 12 *or* 18 hours per week. Thus the third constraint of the original model ($3x_1 + 2x_2 \leq 18$) now becomes

$$3x_1 + 2x_2 = 6 \text{ or } 12 \text{ or } 18.$$

In the preceding notation, $N = 3$ with $d_1 = 6, d_2 = 12$ and $d_3 = 18$. Consequently, management's new requirement should be formulated as follows:

$$\begin{aligned} 3x_1 + 2x_2 &= 6y_1 + 12y_2 + 18y_3 \\ y_1 + y_2 + y_3 &= 1 \end{aligned}$$

and

$$y_1, y_2, y_3 \text{ are binary..}$$

The overall model for this new version of the problem then consists of the original model plus this new set of constraints that replaces the original third constraint. This replacement yield a very tractable MIP formulation.

The Fixed-Charge Problem

It is quite common to incur a fixed charge or setup cost when one is undertaking an activity. For example, such a charge occurs when a production run to produce a batch of a particular product is undertaken and the required production facilities must be set up to initiate the run. In such cases, the total cost of the activity is the sum of a variable cost related to the level of the activity and the setup cost required to initiate the activity. Frequently the variable cost will be at least roughly proportional to the level of the activity. If this is the case, the *total cost* of the activity (say, activity j) can be represented by a function of the form:

$$f_j(x_{ij}) = k_j + c_j x_j \text{ if } x_j > 0$$

or

$$f_j(x_{ij}) = 0 \text{ if } x_j = 0$$

where x_j denotes the level of activity j $(x_j \geq 0)$, k_j denotes the setup cost and c_j denotes the cost for each incremental unit. Were it not for the setup cost k_j, this cost structure would suggest the possibility of a *linear programming* formulation to determine the optimal levels of the competing activities. Fortunately, even with the k_j, MIP can still be used.

To formulate the overall model, suppose that there are n activities, each with the preceding cost structure (with $k_j \geq 0$ in every case and $k_j > 0$ for some $j = 1, 2, ..., n$), and that the problem is to

$$\text{Minimize } Z = f_1(x_1) + f_2(x_2) + ... + f_n(x_n),$$

subject to

given linear programming constraints

To convert this problem to an MIP format, we begin by posing n questions that must be answered yes or no; namely, for each $j = 1, 2, ..., n$, should activity j be undertaken $(x_j > 0)$? Each of these *yes-or-no decisions* is then represented by an auxiliary *binary variable* y_j, so that

$$Z = \sum_{j=1}^{n} (c_j x_j + k_j y_j),$$

where

$$y_j = 1 \text{ if } x_j > 0 \text{ or } y_j = 0 \text{ if } x_j = 0$$

Let M be an extremely large positive number that exceeds the maximum feasible value of any x_j $(j = 1, 2, ..., n)$. Then the constraints

$$x_j \leq My_j \text{ for } j = 1, 2, ..., n$$

will ensure that $y_j = 1$ rather than 0 whenever $x_j = 0$. The one difficulty remaining is that these constraints leave y_j free to be either 0 or 1 when $x_j = 0$. Fortunately, this difficulty is automatically resolved because of the nature of the objective function. The case where $k_j = 0$ can be ignored because y_j can then be deleted from the formulation. So we consider the only other case, namely, where $k_j > 0$. When $x_j = 0$, so that the constraints permit a choice between $y_j = 0$ and $y_j = 1$, $y_j = 0$ must yield a smaller value of Z than $y_j = 1$. Therefore, because the objective is to minimize Z, an algorithm yielding an optimal solution would always choose $y_j = 0$ when $x_j = 0$.

To summarize, the MIP formulation of the fixed-charge problem is:

$$Z = \sum_{j=1}^{n}(c_j x_j + k_j y_j),$$

subject to

the original constraints, plus

$$x_j - My_j \leq 0$$

and

$$y_j \text{ is binary, for } j = 1, 2, ..., n.$$

If the x_j also had been restricted to be integer, then this would be a *pure* IP problem.

Binary Representation of General Integer Values

Suppose that you have a pure IP problem where most of the variables are *binary* variables, but the presence of a few *general* integer variables prevents you from solving the problem by one of the very efficient BIP algorithms now available. A nice way to circumvent this difficulty is to use the *binary representation* for each of these general integer variables. Specifically, if the bounds on an integer variable x are

$$0 \leq x \leq u$$

and if N is defined as the integer such that

$$2^N \leq u < 2^{N+1}$$

then the **binary representation** of x is

$$x = \sum_{i=0}^{N} 2^i y_i,$$

where the y_i variables are (auxiliary) binary variables. Substituting this binary representation for each of the general integer variables (with a different set of auxiliary binary variables for each) thereby reduces the entire problem to a BIP model.

For example, suppose that an IP problem has just two general integer variables x_1 and x_2 along with many binary variables. Also suppose that the problem has non-negativity constraints for both x_1 and x_2 and that the functional constraints include

$$\begin{aligned} x_1 &\leq 5 \\ 2x_1 + 3x_2 &\leq 30 \end{aligned}$$

These constraints imply that $u = 5$ for x_1 and $u = 10$ for x_2, so the above definition of N gives $N = 2$ for x_1 (since $2^2 \leq 5 < 2^3$) and $N = 3$ for x_2 (since $2^3 \leq 10 < 2^4$). Therefore, the binary representations of these variables are

$$\begin{aligned} x_1 &= y_0 + 2y_1 + 4y_2 \\ x_2 &= y_3 + 2y_4 + 4y_5 + 8y_6. \end{aligned}$$

After we substitute these expressions for the respective variables throughout all the functional constraints and the objective function, the two functional constraints noted above become

$$\begin{aligned} y_0 + 2y_1 + 4y_2 &\leq 5 \\ 2y_0 + 4y_1 + 8y_2 + 3y_3 + 6y_4 + 12y_5 + 24y_6 &\leq 30. \end{aligned}$$

Observe that each feasible value of x_1 corresponds to one of the feasible values of the vector (y_0, y_1, y_2), and similarly for x_2 and (y_3, y_4, y_5, y_6). For example, $x_1 = 3$ corresponds to (y_0, y_1, y_2)=$(1, 1, 0)$, and $x_2 = 5$ corresponds to (y_3, y_4, y_5, y_6)=$(1, 0, 1, 0)$.

For an IP problem where *all* the variables are (bounded) general integer variables, it is possible to use this same technique to reduce this problem to a

BIP model. However, this is not advisable for most cases because the explosion in the number of variables involved. Applying a good IP algorithm to the original IP model generally should be more efficient than applying a good BIP algorithm to the much larger BIP model.

In general terms, for *all* the formulation possibilities with auxiliary binary variables discussed here, we need to strike the same note of caution. This approach sometimes requires adding a relatively large number of such variables, which can make the model *computationally infeasible*.

A.3 Non-Linear Programming

A constrained non-linear programming problem deals with the search for a maximum (or minimum) of a function $f(x)$ of n variables $x = (x_1, x_2, ..., x_n)$ subject to a set of inequality constraints $g_j(x) \leq 0, (g_j(x) = 0, j = 1, 2, ..., p)$, and is denoted as

$$\text{Maximize } f(x)$$
$$\text{subject to}$$
$$g_j(x) \leq b_j, j = 1, 2, ..., m$$

If any of the functions $f(x), h(x), g(x)$ is non-linear, then the above formulation is called a constrained non-linear programming problem. The functions $f(x), h(x), g(x)$ can take any form of non-linearity, and it is assumed that they satisfy continuity and differentiability requirements.

No algorithm that will solve *every* specific problem fitting this format is available. However, substantial progress has been made for some important special cases of this problem by making various assumptions about these functions, and research is continuing very actively.

Closely related to the idea of non-linear programming are the notions of convex sets as well as convex and concave functions. We will briefly define these notions below:

Convex set definition. A set $S \epsilon \Re^n$ is said to be convex if the closed line segment joining any two points x_1 and x_2 of the set S, that is, $(1 - \lambda)x_1 + \lambda x_2$, belongs to the set S for each λ such that $0 \leq \lambda \leq 1$.

Convex function definition. Let S be a convex subset of $\Re^n$, and $f(x)$ be a real valued function defined on S. The function $f(x)$ is said to be convex if for any $x_1, x_2 \epsilon S$, and $0 \leq \lambda \leq 1$, we have $f[(1-\lambda)x_1+\lambda x_2] \leq (1-\lambda)f(x_1)+\lambda f(x_2)$. This inequality is called *Jensen's inequality* after the Danish mathematician who first introduced it.

Concave function definition. Let S be a convex subset of $\Re^n$, and $f(x)$ be a real valued function defined on S. The function $f(x)$ is said to be concave if for any $x_1, x_2 \epsilon S$, and $0 \leq \lambda \leq 1$, we have $f[(1-\lambda)x_1+\lambda x_2] \geq (1-\lambda)f(x_1)+\lambda f(x_2)$.

In simpler terms, a convex function is always "curving upward" (or not at all) and a concave function is always "curving downward" (or not at all).

If a non-linear programming problem has no constraints, the objective function being *concave* guarantees that a local maximum is a *global maximum*. (Similarly, the objective function being *convex* ensures that a local minimum is a *global minimum*. If there are constraints, then one more condition will provide this guarantee, namely, that the *feasible region* is a **convex set**. In essence, a convex set is simply a set of points such that, for each pair of points in the collection, the entire line segment joining these two points is also in the collection.

In general, the feasible region for a non-linear programming problem is a convex set whenever all the $g_j(x)$ [for the constraints $g_j(x) \leq b_j$] are convex. The subject of non-linear programming is a very large one and is constantly updated and reviewed. For more information, the reader is encouraged to recur to two very good sources on the matter [16-17].

Appendix B

Historical Notes and Technical Terms

B.1 Historical Notes

• **The Air Commerce Act of 1926**

As airmail began crossing the U.S.A. successfully in the mid-1920s, railroad owners started complaining that this government-sponsored enterprise was cutting into their business. They found a friendly ear in Congressman Clyde Kelly of Pennsylvania, chairman of the House Post Office Committee, who largely represented railroad interests. On February 2, 1925, he sponsored H.R. 7064: the Contract Air Mail Bill, which, when enacted, became the Air Mail Act of 1925 or the Kelly Act. The act authorized the postmaster general to contract for domestic airmail service with commercial air carriers. It also set airmail rates and the level of cash subsidies to be paid to companies that carried the mail. As Kelly explained: The act "permits the expansion of the air mail service without burden upon the taxpayers." By transferring airmail operations to private companies, the government effectively would help create the commercial aviation industry.

The first sign of commercial interest came on April 3, 1925, when the automaker Henry Ford opened a private air freight service between Detroit and Chicago. Soon after, when bids were solicited for the first contract routes, there was no shortage of interested companies submitting bids stating how much they would charge the government.

Eighty percent of the stamp money received by the Post Office was to be paid to the airmail carriers. The quantity of stamps needed depended on the weight of the mail and also on how many of the three zones the mail had to cross. (The country had been divided into three air zones on July 1, 1924.) Companies saw that they would make more money if they carried smaller but heavier pieces of mail. Also, since they would receive the same amount of money no matter how many miles they flew within a zone, they preferred to fly shorter distances within a single zone and save some operating costs.

Harry S. New, postmaster general under President Calvin Coolidge, wanted the airmail carriers to expand their routes and to buy larger airplanes to carry more passengers. He awarded contracts only to the largest companies that bought the largest aircraft, which could accommodate more passengers as well as the mail. New realized that if the airlines sold more passenger tickets, which then numbered only a few hundred each year, they could carry less mail and still make a profit. The companies would receive their income from passengers rather than from the Post Office as payment for carrying the mail. New awarded eight airmail routes to seven airmail carriers, beginning in October 1925. One carrier, Ford Air Transport, won two of the routes and was the first to fly airmail under contract, starting on February 15, 1926.

The postmaster general noticed that airmail operators continued to fly only the shortest routes within their zone, since they would receive no more stamp money for flying longer distances within that zone. To remedy this, on June 3, 1926, the Kelly Act was amended to instead pay 3 dollars per pound of mail (454 grams) for the first 1000 miles (1600 kilometers) and 30 cents per pound for each additional 100 miles (160 kilometers).

In May 1926, Congress passed the Air Commerce Act, which gave the government responsibility for fostering air commerce, establishing airways and aids to air navigation, and making and enforcing safety rules. Under this act, the government supplied money for air navigation facilities so that the routes would become safer to fly, day and night. Management of the route system moved to the new Aeronautics Branch of the Department of Commerce, which was established in August under the leadership of William MacCracken.

By the early part of 1926, contract airmail carriers flew most of the airmail, but government airmail pilots in government airplanes still flew the transcontinental route connecting San Francisco, Omaha, Chicago and New York. This transcontinental line was divided into two segments in 1927. Boeing began contract service on the western sector, between Chicago and San Francisco, on July 1, 1927. National Air Transport took over the eastern sector, between New York and Chicago, on September 1, 1927. Now, all airmail operations had shifted to private companies flying with their own pilots and aircraft.

Other changes were made too. Most of the airfields on the route system had been paid for and managed by the Federal Government through the Post Office Department. They were now handed over to the local government near each airfield to pay for and manage, except for the important mail centers of Omaha and San Francisco and possibly Chicago. In July 1927, the Department of Commerce took over the construction and maintenance of the still-incomplete transcontinental lighted airway. In addition to hundreds of light beacons, the airway's facilities included 95 emergency landing fields and 17 radio stations that had been built since 1921 to provide pilots with weather information.

Improved aircraft technology helped increase the volume of mail and freight that could be carried. Some airplanes could carry passengers, baggage, and airmail. Air-cooled engines replaced water-cooled engines. Some of the new engines generated more than 450 horsepower (336 kilowatts) and helped airlines improve on the average speed of 110 miles per hour (176 kilometers per hour).

In 1928, the Post Office gave operators that had been in business at least two years a 10-year contract that excluded any competitors. The mail carriers still favored the shorter routes within their zones but to meet government requirements, airlines began to merge and create longer routes to more cities.

Pilot groups were founded as well as airline companies. In 1928, the National Air Pilots Association (NAPA) was formed, and by 1931, the Air Line Pilots Association (ALPA). By the spring of 1929, there were 61 U.S. passenger lines, and 47 airmail lines. Airmail volume in 1926 had been 810.555 pounds (402.525 kilograms); by 1929, airmail volume had grown to 7.772.014 pounds (3.532.733 kilograms).

Though the aviation industry made money, the Post Office supported growth of the system and lost more money each year. In 1929, airmail subsidies reached 11.618.000 dollars, but airmail revenues were only 5.273.000 dollars. To keep airmail stamps affordable, the Post Office limited the stamp price to five cents per ounce and made up the difference with tax money.

Airmail carriers learned to use the subsidies to make money regardless of the true public demand for airmail. They sometimes sent postcards to themselves using registered mail, which required a heavy, secure lock. The lock added weight and, therefore, the government had to pay more. Despite such abuses, the postal subsidies encouraged aircraft designers to design aircraft that were more reliable, could fly longer distances, and were less expensive to fly.

Herbert Hoover was elected President in 1928. He would appoint a new postmaster general, Walter Folger Brown, a man who wanted to create a stable and efficient air transport system that served both passengers and the mail. Brown began work on March 6, 1929 and rapidly began to shake up the industry.

- **RTCA, Inc.**

RTCA, Inc. is a private, not-for-profit corporation that develops consensus-based recommendations regarding communications, navigation, surveillance, and air traffic management (CNS/ATM) system issues. RTCA functions as a Federal Advisory Committee. Its recommendations are used by the Federal Aviation Administration (FAA) as the basis for policy, program, and regulatory decisions and by the private sector as the basis for development, investment and other business decisions.

Organized in 1935 as the Radio Technical Commission for Aeronautics, RTCA today includes roughly 250 government, industry and academic organizations from the United States and around the world. Member organizations represent all facets of the aviation community, including government organizations, airlines, airspace user and airport associations, labor unions, plus aviation service and equipment suppliers. A sampling of their domestic membership includes the Federal Aviation Administration, Air Line Pilots Association, Air Transport Association of America, Aircraft Owners and Pilots Association, ARINC Incorporated, Avwrite, The Boeing Company, Department of Commerce, Department of Defense, GARMIN International, Honeywell International, Inc., The Johns Hopkins University, Lockheed Martin, MIT Lincoln Laboratory, MITRE/CAASD, NASA, National Business Aviation Association, and Raytheon.

Because RTCA interests are international in scope, many non-U.S. government and business organizations also belong to RTCA. They currently are supported by approximately 60 International Associates such as Airservices Australia, Airways Corporation of New Zealand, the Chinese Aeronautical Radio Electronics Research Institute (CARERI), EUROCONTROL, NAV Canada, Pilatus Aircraft Limited, Smiths Industries, Society of Japanese Aerospace Companies, Thales Avionics Limited, the United Kingdom Civil Aviation Authority and many more.

B.2 Technical Terms

- **The U.S. NAS**

The National Airspace System (NAS) is the term used to represent the overall environment in which aircraft operate. This includes the aircraft itself, the pilots, the facilities, the tower controllers, the terminal area controllers, the enroute controllers, and the oceanic controllers. It includes the airports, the maintenance personnel and the airline dispatchers. All of this, with their computers, communications equipment, satellite navigation aids, and radars, are part of the NAS.

It covers every aspect of aviation in the United States, beginning with the aircraft itself. The Federal Aviation Administration (FAA) is charged with certifying the aircraft for safe operations. The FAA also certifies maintenance and repair operations, their practices and personnel, and even the certification of spare parts.

Airports are perhaps the most fundamental component of the NAS. No matter the size or complexity of an airport, FAA sets standards for construction and operation of airport facilities.

The NAS includes some 36.000 pieces of equipment operating in hundreds of locations throughout the United States. These can range from simple navigation beacons to very modern Air Route Traffic Control Centers that handle the enroute traffic. The mission of this highly integrated system is to support all phases of flight for aircraft in the United States, from initial flight planning to successful take off, enroute operations, and landing. The system provides communications, navigation, surveillance, display, flight planning, and weather data to controllers, traffic managers and pilots.

The NAS integrates a number of control facilities, radars, computers, and communications systems. Operated by the controller workforce, a staff of some 15.000, they control the aircraft in the system and provide critical data through every stage of their operations. Included in this are the towers themselves, the 171 Terminal Radar Approach Control Facilities throughout the United States, the Air Route Control Centers that control aircraft in the En Route environment and the three Oceanic Control Centers. All of these control centers are linked and managed through an Air Traffic Control System Command Center in Herndon, Virginia. All of these centers are operated and managed by FAA personnel.

- **Airways**

In most cases, aircraft, especially airliners, cannot simply fly from one point directly to another. Rather, they must follow designated airways. Airways are an invisible three-dimensional network of roads that zigzag within controlled airspace. Sometimes called corridors, most are nine miles (14 kilometers) wide. Within these corridors, aircraft fly separated by at least 1000 feet (305 meters) of distance above and below them when they are at altitudes up to 29.000 feet (8.839 meters). Above 29.000 feet (8.839 meters), aircraft are separated by at least 2.000 feet (610 meters) because controllers have more difficulty tracking aircraft at this altitude and flying at the speeds for that altitude.

Each airway carries its own name, required airspeed, radio and cockpit instrument procedures, operating altitudes, and rules for entering and leaving the airway. The low-altitude Victor Airways run from 700 feet (213 meters) above ground level to 18.000 feet (5.486 meters), while Jet Routes run from 18.000 feet to 35.000 feet (5.486 meters to 10.668 meters). Within these broad groups, all navigation aids such as radio transmission stations, visual and satellite checkpoints, and the responsible control center have names and unique abbreviations. This complex language is printed on pilot charts and in thick directories. Low-altitude airways are shown on the sectional aeronautical chart.

At points where these invisible roadways intersect, radio signals from ground stations mesh to form an electronic picture on cockpit instruments that looks like a road intersection. The Federal Aviation Regulations and air traffic controls set rules for how to cross airways and at what altitude, what intersections to use, and at what angle and speed of flight to enter and leave them.

Airways are for civilian aircraft and airliners. A separate system of airways exists for military aircraft that protects civilian aircraft from the very high-speed military operations and which protects military or government areas from unauthorized flights over their land.

- **Radar**

Nothing revolutionized air traffic control more than radar, a system that the British initially developed for air defense in the era preceding World War II. After the war, the U.S. Civil Aeronautics Administration began applying this technology to the problem of keeping civilian flights safely separated.

The Radar (Radio Detection And Ranging) is in essence a remote detection system that is used to locate and identify objects. It consists of an electronic system in which radio waves are bounced off an aircraft or other object in order to detect its presence and locate its position.

A radar system uses four basic components: a transmitter, an antenna, a receiver, and a display. The transmitter generates radio signals. The antenna transmits these signals as electromagnetic radiation into the airspace. When a target, such as an aircraft, enters the airspace, it scatters some of these radio waves, which reach the receiving antenna. An electronic amplifier amplifies these returned signals and displays them on a cathode ray tube (CRT) display where a radar operator can examine them. The location of the object being detected is determined by measuring the time it takes for the radio wave to travel from the transmitter to the object and back to the receiver.

There are many types of radar: The most common is pulse radar where the radio waves are emitted in discrete pulses. Moving-target indication radar is a form of pulse radar can detect the location of moving targets. Airborne moving-target indication radar makes it possible to detect moving targets when the radar unit itself is moving, such as when a moving aircraft detects another moving aircraft. Various pulse radars use pulses that operate on different frequencies for different purposes. There are also various imaging radars that are used to produce images. The most common of these is synthetic aperture radar (SAR), which is primarily used to map the Earth's surface.

Tracking radar can continuously follow a single target to determine its path and predict its future position. Automatic detection and tracking radar consists of targets that show up on a radar screen as tracks rather than as discrete blips. 3-D radar locates it target in terms of a reference point and the horizon. Phased-array radar can track many targets at the same time. Continuous-wave radar transmits and receives signals at the same time. It can distinguish the weak returning signal from the strong transmitted signal. Aircraft use a particular type of continuous-wave radar, called frequency-modulated continuous-wave radar, to determine their height above ground.

• **TRACONS**

There are 185 Terminal Radar Approach Control (TRACON) facilities in the U.S. National Airspace system. Tracons are either located within the control tower or in a separate building located on or near the major airport it serves. Using radar equipment in very dark rooms, these controllers typically work an area of airspace with a 40 mile radius and up to 17,000 feet of altitude, determining the sequence of arriving aircraft and directing departures towards their destinations. This airspace is usually configured to provide service to one primary airport, but may include other airports that are within the radar service area. Aircraft within this area of radar coverage are provided vectors to airports and around other aircraft, terrain and weather.

Tracons are divided up into smaller sections of airspace called sectors. Larger tracons have around eight sectors, while combined tracons, housing several facilities, may have as many as 30 or more sectors.

Tracon operational positions are usually one of three types:

• **Arrival** – Arrival controllers determine the sequence of arrivals into an airport. Larger tracons will further differentiate the arrival position by determining the specific final approach or even downwind pattern the controller will work.

• **Departure** – Departure controllers climb aircraft to their initial altitudes and establish them on a route or heading towards the appropriate departure corridor.

• **Transition** – Transition controllers work all tracon areas that are not focused at the ends of a runway. Primary responsibilities include initial sequencing of arrivals, establishing departures on their route of flight and handling aircraft simply over flying the airport to land somewhere else.

• **Global Positioning System (GPS)**

Originally established by the Department of Defense of The U.S.A., the global positioning system (GPS) is a satellite navigation system. It was originally intended for military operations, but in the 1980s, the U.S. government made the system available for civilian use. GPS works in any weather conditions, anywhere in the world, 24 hours a day [18].

It consists of approximately 24 satellites in orbit around the Earth, several ground tracking stations, and a receiver in the aircraft, other vehicle, or held by an individual. In reality, there are 27 satellites, 24 in operation and three extras in case one fails. The ground control sites watch where the satellites are in orbit and continually correct their reported location and time-of-day signals. This is done so that when the satellite communicates with a receiver, it gives the best possible position it can to help navigate.

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's exact location. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position and display it on the unit's electronic map.

A GPS receiver must be locked on to the signal of at least three satellites to calculate a 2D position (latitude and longitude) and track movement. With four or more satellites in view, the receiver can determine the user's 3D position (latitude, longitude and altitude). Once the user's position has been determined, the GPS unit can calculate other information, such as speed, bearing, track, trip distance, distance to destination, sunrise and sunset time and more.

The 24 satellites that make up the GPS space segment are orbiting the earth about 12.000 miles above us. They are constantly moving, making two complete orbits in less than 24 hours. These satellites are travelling at speeds of roughly 7.000 miles an hour.

GPS satellites are powered by solar energy. They have backup batteries onboard to keep them running in the event of a solar eclipse, when there's no solar power. Small rocket boosters on each satellite keep them flying in the correct path.

Here are some other interesting facts about the GPS satellites (also called NAVSTAR, the official U.S. Department of Defense name for GPS):

• The first GPS satellite was launched in 1978.

• A full constellation of 24 satellites was achieved in 1994.

• Each satellite is built to last about 10 years. Replacements are constantly being built and launched into orbit.

• A GPS satellite weighs approximately 1.5 to 2 tons and is about 5 meters across with the solar panels extended. The orbits are arranged so that at any time, anywhere on Earth, there are at least four satellites "visible" in the sky.

• Transmitter power is only 50 watts or less.

GPS satellites transmit two low power radio signals, designated L1 and L2. Civilian GPS uses the L1 frequency of 1575.42 MHz in the UHF band. The signals travel by line of sight, meaning they will pass through clouds, glass and plastic but will not go through most solid objects such as buildings and mountains.

A GPS signal contains three different bits of information a pseudorandom code, ephemeris data and almanac data. The pseudorandom code is simply an I.D. code that identifies which satellite is transmitting information.

Ephemeris data, which is constantly transmitted by each satellite, contains important information about the status of the satellite (healthy or unhealthy), current date and time. This part of the signal is essential for determining a position.

The almanac data tells the GPS receiver where each satellite should be at any time throughout the day. Each satellite transmits almanac data showing the orbital information for that satellite and for every other satellite in the system.

Factors that can degrade the GPS signal and thus affect accuracy include the following:

• **Ionosphere and troposphere delays** The satellite signal slows as it passes through the atmosphere. The GPS system uses a built-in model that calculates an average amount of delay to partially correct for this type of error

• **Signal multipath** This occurs when the GPS signal is reflected off objects such as tall buildings or large rock surfaces before it reaches the receiver. This increases the travel time of the signal, thereby causing errors.

• **Orbital errors** Also known as ephemeris errors, these are inaccuracies of the satellite's reported location.

• **Number of satellites visible** The more satellites a GPS receiver can "see," the better the accuracy. Buildings, terrain, electronic interference, or sometimes even dense foliage can block signal reception, causing position errors or possibly no position reading at all. GPS units typically will not work indoors, underwater or underground.

• **Satellite geometry/shading** This refers to the relative position of the satellites at any given time. Ideal satellite geometry exists when the satellites are located at wide angles relative to each other. Poor geometry results when the satellites are located in a line or in a tight grouping.

• **Intentional degradation of the satellite signal** Selective Availability (SA) is an intentional degradation of the signal once imposed by the U.S. Department of Defense. SA was intended to prevent military adversaries from using the highly accurate GPS signals. The government turned off SA in May 2000, which significantly improved the accuracy of civilian GPS receivers.

Appendix C

Program Description and Software Codes

As noted earlier, the algorithm has been implemented in the GAMS software package which is a very useful tool for solving large mathematical programming problems, especially in the areas of function optimizing and compact representation of large and complex models. For our case, the software codes are presented for each case and explanatory comments follow to help the reader comprehend their structure. We will comment only on the first code (the three symmetric aircraft case) since the rest of the codes use a very similar approach and only the number of aircraft and some other minor details are different.

GAMS Code for conflict resolution of 3 aircraft symmetrically placed on the control circle

```
Set i "aircraft" /1* 3/;
alias (i,j);
Parameters pi,msd,M,omega(i,j),alpha(i,j),l(i,j),r(i,j);
Parameters radius, distance;
*Radius of the control volume distance of the final configuration point from
the cross-symmetric point of entry
radius=108; distance=108; pi=3.14159;
Parameters count(i) / 1 1 2 2 3 3 /;
*X-coordinates in Km
Parameters x(i) / 1 108 2 54 3 54 /;
*Y-coordinates in Km
Parameters y(i) / 1 0 2 93.531 3 -93.531 /;
*Initial heading angles in rads
```

```
Parameters theta(i) / 1 3.141 2 -1.047 3 1.047 /;
*Initial velocities in Km/min
Parameters u(i) / 1 15 2 15 3 15 /;
*minimum safe distance in Km
msd=5.4;
*Big M
M=50;
*Consider only pairs (i,j) where i≠j. This avoids including the same pair
twice, i.e. we include pair (1,2) but not (2,1)
omega(i,j) $ (count(i)<count(j) and x(i)=x(j))=pi/2;
omega(i,j) $ (count(i)<count(j) and x(i)≠x(j) )=arctan((y(i)-y(j))/(x(i)-x(j)));
alpha(i,j) $ (count(i)<count(j)) = sqrt((x(i)-x(j))*(x(i)-x(j))+(y(i)-y(j))*(y(i)-
y(j)));
l(i,j) $ (count(i)<count(j)) = omega(i,j)+ abs(arctan((msd/alpha(i,j))/(sqrt(1-
(msd/alpha(i,j))**2))));
r(i,j) $ (count(i)<count(j)) = omega(i,j)- abs(arctan((msd/alpha(i,j))/(sqrt(1-
(msd/alpha(i,j))**2)))) ;
Variable t,q(i),qdot(i),p(i),phi(i),d(i);
Binary variable B1(i),B2(i),B3(i);
Equations auxiliary(i),delay(i),time,velocity1(i),velocity2(i),velocity3(i),angle1(i),
angle2(i), A1(i,j),A2(i,j),A3(i,j),A4(i,j),A5(i,j),A6(i,j);
p.l(i)=0.01; phi.l(i)=0.01;
time.. t=e=sum(i,d(i));
auxiliary(i).. phi(i)=e=arctan( (radius*sin(2*p(i))) / (distance+2*radius*
(sin(p(i)))*(sin(p(i)))) );
delay(i).. d(i)=e=abs( (2*radius+distance)/u(i) - (2*radius*abs(cos(p(i))))/
(u(i)+q(i)) - ((radius*abs(sin(2*p(i)))) / abs(sin(phi(i)))) / (u(i)+q(i)+qdot(i))
);

velocity1(i).. q(i)=l=15.66-u(i);
velocity2(i).. -q(i)=l=u(i)-14.4;
velocity3(i).. qdot(i)=e=15.66-(u(i)+q(i));

A1(i,j).. ((u(i)+q(i))*cos(theta(i)+p(i))-(u(j)+q(j))*cos(theta(j)+p(j))-M*B1(i))
$ (count(i)<count(j))=l=0;

A2(i,j).. ((u(i)+q(i))*sin(theta(i)+p(i))-(u(i)+q(i))*cos(theta(i)+p(i))*
sin(l(i,j))/cos(l(i,j))-(u(j)+q(j))*sin(theta(j)+p(j)) +(u(j)+q(j))*cos(theta(j)+p(j))*
sin(l(i,j))/cos(l(i,j))-M*B2(i)-M*B1(i)) $ (count(i)<count(j))=l=0;

A3(i,j).. (-(u(i)+q(i))*sin(theta(i)+p(i))+(u(i)+q(i))*cos(theta(i)+p(i))*
```

```
sin(r(i,j))/cos(r(i,j))+(u(j)+q(j))*sin(theta(j)+p(j)) -(u(j)+q(j))*cos(theta(j)+p(j))*
sin(r(i,j))/cos(r(i,j))+M*B2(i)-M*B1(i)) $ (count(i)<count(j))=l=M;

angle1(i).. p(i)=l=0.1; angle2(i).. -p(i)=l=0.1;

Model nlc /all/ ; option domlim=10;
option nlp=conopt2;
option mip=cplex;
option rminlp=conopt2;
option minlp=dicopt;

solve nlc using rminlp minimizing t;
solve nlc using minlp minimizing t; display omega,l,alpha,r;
```

Though there are some helpful comments contained in the software code, we will now briefly analyze it and see how it behaves. First of all, we define the number of aircraft (in our case, 3) with the command *set* and we also define an *alias* for this set which is generally useful in models that are concerned with the interactions of elements within the same set.

Next, we proceed to define the various parameters: the maximum safe distance for collision avoidance (msd), the *big M* for the MINLP problem formulation (see page 37), as well as the other parameters used in the VC and HAC model, exactly as described in the corresponding sections. We also define the constant parameters of circle radius and distance of final configuration point from exit point.

After some values are set for the initial configuration of the aircraft (x-y coordinates, heading angles and velocities), we calculate the necessary variables, namely, the $\omega_{ij}, A_{ij}, l_{ij}$ and r_{ij} for each pair of aircraft (i, j). Note the \$ operator which is a conditional operator in GAMS. The term *\$(condition)* can be read as "such that condition is valid" where condition is a logical condition. For example, the following simple condition: **if (b > 1.5), then a = 2**, becomes in GAMS: **a\$(b > 1.5) = 2;**

Finally, we define and formulate the necessary MINLP constraints and inequalities (which GAMS refers to with the general term *equations*). Analytically, *time* refers to the total summing of the delays for each aircraft, *delay* calculates the delay itself (see page 35), *velocity1-velocity3* and *angle1-angle2* impose some bounds on the velocities and heading angles, while *A1-A6* are the main constraints as expressed in pages 34-35. The program concludes with the commands that select the various solvers for the model. In our case, we firstly solve a relaxed version of the MINLP problem, in which the integer restrictions

for variables *B1-B3* do not apply. This allows the program to converge quickly around a small set of feasible solutions and then, after imposing the integer condition, to find more easily the desired values.
The rest of the software codes follow:

GAMS Code for conflict resolution of 5 aircraft symmetrically placed on the control circle

```
Set i "aircraft" /1* 5/;
alias (i,j);
Parameters pi,msd,M,omega(i,j),alpha(i,j),l(i,j),r(i,j);
Parameters radius, distance;
*Radius of the control volume distance of the final configuration point from
the cross-symmetric point of entry
radius=108; distance=108; pi=3.14159;
Parameters count(i) / 1 1 2 2 3 3 4 4 5 5 /;
*X-coordinates in Km
Parameters x(i) / 1 108 2 33.3738 3 -87.3738 4 -87.3738 5 33.3738 /;
*Y-coordinates in Km
Parameters y(i) / 1 0 2 102.714 3 63.48876 4 -63.48876 5 -102.714 /;
*Initial heading angles in rads
Parameters theta(i) / 1 0 2 -1.885 3 -0.628 4 0.628 5 1.885 /;
*Initial velocities in Km/min
Parameters u(i) / 1 15 2 15 3 15 4 15 5 15 /;
*minimum safe distance in Km
msd=5.4;
*Big M
M=50;
*Consider only pairs (i,j) where i≠j. This avoids including the same pair
twice, i.e. we include pair (1,2) but not (2,1)
omega(i,j) $ (count(i)<count(j) and x(i)=x(j))=pi/2;
omega(i,j) $ (count(i)<count(j) and x(i)≠x(j) )=arctan((y(i)-y(j))/(x(i)-x(j)));
alpha(i,j) $ (count(i)<count(j)) = sqrt((x(i)-x(j))*(x(i)-x(j))+(y(i)-y(j))*(y(i)-
y(j)));
l(i,j) $ (count(i)<count(j)) = omega(i,j)+ abs(arctan((msd/alpha(i,j))/(sqrt(1-
(msd/alpha(i,j))**2))));
r(i,j) $ (count(i)<count(j)) = omega(i,j)- abs(arctan((msd/alpha(i,j))/(sqrt(1-
(msd/alpha(i,j))**2)))) ;
Variable t,q(i),qdot(i),p(i),phi(i),d(i);
Binary variable B1(i),B2(i),B3(i);
```

```
Equations auxiliary(i),delay(i),time,velocity1(i),velocity2(i),velocity3(i),angle1(i),
angle2(i), A1(i,j),A2(i,j),A3(i,j),A4(i,j),A5(i,j),A6(i,j);
p.l(i)=0.01; phi.l(i)=0.01;
time.. t=e=sum(i,d(i));
auxiliary(i).. phi(i)=e=arctan( (radius*sin(2*p(i))) / (distance+2*radius*
(sin(p(i)))*(sin(p(i)))) );
delay(i).. d(i)=e=abs( (2*radius+distance)/u(i) - (2*radius*abs(cos(p(i))))/
(u(i)+q(i)) - ((radius*abs(sin(2*p(i)))) / abs(sin(phi(i)))) / (u(i)+q(i)+qdot(i))
);

velocity1(i).. q(i)=l=15.66-u(i);
velocity2(i).. -q(i)=l=u(i)-14.4;
velocity3(i).. qdot(i)=e=15.66-(u(i)+q(i));

A1(i,j).. ((u(i)+q(i))*cos(theta(i)+p(i))-(u(j)+q(j))*cos(theta(j)+p(j))-M*B1(i))
$ (count(i)<count(j))=l=0;

A2(i,j).. ((u(i)+q(i))*sin(theta(i)+p(i))-(u(i)+q(i))*cos(theta(i)+p(i))*
sin(l(i,j))/cos(l(i,j))-(u(j)+q(j))*sin(theta(j)+p(j)) +(u(j)+q(j))*cos(theta(j)+p(j))*
sin(l(i,j))/cos(l(i,j))-M*B2(i)-M*B1(i)) $ (count(i)<count(j))=l=0;

A3(i,j).. (-(u(i)+q(i))*sin(theta(i)+p(i))+(u(i)+q(i))*cos(theta(i)+p(i))*
sin(r(i,j))/cos(r(i,j))+(u(j)+q(j))*sin(theta(j)+p(j)) -(u(j)+q(j))*cos(theta(j)+p(j))*
sin(r(i,j))/cos(r(i,j))+M*B2(i)-M*B1(i)) $ (count(i)<count(j))=l=M;

angle1(i).. p(i)=l=0.1; angle2(i).. -p(i)=l=0.1;

Model nlc /all/ ; option domlim=10;
option nlp=conopt2;
option mip=cplex;
option rminlp=conopt2;
option minlp=dicopt;

solve nlc using rminlp minimizing t;
solve nlc using minlp minimizing t; display omega,l,alpha,r;
```

GAMS Code for conflict resolution of 7 aircraft symmetrically placed on the control circle

```
Set i "aircraft" /1* 7/;
```

```
alias (i,j);
Parameters pi,msd,M,omega(i,j),alpha(i,j),l(i,j),r(i,j);
Parameters radius, distance;
*Radius of the control volume distance of the final configuration point from
the cross-symmetric point of entry
radius=108; distance=108; pi=3.14159;
Parameters count(i) / 1 1 2 2 3 3 4 4 5 5 6 6 7 7 /;
*X-coordinates in Km
Parameters x(i) / 1 108 2 67.337 3 -24.0321 4 -97.3046 5 -97.3046 6 -24.0321
7 67.337 /;
*Y-coordinates in Km
Parameters y(i) / 1 0 2 84.44 3 105.2922 4 46.8622 5 -46.8622 6 -105.2922
7 -84.44 /;
*Initial heading angles in rads
Parameters theta(i) / 1 3.14 2 -2.244 3 -1.346 4 -0.448 5 0.448 6 1.346 7
2.244 /;
*Initial velocities in Km/min
Parameters u(i) / 1 15 2 15 3 15 4 15 5 15 6 15 7 15 /;
*minimum safe distance in Km
msd=5.4;
*Big M
M=50;
*Consider only pairs (i,j) where i≠j. This avoids including the same pair
twice, i.e. we include pair (1,2) but not (2,1)
omega(i,j) $ (count(i)<count(j) and x(i)=x(j))=pi/2;
omega(i,j) $ (count(i)<count(j) and x(i)≠x(j) )=arctan((y(i)-y(j))/(x(i)-x(j)));
alpha(i,j) $ (count(i)<count(j)) = sqrt((x(i)-x(j))*(x(i)-x(j))+(y(i)-y(j))*(y(i)-
y(j)));
l(i,j) $ (count(i)<count(j)) = omega(i,j)+ abs(arctan((msd/alpha(i,j))/(sqrt(1-
(msd/alpha(i,j))**2))));
r(i,j) $ (count(i)<count(j)) = omega(i,j)- abs(arctan((msd/alpha(i,j))/(sqrt(1-
(msd/alpha(i,j))**2)))) ;
Variable t,q(i),qdot(i),p(i),phi(i),d(i);
Binary variable B1(i),B2(i),B3(i);
Equations auxiliary(i),delay(i),time,velocity1(i),velocity2(i),velocity3(i),angle1(i),
angle2(i), A1(i,j),A2(i,j),A3(i,j),A4(i,j),A5(i,j),A6(i,j);
p.l(i)=0.01; phi.l(i)=0.01;
time.. t=e=sum(i,d(i));
auxiliary(i).. phi(i)=e=arctan( (radius*sin(2*p(i))) / (distance+2*radius*
(sin(p(i)))*(sin(p(i)))) );
delay(i).. d(i)=e=abs( (2*radius+distance)/u(i) - (2*radius*abs(cos(p(i))))/
```

```
(u(i)+q(i)) - ((radius*abs(sin(2*p(i)))) / abs(sin(phi(i)))) / (u(i)+q(i)+qdot(i))
);

velocity1(i).. q(i)=l=15.66-u(i);
velocity2(i).. -q(i)=l=u(i)-14.4;
velocity3(i).. qdot(i)=e=15.66-(u(i)+q(i));

A1(i,j).. ((u(i)+q(i))*cos(theta(i)+p(i))-(u(j)+q(j))*cos(theta(j)+p(j))-M*B1(i))
$ (count(i)<count(j))=l=0;

A2(i,j).. ((u(i)+q(i))*sin(theta(i)+p(i))-(u(i)+q(i))*cos(theta(i)+p(i))*
sin(l(i,j))/cos(l(i,j))-(u(j)+q(j))*sin(theta(j)+p(j)) +(u(j)+q(j))*cos(theta(j)+p(j))*
sin(l(i,j))/cos(l(i,j))-M*B2(i)-M*B1(i)) $ (count(i)<count(j))=l=0;

A3(i,j).. (-(u(i)+q(i))*sin(theta(i)+p(i))+(u(i)+q(i))*cos(theta(i)+p(i))*
sin(r(i,j))/cos(r(i,j))+(u(j)+q(j))*sin(theta(j)+p(j)) -(u(j)+q(j))*cos(theta(j)+p(j))*
sin(r(i,j))/cos(r(i,j))+M*B2(i)-M*B1(i)) $ (count(i)<count(j))=l=M;

angle1(i).. p(i)=l=0.1; angle2(i).. -p(i)=l=0.1;

Model nlc /all/ ; option domlim=10;
option nlp=conopt2;
option mip=cplex;
option rminlp=conopt2;
option minlp=dicopt;

solve nlc using rminlp minimizing t;
solve nlc using minlp minimizing t; display omega,l,alpha,r;
```

GAMS Code for conflict resolution of 9 aircraft symmetrically placed on the control circle

```
Set i ”aircraft” /1* 9/;
alias (i,j);
Parameters pi,msd,M,omega(i,j),alpha(i,j),l(i,j),r(i,j);
Parameters radius, distance;
*Radius of the control volume distance of the final configuration point from
the cross-symmetric point of entry
radius=108; distance=108; pi=3.14159;
```

```
Parameters count(i) / 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 /;
*X-coordinates in Km
Parameters x(i) / 1 108 2 82.7328 3 18.754 4 -54 5 -101.4868 6 -101.4868
7 -54 8 18.754 9 82.7328 /;
*Y-coordinates in Km
Parameters y(i) / 1 0 2 69.4211 3 106.3596 4 93.528 5 36.9412 6 -36.9412
7 -93.528 8 -106.3596 9 -69.4211 /;
*Initial heading angles in rads
Parameters theta(i) / 1 3.14 2 -2.443 3 -1.745 4 -1.047 5 -0.349 6 0.349 7
1.047 8 1.745 9 2.443 /;
*Initial velocities in Km/min
Parameters u(i) / 1 15 2 15 3 15 4 15 5 15 6 15 7 15 8 15 9 15 /;
*minimum safe distance in Km
msd=5.4;
*Big M
M=50;
*Consider only pairs (i,j) where i≠j. This avoids including the same pair
twice, i.e. we include pair (1,2) but not (2,1)
omega(i,j) $ (count(i)<count(j) and x(i)=x(j))=pi/2;
omega(i,j) $ (count(i)<count(j) and x(i)≠x(j) )=arctan((y(i)-y(j))/(x(i)-x(j)));
alpha(i,j) $ (count(i)<count(j)) = sqrt((x(i)-x(j))*(x(i)-x(j))+(y(i)-y(j))*(y(i)-
y(j)));
l(i,j) $ (count(i)<count(j)) = omega(i,j)+ abs(arctan((msd/alpha(i,j))/(sqrt(1-
(msd/alpha(i,j))**2))));
r(i,j) $ (count(i)<count(j)) = omega(i,j)- abs(arctan((msd/alpha(i,j))/(sqrt(1-
(msd/alpha(i,j))**2)))) ;
Variable t,q(i),qdot(i),p(i),phi(i),d(i);
Binary variable B1(i),B2(i),B3(i);
Equations auxiliary(i),delay(i),time,velocity1(i),velocity2(i),velocity3(i),angle1(i),
angle2(i), A1(i,j),A2(i,j),A3(i,j),A4(i,j),A5(i,j),A6(i,j);
p.l(i)=0.01; phi.l(i)=0.01;
time.. t=e=sum(i,d(i));
auxiliary(i).. phi(i)=e=arctan( (radius*sin(2*p(i))) / (distance+2*radius*
(sin(p(i)))*(sin(p(i)))) );
delay(i).. d(i)=e=abs( (2*radius+distance)/u(i) - (2*radius*abs(cos(p(i))))/
(u(i)+q(i)) - ((radius*abs(sin(2*p(i)))) / abs(sin(phi(i)))) / (u(i)+q(i)+qdot(i))
);

velocity1(i).. q(i)=l=15.66-u(i);
velocity2(i).. -q(i)=l=u(i)-14.4;
velocity3(i).. qdot(i)=e=15.66-(u(i)+q(i));
```

```
A1(i,j).. ((u(i)+q(i))*cos(theta(i)+p(i))-(u(j)+q(j))*cos(theta(j)+p(j))-M*B1(i))
$ (count(i)<count(j))=l=0;

A2(i,j).. ((u(i)+q(i))*sin(theta(i)+p(i))-(u(i)+q(i))*cos(theta(i)+p(i))*
sin(l(i,j))/cos(l(i,j))-(u(j)+q(j))*sin(theta(j)+p(j)) +(u(j)+q(j))*cos(theta(j)+p(j))*
sin(l(i,j))/cos(l(i,j))-M*B2(i)-M*B1(i)) $ (count(i)<count(j))=l=0;

A3(i,j).. (-(u(i)+q(i))*sin(theta(i)+p(i))+(u(i)+q(i))*cos(theta(i)+p(i))*
sin(r(i,j))/cos(r(i,j))+(u(j)+q(j))*sin(theta(j)+p(j)) -(u(j)+q(j))*cos(theta(j)+p(j))*
sin(r(i,j))/cos(r(i,j))+M*B2(i)-M*B1(i)) $ (count(i)<count(j))=l=M;

angle1(i).. p(i)=l=0.1; angle2(i).. -p(i)=l=0.1;

Model nlc /all/ ; option domlim=10;
option nlp=conopt2;
option mip=cplex;
option rminlp=conopt2;
option minlp=dicopt;

solve nlc using rminlp minimizing t;
solve nlc using minlp minimizing t; display omega,l,alpha,r;
```

The codes for the random aircraft configuration for 3, 5 and 7 aircraft are exactly the same as the ones for the symmetric configuration except they have different numerical values for their initial aircraft configurations, so they are not presented here.

Bibliography

[1] www.centennialofflight.gov/index2.cfm, *U.S. Centennial of Flight Commission Website*.

[2] www.faa.gov, *The Federal Aviation Administration Website*.

[3] *The Future of Air Traffic Control: Human Operators and Automation*. The National Academy Press, 1998.

[4] www.rtca.org/default.asp, *The Radio Technical Commission for Aeronautics Website*.

[5] J. M. Hoekstra. Designing for Safety: the Free Flight Air Traffic Management Concept, *Doctorate Thesis*, Delft University of Technology, Holland, 2001

[6] A. Bicchi and L. Pallottino. On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, December 2000.

[7] L. Pallottino, E. Feron and A. Bicchi. Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming. *Journal*, 2001.

[8] Z.-H. Mao, E. Feron and K. Bilimoria. Stability of Intersecting Aircraft Flows Under Decentralized Conflict Avoidance Rules. *The American Institute of Aeronautics and Astronautics*, AIAA-2000-4271, 2001.

[9] L. Pallottino, E. Feron and A. Bicchi. Mixed Integer Programming for Aircraft Conflict Resolution. *The American Institute of Aeronautics and Astronautics*, AIAA-2001-4295, 2001.

[10] D. Dugail, Z.-H. Mao and E. Feron. Stability of Intersecting Aircraft Flows Under Centralized and Decentralized Conflict Avoidance Rules. *The American Institute of Aeronautics and Astronautics*, AIAA-2000-4296, 2001.

[11] A. Richards, J. How, T. Schouwenaars and E. Feron. Plume Avoidance Maneuver Planning Using Mixed Integer Linear Programming. *The American Institute of Aeronautics and Astronautics*, AIAA-2000-4091, 2001.

[12] W. P. Niedringhaus. Stream Option Manager (SOM): Automated Integration of Aircraft Separation, Merging, Stream Management and Other Air Traffic Control Functions. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 9, September 1995.

[13] C. Tomlin, G. J. Pappas and S. Sastry. Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems. *IEEE Transactions on Automatic Control*, July 1998.

[14] E. Frazzoli, Z.-H. Mao, J.-H. Oh and E. Feron. Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming. *Journal of Guidance, Control and Dynamics*, 2000.

[15] www.gams.de, GAMS IDE 2.0.20.0. *GAMS User Guide*, 2002.

[16] C. A. Floudas. *Nonlinear and Mixed-Integer Optimization*. Oxford University Press, 1995.

[17] F. S. Hillier and G. J. Liebermann. *Introduction to Operations Research*. McGraw-Hill Inc., 1995.

[18] www.garmin.com/aboutGPS, *Garmin GPS Website*.

[19] www.nlr.nl, *The National Aerospace Laboratory Website*.

[20] www.airtrafficcafe.com, *The Air Traffic Cafe Website*.

[21] www.aerosite.net.

Index

A

B

C

F

G

H

L

M

N

O

R

S

T

V