Free Flight Control using Generalized Bender’s Decomposition for Automatic Conflict Avoidance

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

Michael Chadjitheocharous

July 2005

Contents

List of Figures

CHAPTER 1: INTRODUCTION

1.1:AIR TRAFFIC MANAGEMENT

1.1.1:HISTORICAL PART

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. 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 towersbegan 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” 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, 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 , 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, radiobroadcasts 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 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.

1.1.2:DEFINITION

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 nautic miles. It depends on the equipment available to control aircraft. It is usually 8 or 5 Nautic 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, ...): 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.

Then why is it so difficult to have aircraft separated? Answers are different regarding the country you live in. In the USA, airport capacity is the main problem, while in Europe, and mainly in France, En Route capacity is the critical point.

In the USA, the problem is mainly an airport capacity problem. This problem exists also in Europe on the biggest airport. For example, on a given airport, you can not have more than six aircraft landing or departing each 10 minutes.

If you have ever taken off at JFK or Chicago International Airport, you have already noticed that there is a serious queue before departing. The same is true for aircraft landing. They are usually

waiting inside stacks, close to the airport landing airway. The problem is then to allocate slots for landing to aircraft;
as the system is not fully predictable, you can not give to a aircraft a precise slot for landing when it takes off thousand miles away.
Many things can shorten its flight time (good winds), or delay the aircraft (storms for example). So, when it arrives, the aircraft as usually to wait a few minutes before landing.
En Route capacity is a problem mainly in Europe. As stated above, airspace is divided in control sectors, each sector being managed by, usually, two Air Traffic controllers. however, the capacity of a sector is limited.
A controller can not handle more than a certain number of aircraft in its sector. This is called sector capacity. However, it is not possible to divide a sector in two just to increase capacity of the general ATC system, and it is easy to understand why.
On the one hand, the sector has to be large enough to enable the controller to perform conflict resolution inside the sector. But there is also a problem linked with controller workload. How is controller workload computed? There are three factors to consider:
Monitoring workload: it is simply the monitoring of the aircraft in the controller's sector. It depends mainly on the number of planes.
Resolution workload: the resolution induced by the resolution of conflicts
Coordination workload: coordination is a task that each controller must perform when a aircraft enters (or leaves) its sector. Basically, it is a negociation with the controller which had the aircraft in charge (or that will take the aircraft in charge) regarding, for example, the level the aircraft will enter or leave the sector. The

number of coordinations will increase when the size of the sector becomes smaller, and coordination workload could soon become the main factor of stress.

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: FREE FLIGHT

”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.”

These are the words by a working committee on free flight sponsored by RTCA .

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:

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

II. 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

III. 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[3]. 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 order to avoid the rigid structure which puts strict constraints on aircraft trajectories , which could otherwise follow wind-optimal or user preferred routes. Also, while a data link between aircraft and ground is being investigated as a replacement for the current voice communication over radio channels between pilot and controller , there is a limit to the amount of information processing that a controller can perform with this data.

Two new technologies for air traffic control will be certified for use in the cockpit in the very near future: a positioning system based on GPS, and a data communication network linking aircraft to each other and to the ground control system, called ADS (Automatic Dependent Surveillance), or ADS-B (Broadcast). Both of these technologies will have the effect of moving today's ground-based navigation and communication equipment into the air. These technologies are expected toprovide short term improvements; they will not provide a long term solution to the air traffic problem. The result is a perceived need in the air traffic, airline, and avionics communities for a new *architecture*, which integrates new technologies for data storage, processing, communications, and display, into a safe and efficient air traffic management system. The airlines are proponents of a decentralized architecture featuring *free flight*, meaning that each aircraft plans and tracks its own dynamic trajectory with minimal interference from ATC [4]. Many view this as a radical solution, but a recent study funded by NASA suggests that distributing some of the control authority to each aircraft would help improve the efficiency of the system as a whole. While the degree of decentralization and level of automation in a new air traffic management system are still under debate (since it is very difficult to estimate the increase in efficiency from distributing the control authority), the integrity of any automated functionality in a new air traffic management system depends on a *provably-safe* design, and high confidence that the control actions will not fail.

CHAPTER 2 : 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 progr
amming, so that the basis for the more advanced concepts will be understood.

2.1: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, $x_{2,\dots\dots}$ 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.

2.1.1:Comprehensive Examples

2.1.1.1:Example1 : Pollution control

In the simplest pollution control problem there are emission sources i =1…. m and receptor points j =1…n [5].For every source i , a finite set K(i) of treatment technologies is available. Each technology k(i) $\in$ K(i) has cost c_{iki} and is associated with an emission level e_{iki}. The emissions are transferred to receptors to produce depositions

$$y_j(k_1,\ldots,k_m,\theta) = \sum_{i=1}^{m} t_{ij}(\theta)e_{ik_i},$$

where $t_{ij}(\theta)$ are some random transfer coefficients. Finally there are some target levels (ambient norms) of deposition q_j for the receptors j=1…n They are used to formulate a penalty cost $\varphi_j(y_j)$ associated with each deposition e.g.,

$$\varphi_j(y_j)=\max(0\ ,\ y_j-g_j)$$

The problem is to find the technologies $k_1,\ldots,k_m$ so as to minimize the penalty function

$$F(k_1,\ldots,k_m) = \mathbf{E}\left\{\sum_{j=1}^{n}\varphi_j\left(\sum_{i=1}^{m} t_{ij}(\theta)e_{ik_i}\right)\right\}$$

subject to the budget constraint

$$\sum_{i=1}^{m} c_{ik_i} \leq R.$$

2.1.1.2:Example2 : Facility location

A set N={1,2…n} of potential facility locations and a set of clients I={1,2…m}
are given. A facility placed at location j costs c_j and has capacity u_j _
Clients have random demands $d_i(\theta)$, i=1,…,m and the unit cost to satisfy the demand of client i from
facility j is q_{ij} .There is also a shortage cost q_{i0} for each unit of client's i demand, which
is not satisfied by any of the facilities. The problem is to choose locations of facilities that
minimize the total expected cost.
Defining binary variables

$$x_j = \begin{cases} 1 & \text{if facility is placed at } j, \\ 0 & \text{otherwise,} \end{cases}$$

one can formalize the problem as follows

$$\min\left[F(x) = \sum_{j=1}^{n} c_j x_j + \mathbf{E}\varphi(x,\theta)\right]$$

$$x_j \in \{0,1\},\ j = 1,\ldots,n,$$

where φ(x,θ) is defined as the minimum cost of satisfying the demand. It is the optimalvalue of the transportation problem

$$\min \sum_{i=1}^{m}\sum_{j=0}^{n} q_{ij}y_{ij}$$

$$\sum_{j=0}^{n} y_{ij} = d_i(\theta),\ i = 1, \ldots, m$$

$$\sum_{i=1}^{m} y_{ij} \le x_j u_j,\ j = 1, \ldots, n.$$

$$y_{ij} \ge 0,\ i = 1, \ldots, m,\ j = 1, \ldots, n,$$

where y_{ij} is the demand of client I served by facility J.

2.1.2:Formulation of the Linear Programming Model

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

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 theindividual 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[7].

Z=value of overall measure of performance.

x_j=level of activity j (for $j = 1, 2, \ldots, n$).

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 $i = 1, 2,\ldots,m$).

a_{ij}=amount of resource *i* consumed by each unit of activity *j*.

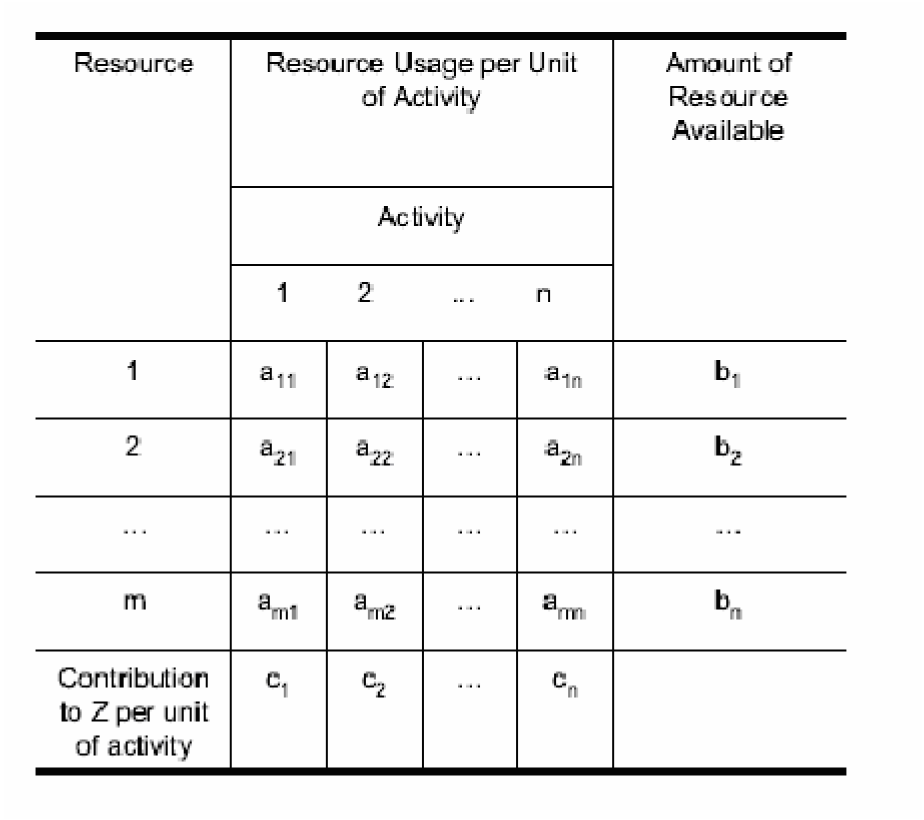
2.1.3:Assumptions of Linear Programming

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[8].

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	

Table 2. 1:Data needed for 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.

2.1.4: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))[9]. 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.

2.2: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 j^{th} 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.

2.2.1: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.

2.2.2: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)[10]. 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:

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

$$x_1 + 4x_2 \leq 16 + M(1 - y)$$

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 xj 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 $y1$ and $y2$ 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.

2.2.3: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:

$f_1(x_1, x_2,\ldots, x_n) \leq d_1$

$f_2(x_1, x_2,\ldots , x_n) \leq d2$

.

.

.

$f_n(x_1, x_2,\ldots , x_n) \leq d_n$

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

$f_1(x_1, x_2,\ldots, x_n) \leq d_1+My_1$

$f_2(x_1, x_2,\ldots , x_n) \leq d2+My_2$

.

.

.

$f_n(x_1, x_2, \ldots , x_n) \leq d_n + My_n$

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

and

y_i is binary, for $i = 1, 2, \ldots, N$,

where M is an extremely large positive number. For each binary variable y_i ($i = 1, 2, \ldots, N$), note that $yi = 0$ makes $M y_i = 0$, which reduces the newconstraint 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 yi 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.

2.2.4: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

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

One special case is where this function is

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

as on the left-hand side of a linear programming constraint. Another special case is where $f(x_1, x_2, \ldots, x_n) = x_j$ 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:

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

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

and

y_i is binary, for $i = 1, 2, \ldots, N$,

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 yi must equal 1 and the others must equal 0, so exactly one di is being chosen as the value of the function. In this case, there are N yes-or-no questions being asked, namely, should di be the value chosen (i = 1, 2,…,N)? Because the yi respectively represent these yes-or-no decisions, the second constraint makes them mutually exclusive alternatives.

2.2.5: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:

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

or

$$f_j(x_{ij}) = 0 \quad , \quad x_j = 0$$

where x_j denotes the level of activity j ($x_j >= 0$), 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 $k_j >= 0$ in every case and $k_j > 0$ for some $j = 1, 2, \dots, n$), and that the problem is to

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

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 yj , so that

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

where

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

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 xj (j = 1, 2,…, n). Then the constraints

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

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 yj 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 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.

2.2.6: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

$$x_1 \leq 5$$
$$2x_1 + 3x_2 \leq 30$$

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

$$x_1 = y_0 + 2y_1 + 4y_2$$
$$x_2 = y_3 + 2y_4 + 4y_5 + 8y_6$$

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

$$y_0 + 2y_1 + 4y_2 \leq 5$$
$$2y_0 + 4y_1 + 8y_2 + 3y_3 + 6y_4 + 12y_5 + 24y_6 \leq 30$$

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.

2.3: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) . 0, (g_j (x) =0, j =1, 2, …, p), and is denoted as

$$\begin{gathered}\text{Maximize } f(x)\\ \text{subject to}\\ g_j(x) < b_j \quad , j = 1, 2, \ldots, m\end{gathered}$$

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 S²<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 -λ) x_1 +λ 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 R^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, $x_2 \in S$, 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 R^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, $x_2 \in S$, 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.

2.4:Mixed Integer NonLinear Programming

Recently, the area of Mixed Integer Nonlinear Programming (MINLP) has experienced tremendous growth and a flourish of research activity. We will give a brief overview of past developments in the MINLP arena and discuss some of the future work that can aid the development of MINLP.

2.4.1:Introduction

Mixed Integer Nonlinear Programming (MINLP) refers to mathematical programming with continuous and discrete variables and nonlinearities in the objective function and constraints[12]. The use of MINLP is a natural approach of formulating problems where it is necessary to simultaneously optimize the system structure (discrete) and parameters (continuous).

MINLPs have been used in various applications, including the process industry and the financial, engineering, management science and operations research sectors. It includes problems in process flow sheets, portfolio selection, batch processing in chemical engineering (consisting of mixing, reaction, and centrifuge separation), and optimal design of gas or water transmission networks. Other areas of interest include the automobile, aircraft, and VLSI manufacturing areas. The needs in such diverse areas have motivated research and development in MINLP solver technology, particularly in algorithms for handling large-scale, highly combinatorial and highly nonlinear problems.

The general form of a MINLP is:

$$\begin{array}{lll} \text{minimize} & f(x,y) & \\ \text{subject to} & g(x,y) \leq 0 & \\ & x \in X & \\ & y \in Y & \text{integer} \end{array}$$

(2.4.1)

The function f(x,y) is a nonlinear objective function and g(x,y) a nonlinear constraint function. The variables x, y are the decision variables, where y is required to be integer valued. X and Y are bounding-box-type restrictions on the variables.

2.4.2:Algorithms

MINLP problems are precisely so difficult to solve, because they combine all the difficulties of both of their subclasses: the combinatorial nature of mixed integer programs (MIP) and the difficulty in solving nonconvex (and even convex) nonlinear programs (NLP). Because subclasses MIP and NLP are among the class of theoretically difficult problems (NP-complete), so it is not surprising that solving MINLP can be a challenging and daring venture. Fortunately, the component structure of MIP and NLP within MINLP provides a collection of natural algorithmic approaches, exploiting the structure of each of the subcomponents.

CHAPTER 3

3.1:Flight Conditions

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

Aircrafts are identified by points in the plane (position) and angles (heading angle, direction) and thus by a point (x, y, θ) Є $R\mathrm{x}R\mathrm{x}S^1$. Let $(x_1(t),y_1(t),\theta_1(t))$ be the configuration of aircraft *1* at time t, and $(x_2(t),y_2(t),\theta_2(t))$ the configuration of aircraft *2* at the same time t. A conflict occurs among these aircrafts when the distance between them is less than *d* i.e. a conflict between aircraft *i* and *j* occurs if for some value of t,

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

In both cases each aircraft is allowed to make maneuver, at time t = 0, to avoid all possible conflicts with other aircraft. We assume that no conflict occurs at time t = 0. Considering the aircraft as disks 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 disc* of the aircraft.

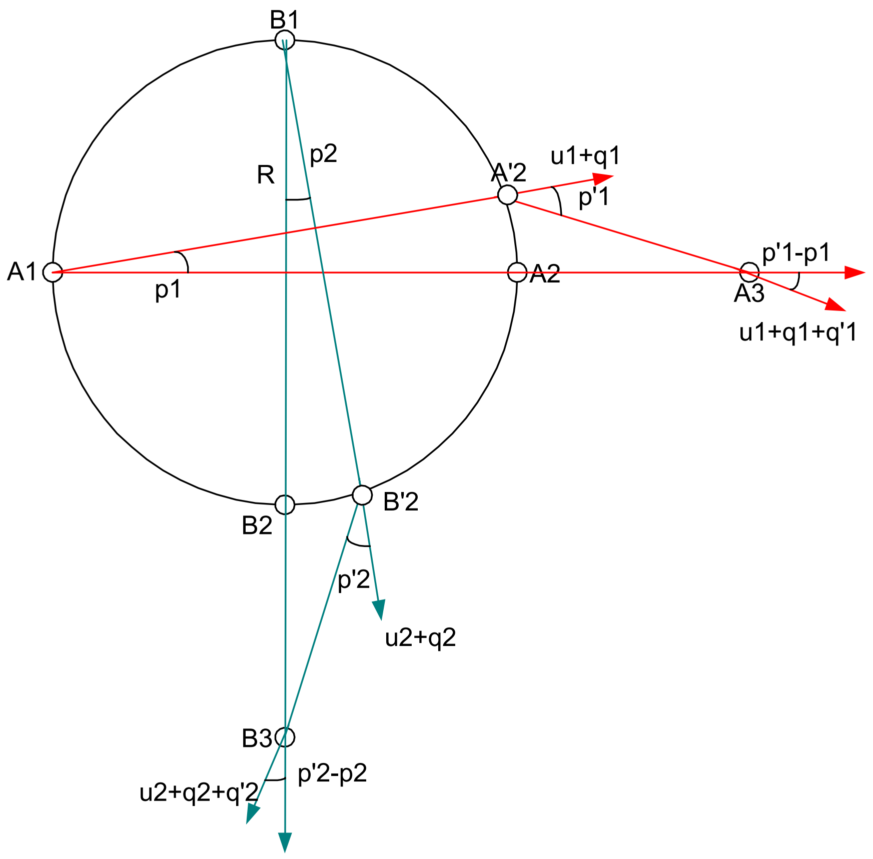


Figure 3.1 A comprehensive example figure about the aircraft's behavior before and after conflict resolution.

As mentioned previously, to avoid possible conflicts we consider two different cases:

a. we allow aircraft to change the velocity of flight but the direction of motion remains fixed. We will refer to this case as the *Velocity Change Problem* (VC problem).
b. aircraft fly at the same velocity *u* and are only allowed to change instantaneously the direction of flight. We will refer to this case as the *Heading Angle Problem* (HAC problem).

3.1.1:VELOCITY CHANGE

We will now examine the velocity change problem in order to find applicable values of the velocity of both aircrafts.
The VC problem consists of aircraft that fly along a given fixed direction and can maneuver only once with a velocity variation. We consider two aircrafts denoted 1 and 2 respectively. The i-th (i = 1,2) aircraft changes its velocity of a quantity *qi* that can be positive (acceleration), negative (deceleration), or null (no velocity change). Each aircraft has upper and lower bounds on the velocity *ui* :

$$v_{i,min} \leq v_i \leq v_{i,max}. \qquad 3.4)$$

Let *(x_i, y_i , θ_i), i = 1,2*, be the aircraft position and direction of motion and *ui* be the initial velocity. Referring to Figure 3.1, we consider the two velocity vectors:

$$\hat{v}_1 = \begin{bmatrix} (v_1 + q_1)cos\theta_1 \\ (v_1 + q_1)sin\theta_1 \end{bmatrix};$$

(3.5)

and

$$\hat{v}_2 = \begin{bmatrix} (v_2 + q_2)cos\theta_2 \\ (v_2 + q_1)sin\theta_2 \end{bmatrix};$$

(3.6)

and the difference vector:

$$\hat{v}_1 - \hat{v}_2 = \begin{bmatrix} (v_1 + q_1)cos\theta_1 - (v_2 + q_2)cos\theta_2 \\ (v_1 + q_1)sin\theta_1 - (v_2 + q_1)sin\theta_2 \end{bmatrix}; \quad (3.7)$$

The two lines parallel to $\hat{u}_1$-$\hat{u}_2$ that are tangent to aircraft 2, localize the segment on the direction on motion of aircraft 1 (refer to 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 aircraft 1 with his safe disc intersects the shadow generated by aircraft 2, or vice-versa since $\hat{u}_1$-$\hat{u}_2$ and $\hat{u}_2$-$\hat{u}_1$ are parallel.

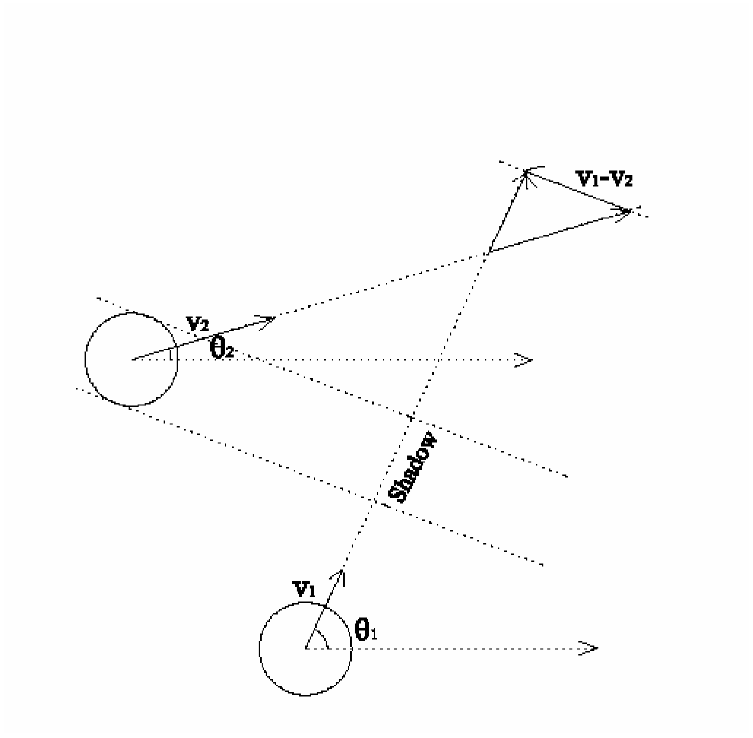


Figure 3.2 *Projection of the motion for conflict avoidance of two aircrafts constraints in the case of intersecting trajectories for the VC problem. In the case aircraft 1 do not intersect the shadow generated by aircraft 2 then no conflict will occur between the two aircraft.*

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 $l_{12} = \omega_{12} + \alpha$ and $r_{12} = \omega_{12} - \alpha$ with $\alpha = arcsin(d / A_{12})$ 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 *x*-axis.

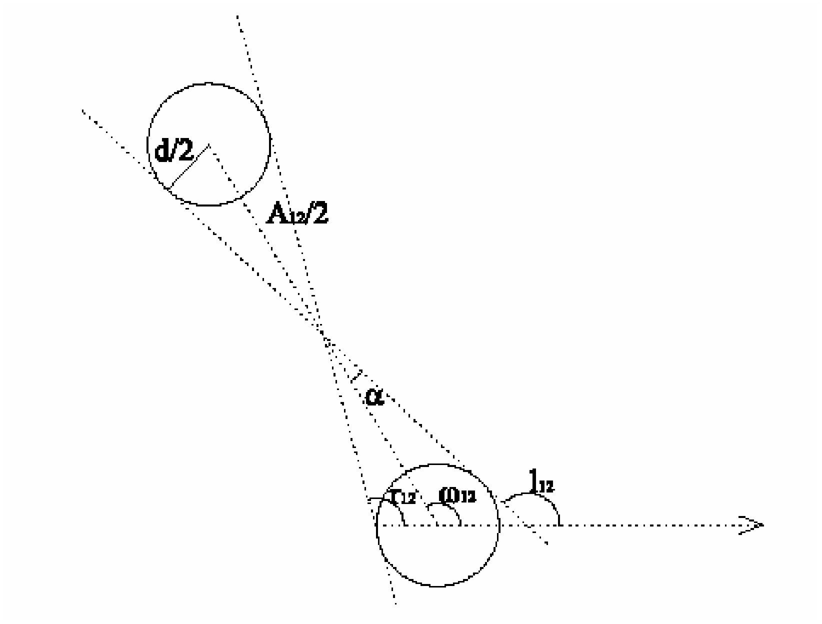


Figure 3.3: *The two non parallel straight lines tangent to the safety discs of radius d/2 for two aircrafts at distance A12 / 2.*

From the above conditions we can say that no conflict occurs if:

$$\frac{(v_1+q_1)sin\theta_1-(v_2+q_2)sin\theta_2}{(v_1+q_1)cos\theta_1-(v_2+q_2)cos\theta_2} \geq \tan(l_{12}) \qquad (3.8)$$

or

$$\frac{(v_1+q_1)sin\theta_1-(v_2+q_2)sin\theta_2}{(v_1+q_1)cos\theta_1-(v_2+q_2)cos\theta_2} \leq \tan(r_{12}) \qquad (3.9)$$

We now distinguish two possible cases:

1) $(u_1+q_1)\cos(\theta_1)-(u_2+q_2)\cos(\theta_2)<0$

and

2) $(u_1+q_1)\cos(\theta_1)-(u_2+q_2)\cos(\theta_2)>0$

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*) $(u_i+q_i)\cos(\theta_i)-(u_j+q_j)\cos(\theta_j)<0$**

$$\begin{cases} q_i\cos(\theta_i)-q_j\cos(\theta_j)\le -u_i\cos(\theta_i)+u_j\cos(\theta_j) \\ -h_i q_i+h_j q_j\le u_i h_i-u_j h_j \end{cases}$$

or

$$\begin{cases} q_i\cos(\theta_i)-q_j\cos(\theta_j)\le -u_i\cos(\theta_i)+u_j\cos(\theta_j) \\ k_i q_i-k_j q_j\le -u_i k_i+u_j k_j \end{cases}$$

***Case 2*) $(u_i+q_i)\cos(\theta_i)-(u_j+q_j)\cos(\theta_j)>0$**

$$\begin{cases} -q_i\cos(\theta_i)+q_j\cos(\theta_j)\le u_i\cos(\theta_i)-u_j\cos(\theta_j) \\ h_i q_i-h_j q_j\le -u_i h_i+u_j h_j \end{cases}$$

or

$$\begin{cases} -q_i \cos(\theta_i) + q_j \cos(\theta_j) \leq u_i \cos(\theta_i) - u_j \cos(\theta_j) \\ -k_i q_i + k_j q_j \leq u_i k_i - u_j k_j \end{cases}$$

Now we have the or-constrains for the VC problem. We also notice that the above constrains are linear in the velocity variation q_i, $i = 1,2$ and so are the constrains for the upper and lower bounds in (3.4).

3.1.2: HEADING ANGLE CHANGE

We will now examine the heading angle deviation problem in order to find applicable values of the heading angles of both aircrafts.
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 *pi.* In this section we show that it is possible to predict the existence of conflicts between the two aircraft based on those aircraft's initial configurations.
Each aircraft can maneuver only once with an instantaneous heading angle deviation and then we suppose that the *i-th* aircraft changes its

heading angle of a quantity *pi* that can be positive (left turn), negative (right turn) or null (no deviation).

The problem then is to find an admissible value of *pi* for each aircraft such that all conflicts are avoided, the new heading angle and direction of flight is then *θi* + *pi.*

In this section we formulate, through geometrical construction, conflict avoidance constraints that are linear in the unknowns P_i, $\forall i$ = l,...,n.

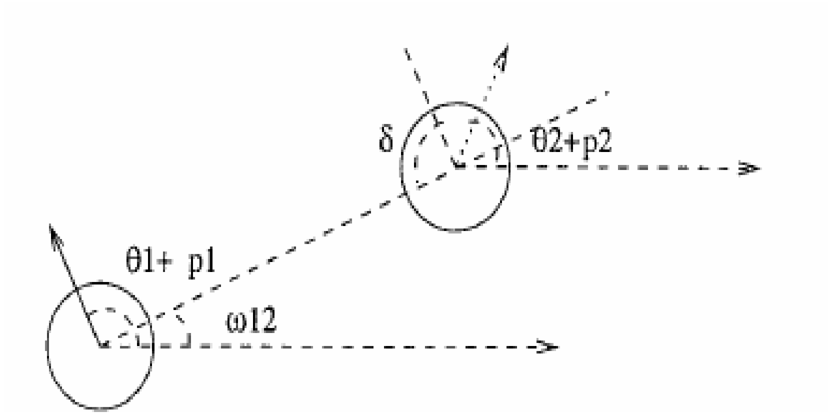


Figure3.4: *Case of two aircraft, if the heading angle of Aircraft 2 does not lie in the outlined sector of amplitude 6 then the trajectories do not intersect and no conflict will occur.*

3.2.1: Non-intersecting directions of motion

Consider the case when the geometric half-lines representing the extrapolated trajectories of the two aircraft do not intersect. Consider for example Figure 1: Aircraft 1 has heading angle $\theta_1 + p_1$. Assume the second aircraft is on the straight line forming an angle ω_{12} with the x-axis. If the heading angle ($\theta_2 + p_2$) of the second aircraft doesn't lie in the outlined sector of amplitude δ then the half lines obtained by projecting forward the motion of both aircraft do not intersect. The condition upon such a case occurs may be expressed easily via some inequality constraints. Let ω_{12} be the angle between the line that joins the aircraft and the horizontal axis, we have to consider all the possible cases of relative position and then we need to distinguish the case 1($0 < \omega_{12} < \pi$) from the
case 2 ($-\pi < \omega_{12} < 0$).
Furthermore building the constraints we obtain the quantity $g_1 = \theta_1 + p_1 - \omega_{12}$ and then we consider all the possible cases for g_1: If $g_1 \geq \pi$ or $g_1 \leq -\pi$ we have to shift the value g_1 of a quantity $-\pi$ or π respectively, so that the values that we consider lie in $[-\pi, \pi]$, no shift for the case $-\pi \leq g_1 \leq \pi$ is needed. Due to those possible cases we obtain three groups of constraints for each one of the two cases of $\omega_{1,2}$. Then the following conflict avoidance constraints, linear in p_1 and p_2, have been obtained by geometric construction:

1. Case: $0 \leq \omega_{12} \leq \pi$

$$\begin{cases} \omega_{12} - \pi - \theta_1 \leq p_1 \leq \omega_{12} - \theta_1 \\ \begin{cases} \theta_1 + p_1 - \theta_2 \leq p_2 \leq \pi \\ \text{or} \\ -\pi - \theta_2 \leq p_2 \leq \omega_{12} - \pi - \theta_2; \end{cases} \end{cases}$$
or
$$\begin{cases} \omega_{12} - \theta_1 \leq p_1 \leq \pi - \theta_1 \\ \omega_{12} - \pi - \theta_2 \leq p_2 \leq p_1 + \theta_1 - \theta_2 \end{cases}$$
or
$$\begin{cases} -\pi - \theta_1 \leq p_1 \leq \omega_{12} - \pi - \theta_1 \\ \begin{cases} \omega_{12} - \pi - \theta_2 \leq p_2 \leq \pi - \theta_2; \\ \text{or} \\ -\pi - \theta_2 \leq p_2 \leq p_1 + \theta_1 - \theta_2; \end{cases} \end{cases}$$

2. Case: $-\pi \leq \omega_{12} \leq 0$

$$\begin{cases} \omega_{12} - \theta_1 \leq p_1 \leq \omega_{12} + \pi - \theta_1 \\ \begin{cases} -\pi - \theta_2 \leq p_2 \leq p_1 + \theta_1 - \theta_2 \\ \text{or} \\ \omega_{12} + \pi - \theta_2 \leq p_2 \leq \pi - \theta_2; \end{cases} \end{cases}$$
or
$$\begin{cases} -\pi - \theta_1 \leq p_1 \leq \omega_{12} - \theta_1 \\ p_1 + \theta_1 - \theta_2 \leq p_2 \leq \pi + \omega_{12} - \theta_2 \end{cases}$$
or
$$\begin{cases} \omega_{12} + \pi - \theta_1 \leq p_1 \leq \pi - \theta_1 \\ \begin{cases} -\pi - \theta_2 \leq p_2 \leq \omega_{12} + \pi - \theta_2; \\ \text{or} \\ p_1 + \theta_1 - \theta_2 \leq p_2 \leq \pi - \theta_2; \end{cases} \end{cases}$$

Just one of the two groups of constraints will be included in the model as or-constraint depending on the sign of ω_{12}. In the general case of *n* aircraft, we have one of those group of or-constraints for each pair of aircraft (i, j), for i < j .

3.2.2: Intersecting directions of motion

For the other cases not included in the previous section we can refer to figure 2, and consider two aircraft x_1, y_1) and (x_2,y_2) with heading angles θ_1 and θ_2 respectively, we consider for simplicity $p_1= p_2 = 0$, the general equation will be expressed in the next section. We compute the amplitude (θ_1- θ_2) of the angle formed by the intersection of the aircraft flight directions and the amplitude((θ_1+ θ_2)/2) of the angle of his bisector (straight line *b)* with the x-axis . The bisector *b* is then a straight line that forms an angle ((θ_1+ θ_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), orthogonal to the bisector represents also the projection of one aircraft along the direction of motion of the other i.e. the two straight lines in this family that are tangent to the aircraft 1 localize a segment on the direction on motion of 2 (refer to Figure 3.3). We will refer to this segment as the *shadow* of aircraft 1 along the direction of 2. A conflict occurs if the aircraft 2 with his safe disc intersects the shadow generated by aircraft 1 or vice-versa since the angle m_{12} is symmetric in $\theta_1$and θ_2 .

The condition of non intersection of the shadows is equivalent to the following two conditions:

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

or

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

with $m_{12} = (\theta_1+ \theta_2+ \pi)/2$.

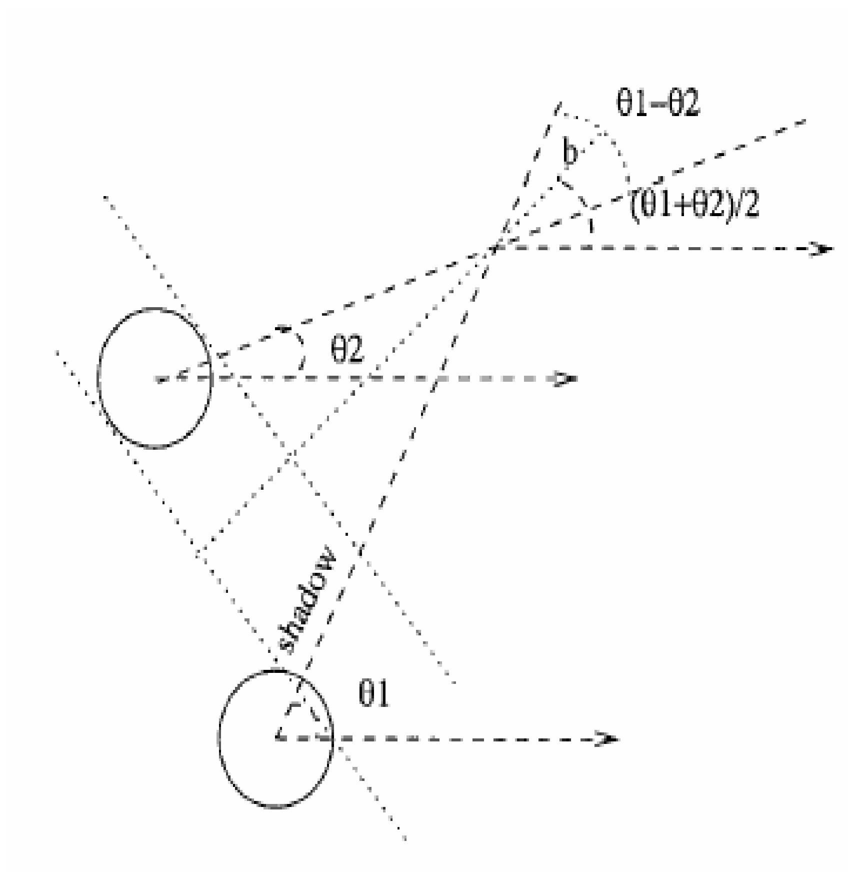


Figure 3.5: *Geometric construction for conflict avoidance constrains in the case of intersecting trajectories for the HAC problem. In this case aircraft 1intersect the shadowof aircraft 2, then a future conflict has been detected.2*

3.3: OVERALL APPROACH

Having shown the constraints that come from the velocity change problem (VC) and the heading angle problem (HAC) we proceed by combining the results of these two methods in an overall approach. So, we replace in these equations the heading angle *θi* which is constant in the velocity change problem, with the heading angle plus the heading angle deviation, thus *θi* + *pi* where *pi* is the angle deviation of the i-th i=1,2 aircraft from his originally scheduled flight plan. The final set of constrains for the two aircrafts we have discussed is used for simulation methods and is:

$$\frac{(u_1+q_1)\sin(\theta_1+p_1)-(u_2+q_2)\sin(\theta_2+p_2)}{(u_1+q_1)\cos(\theta_1+p_1)-(u_2+q_2)\cos(\theta_2+p_2)} \geq \tan(l_{12})$$

or

$$\frac{(u_1+q_1)\sin(\theta_1+p_1)-(u_2+q_2)\sin(\theta_2+p_2)}{(u_1+q_1)\cos(\theta_1+p_1)-(u_2+q_2)\cos(\theta_2+p_2)} \leq \tan(r_{12})$$

Further analysis of the above inequalities leads us to considering two cases:

1) $(u_1+q_1)\cos(\theta_1+p_1)-(u_2+q_2)\cos(\theta_2+p_2)<0$
2) $(u_1+q_1)\cos(\theta_1+p_1)-(u_2+q_2)\cos(\theta_2+p_2)>0$

where : u_1=velocity of aircraft 1

u_2 =velocity of aircraft 2

q_1=velocity deviation of aircraft 1

q_2=velocity deviation of aircraft 2

θ_1=heading angle of aircraft 1

θ_2=heading angle of aircraft 2

p_1=heading angle deviation of aircraft 1

p_2=heading angle deviation of aircraft 2

We also have to set the following equalities in order to express in a more obvious way the final set of constraints.

$$h_1=\tan(l_{12})\cos(\theta_1+p_1)-\sin(\theta_1+p_1)$$
$$h_2=\tan(l_{12})\cos(\theta_2+p_2)-\sin(\theta_2+p_2)$$
$$\kappa_1=\tan(r_{12})\cos(\theta_1+p_1)-\sin(\theta_1+p_1)$$
$$\kappa_2=\tan(r_{12})\cos(\theta_2+p_2)-\sin(\theta_2+p_2)$$

We are now ready to write the two cases constraints for the overall approach

3.3.1 : *$(u_1+q_1)\cos(\theta_1+p_1)-(u_2+q_2)\cos(\theta_2+p_2)<0$*

$$\begin{cases} \cos(\theta_1+p_1)\, q_1 - \cos(\theta_2+p_2)\, q_2 \leq -u_1 \cos(\theta_1+p_1) + u_2 \cos(\theta_2+p_2) \\ -h_1 q_1 + h_2 q_2 \leq u_1 h_1 - u2 h_2 \end{cases}$$

or

$$\begin{cases} \cos(\theta_1+p_1)\, q_1 - \cos(\theta_2+p_2)\, q_2 \leq -u_1 \cos(\theta_1+p_1) + u_2 \cos(\theta_2+p_2) \\ k_1 q_1 - k_2 q_2 \leq -u_1 k_1 + u_2 k_2 \end{cases}$$

3.3.2: ***$(u_1+q_1)\cos(\theta_1+p_1) - (u_2+q_2)\cos(\theta_2+p_2) > 0$***

$$\begin{cases} -\cos(\theta_1+p_1)\, q_1 + \cos(\theta_2+p_2)\, q_2 \leq u_1 \cos(\theta_1+p_1) - u_2 \cos(\theta_2+p_2) \\ h_1 q_1 - h_2 q_2 \leq -u_1 h_1 + u2 h_2 \end{cases}$$

or

$$\begin{cases} -\cos(\theta_1+p_1)\, q_1 + \cos(\theta_2+p_2)\, q_2 \leq u_1 \cos(\theta_1+p_1) - u_2 \cos(\theta_2+p_2) \\ -k_1 q_1 + k_2 q_2 \leq u_1 k_1 - u_2 k_2 \end{cases}$$

Let us examine now the first sub case, which is the following two inequalities :

$$\cos(\theta_1+p_1)\, q_1 - \cos(\theta_2+p_2)\, q_2 \leq - u_1 \cos(\theta_1+p_1) + u_2 \cos(\theta_2+p_2) \quad (1)$$

$$-h_1q_1+h_2q_2 \leq u_1h_1-u_2h_2 \quad (2)$$

we are aware of

$$h_1=\tan(l_{12}) \cos(\theta_1+p_1) - \sin(\theta_1+ p_1)$$

$$h_2=\tan(l_{12}) \cos(\theta_2+ p_2)- \sin(\theta_2+ p_2)$$

we replace the two equations at (2) inequality

(2) => $- [\tan(l_{12}) \cos(\theta_1+p_1)- \sin(\theta_1+p_1)]\, q_1 + [\tan(l_{12}) \cos(\theta_2+ p_2)- \sin(\theta_2+ p_2)]\, q_2 \leq$

$[\tan(l_{12}) \cos(\theta_1+p_1)- \sin(\theta_1+p_1)]\, u_1 - [\tan(l_{12}) \cos(\theta_2+ p_2)- \sin(\theta_2+ p_2)]\, u_2$

=> $- q_1 \tan(l_{12}) \cos(\theta_1+p_1) + q_1 \sin(\theta_1+p_1) + q_2 \tan(l_{12}) \cos(\theta_2+ p_2) - q_2 \sin(\theta_2+ p_2) \leq$

$u_1 \tan(l_{12}) \cos(\theta_1+p_1) - u_1 \sin(\theta_1+p_1) - u_2 \tan(l_{12}) \cos(\theta_2+ p_2) + u_2 \sin(\theta_2+ p_2)$

⇨ $q_1 \sin(\theta_1+p_1) + u_1 \sin(\theta_1+p_1) - q_2 \sin(\theta_2+ p_2) - u_2 \sin(\theta_2+ p_2) \leq$

$u_1 \tan(l_{12}) \cos(\theta_1+p_1) + q_1 \tan(l_{12}) \cos(\theta_1+p_1) - u_2 \tan(l_{12}) \cos(\theta_2+ p_2) - q_2 \tan(l_{12}) \cos(\theta_2+ p_2)$

⇨ $(u_1 + q_1) \sin(\theta_1+p_1) -(u_2 + q_2) \sin(\theta_2+p_2) \leq$

$\tan(l_{12}) [(u_1 + q_1) \cos(\theta_1+p_1) -(u_2 + q_2) \cos(\theta_2+ p_2)]$

It is obvious that we conclude again at the first inequality we used for the overall approach. Similarly we work for the other three pairs of inequalities to end up at the same assumption as before. In this way we actually managed to define the constraints we will use for our cost function, the metric we are going to optimize, which is going to be presented at the consecution. Gazing through the ATC literature[2-10], we considered that one suitable function would be that of the sum of the absolute prices of both aircraft's velocity and heading angle deviations. More specifically this function would be of the form**:**

$$\mathbf{f(x)=|x_1| +| x_2|+| x_3|+| x_4|}$$

With this property, we are giving our optimization software a further assistance towards optimization.

In conclusion, we note that our problem as it has been transformed is now a non-linear problem, soon to be transformed in a mixed-integer non-linear problem in the later section. As a logical result we will have to expect larger executional times and slower performance than before, which may have an impact in real time situations, but we have to consider that our approach is much more realistic than before since we allow both velocity and heading angle deviations.

3.4: Mixed integer non-linear optimization

3.4.1: Overview of MINLP Algorithms

1. Generalized Benders Decomposition

In the pioneering work of Geoffrion on the Generalized Benders Decomposition **GBD** two sequences of updated upper (nonincreasing) and lower (nondecreasing) bounds are created that converge in a finite number of iterations. The upper bounds correspond to solving subproblems in the x variables by fixing the y variables, while the lower bounds are based on duality theory.

2. Branch and Bound

The Branch and Bound **BB** approaches start by solving the continuous relaxation of the MINLP and subsequently perform an implicit enumeration where a subset of the 0-1 variables is fixed at each node.

3. Outer Approximation

The Outer Approximation **OA** addresses problems with nonlinear inequalities and creates sequences of upper and lower bounds as the **GBD** , but it has the distinct feature of using primal information.

4. Feasibility Approach

The feasibility approach **FA** rounds the relaxed NLP solution to an integer solution with the least local degradation by successively forcing the superbasic variables to become nonbasic **.**

3.4.2: Problem formulation

We are now faced with a non-linear optimization problem awaiting to be solved with efficiency and a relative speed in computational time. Every optimization package requires that the constraints used must be of the form of and-constraints which means that they have to be satisfied simultaneously.

3.4.3: Writing or-constraints as mixed-integer programming constraints

Consider now an example of or-groups of constraints similar to the conflict avoidance constraints described earlier :

$C_1 \leq 0$

and

$C_2 \leq 0$

or

$C_3 \leq 0$

and

$C_4 \leq 0$

or

$C_5 \leq 0$

and

$$C_6 \leq 0$$

or

$$C_7 \leq 0$$

and

$$C_8 \leq 0$$

where the terms C_i, $i = 1,...,7$ are non-linear expressions in the decision variables (heading angle and velocity change deviations). The way to transform these or-constraint into more convenient and-constraint is to introduce Boolean variables. Let y_k with $k = 1,2,3$, be a binary number that takes value 1 when one of the or-constraint is active and zero otherwise. For example

$y_1 = 1$ if constraints $C_1$and C_2 are active $y_1= 0$ otherwise. Let *M* be a large arbitrary number, then the previous set of constraint is equivalent to

$$C_1\text{-}M\ y_1 \leq 0$$
$$C_2\text{-}M\ y_1 \leq 0$$
$$C_3\text{-}M\ y_2 \leq 0$$
$$C_4\text{-}M\ y_2 \leq 0$$
$$C_5\text{-}M\ y_3 \leq 0$$
$$C_6\text{-}M\ y_3 \leq 0$$
$$C_7\text{-}M\ y_4 \leq 0$$
$$C_8\text{-}M\ y_4 \leq 0$$
$$y_1+ y_2+ y_3+ y_4 \leq 2$$

The last constraint indicates that at least one of the four groups of and-constraints must be verified.

So the final inequality constraints for our problem have the following form(*function g(x,y)*)

$g_1(x,y)=(u_1+x_1)\cos(\theta_1+x_2) - (u_2+x_3)\cos(\theta_2+x_4) - My_1$

$g_2(x,y)=-h_1(x_1+u_1) + h_2(u_2+x_3) - My_1$

$g_3(x,y)=(u_1+x_1)\cos(\theta_1+x_2) - (u_2+x_3)\cos(\theta_2+x_4) - My_2$

$g_4(x,y)=k_1(x_1+u_1) - k_2(u_2+x_3) - My_2$

$g_5(x,y)=-(u_1+x_1)\cos(\theta_1+x_2) + (u_2+x_3)\cos(\theta_2+x_4) - My_3$

$g_6(x,y)= h_1(x_1+u_1) - h_2(u_2+x_3) - My_3$

$g_7(x,y)=-(u_1+x_1)\cos(\theta_1+x_2) + (u_2+x_3)\cos(\theta_2+x_4) - My_4$

$g_8(x,y)=-k_1(x_1+u_1) + k_2(u_2+x_3) - My_4$

$g_9(x,y)=y_1+y_2+y_3+y_4 \leq 2$

Mathematical Description

The general MINLP formulation for our problem can be stated as

$$\min_{x,y} f(x,y)$$

$$h(x,y)=0$$

$$g(x,y) \leq 0$$

$$x \in X \subseteq R^n$$

$$y \in Y=\{0,1\}^q$$

In our case we must notice that we have no equality constraints so the case of constraints h(x,y)=0 does not exist.
Our cost function as we mentioned before is the following one:

$$\mathbf{f(x)=|x_1| +| x_2|+| x_3|+| x_4|}$$

where $\mathbf{x_{1=}}$ q_1=velocity deviation of aircraft 1
$\mathbf{x_{2=}}$ p_1=heading angle deviation of aircraft 1
$\mathbf{x_{3=}}$ q_2=velocity deviation of aircraft 2
$\mathbf{x_{4=}}$ p_2=heading angle deviation of aircraft 2

and

$$x \in R^+_0 \times \{ [0,2\pi]\} \times R^+_0 \times \{ [0,2\pi]\}$$

We must also define the values of *n* and *q for x set and y set which are n=4 and q=4 as it is obvious from the binary usage of y.*
We are now ready to fulfill the three conditions of Generalized Bender's Decomposition in order to be able to use this method for obtaining the desired goal.

Condition 1

From the observation of set X we used ($x \in R^+_0 x\{[0,2\pi]\} x R^+_0 x\{[0,2\pi]\}$) it is obvious that the specific set of values x for our cost function is non empty and also its is convex(Appendix)
In order to prove the convexity of our cost function we will use directly the definition for a convex function.

Definition (convex function) *Let S be a convex subset or R^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, $x_2 \in$ 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.

Our cost function have the following form

$$\mathbf{f(x)=|x_1| +| x_2|+| x_3|+| x_4|}$$

We need to prove that

$$f[(1-\lambda)\, x_1+\lambda\, x_2] \leq (1-\lambda)f(x_1)+\lambda\, f(x_2) \qquad \mathbf{(1)}$$

$$f[(1-\lambda)\, x_1+\lambda\, x_2]=|(1-\lambda)x_{11}+\lambda x_{12}|+|(1-\lambda)x_{21}+\lambda x_{22}|+|(1-\lambda)x_{31}+\lambda x_{32}|+ |(1-\lambda)x_{41}+\lambda x_{42}| \qquad \mathbf{(2)}$$

We are going now to analyze the second part of **(1)** inequality

$f(x_1)=|x_{11}|+|x_{21}|+|x_{31}|+|x_{41}|$

and $f(x_2)=|x_{12}|+|x_{22}|+|x_{32}|+|x_{42}|$

So if we use the equalities we just deployed to the second part of **(1)** inequality we have the following results

$(1-\lambda)f(x_1)+\lambda\, f(x_2)=(1-\lambda)[|x_{11}|+|x_{21}|+|x_{31}|+|x_{41}|\,] + \lambda\, [|x_{12}|+|x_{22}|+|x_{32}|+|x_{42}|\,]$
$= \underline{(1-\lambda)\, |x_{11}| + \lambda\, [|x_{12}|}+ \underline{(1-\lambda)\, |x_{21}| + \lambda\, |x_{22}|}+ \underline{(1-\lambda)\, |x_{31}| + \lambda\, |x_{32}|} +$
$\underline{(1-\lambda)\, |x_{41}| + \lambda\, |x_{42}|}$

We separately examine now the above equation using each one of the four underlined parts.

It is of common knowledge that $|x+y| \leq |x| + |y|$

So we come to the following conclusions

$|(1-\lambda)x_{11}+\lambda x_{12}|\leq (1-\lambda)[|x_{11}|+\lambda|x_{12}|$ and

$|(1-\lambda)x_{21}+\lambda x_{22}|\leq(1-\lambda)[|x_{21}|+\lambda|x_{22}|$ and

$|(1-\lambda)x_{31}+\lambda x_{32}|\leq(1-\lambda)[|x_{31}|+\lambda|x_{32}|$ and

$|(1-\lambda)x_{41}+\lambda x_{42}|\leq (1-\lambda)[|x_{41}|+\lambda|x_{42}|$

Summarizing the previous four inequalities we reach at the following inequality

$|(1-\lambda)x_{11}+\lambda x_{12}|+|(1-\lambda)x_{21}+\lambda x_{22}|+|(1-\lambda)x_{31}+\lambda x_{32}|+ |(1-\lambda)x_{41}+\lambda x_{42}| \leq$
$(1-\lambda) |x_{11}| + \lambda |x_{12}|+ (1-\lambda) |x_{21}| + \lambda |x_{22}|+ (1-\lambda) |x_{31}| + \lambda |x_{32}| + (1-\lambda) |x_{41}|$
$+ \lambda |x_{42}|$

$\Rightarrow f[(1-\lambda) x_1+\lambda x_2] \leq (1-\lambda)f(x_1)+\lambda f(x_2)$

So , our cost function qualifies *Jensen's Inequality* which means that f(x,y) is convex function.

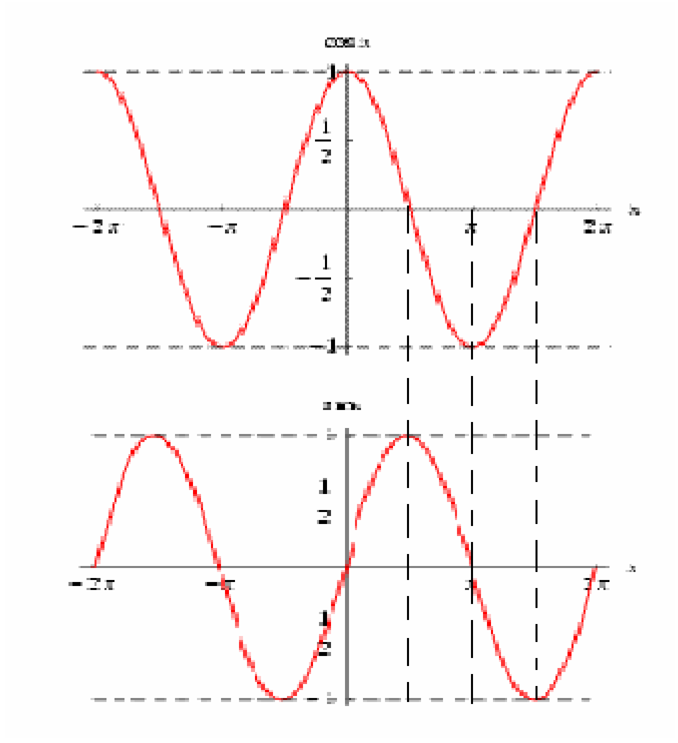
In our case we do not have equality constraints [h(x,y)] so we will automatically examine the inequality constraints .

g(x,y) has the following form $g:R^4 x R^4 \rightarrow R^9$

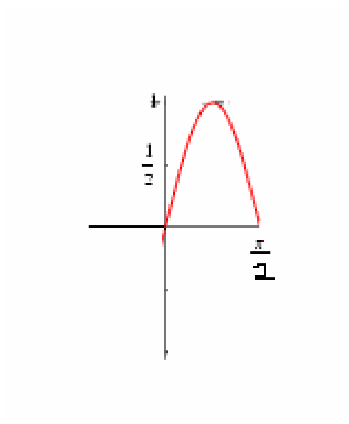
$$g(x,y)=\begin{bmatrix} g_1(x,y) \\ g_2(x,y) \\ g_3(x,y) \\ g_4(x,y) \\ g_5(x,y) \\ g_6(x,y) \\ g_7(x,y) \\ g_8(x,y) \\ g_9(x,y) \end{bmatrix}$$

We must notice here that because of the form of g(x,y) it was quite difficult to find the proof for the convexity of the specific function.We tride to use Jensen's Inequality but we could not make it work for this kind of a function. The second effort was with Hessian matrices but the result was the same as before, inadequate results.

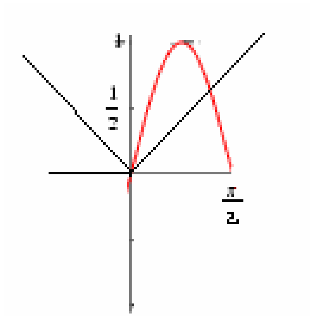
So we come to a more "simple" method. We analyzed each part of g(x,y) separately. We noticed that $g_1(x,y)$, $g_3(x,y)$, $g_5(x,y)$, $g_7(x,y)$ are graphically presented in the 2-dimension level with cosines while $g_2(x,y)$, $g_4(x,y)$, $g_6(x,y)$, $g_8(x,y)$ are presented with tangents. We analyzed more the tangents so we obviously concluded to combine the graphs of cosine and sine in order to find the convexity of g(x,y) and as its is shown in the graphs below, which shows that g(x,y) is convex when $x \in [\pi, 3/2\,\pi]$



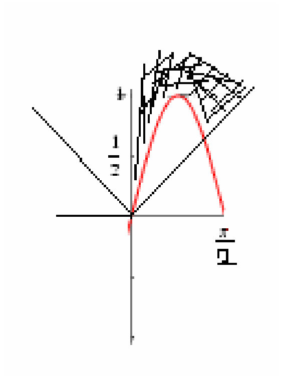
We now check all possible angles , starting from [0, 3π/2] , and continuing with [π,2π] .These are the angles aircraft enter the alert sphere.We observe that at the closed interval [0, π/2] the cosine and the sine are of concave form. This prevents us from using the Bender's Decomposition method in order to achieve a local or even a global minimum for our cost function f(x, y). The linear combination of a cosine function and a sine function at the closed interval [0, π/2] has approximately the following form :



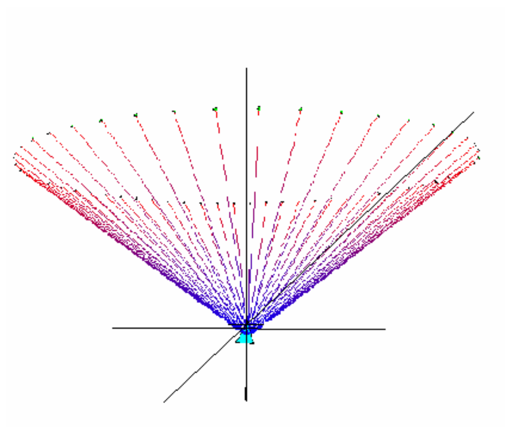
The plotting of our cost function in the graph above, is shown below



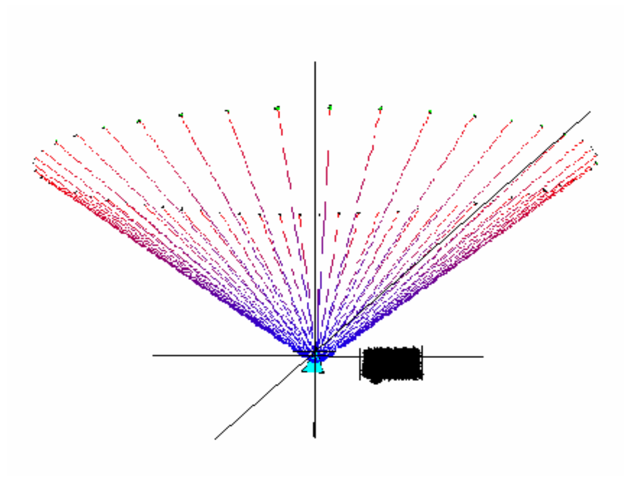
Due to the fact that g(x) ≤0 we may find an global minimum for our cost function in the shaded interval shown below:



This graph is made for one of the four independent of the cost function . Considering x∈[0,π/2] and using all four variables we get a 4-D cone ,open from above and it's boundaries around zero are included in it , since it is closed.



Because of the closedness property the minimum for the cost function is attendable.



The same assumptions hold for the other two intervals [$\pi/2,\pi$] and [$3\pi/2, 2\pi$] .We also observe that there is no convexity on the inequality constraints, but it is easily shown that the existence of closed sets and the form of our cost function (cone) guarantees minimum attainability.

Before starting with the proof of the second condition we add that all the convexity conditions were taken under the assumption that $y \in Y=\{0,1\}^4$ and the values of the boolean variable y are fixed .

Condition 2

The set $Zy=\{z \in R^4 : g(x,y) \leq z$ for some $x \in X \}$ is closed for each fixed $y \in Y$. In other words we must prove that there is a set Zy such that its values are actually the upper bounds of convex function g(x,y) for some $x \in X$ when we take consideration of fixed $y \in Y$. We must notice here that the second condition for the Generalized Bender's Decomposition is not stringent. It can be satisfied also if we satisfy the two following conditions:

i. x is bounded and closed which is shown before in order to prove the convexity of g(x,y), we had to choose a bounded and closed set of values for x

ii. g(x,y) is continuous on x for each fixed $y \in Y$ which is also a fact if we consider that we already have proved that g(x,y) is convex and from the same diagram is obvious that it is also continuous

Condition 3

For eached fixed $y \in Y \cap V$ where

$$V=\{y:g(x,y) \leq 0 \text{ for some } x \in X\}$$

one of the two following conditions hold:

i. our problem has a finite solution and has an optimal multiplier vector for the equalities and inequalities

ii. our problem is unbounded that is , its objective function value goes to –00

We must notice here that the specific condition is satisfied if a first-order constraint qualification holds for our problem after fixing $y \in Y \cap V$.
We choose to satisfy the Slater first-order Constraint.

Slater Constraint Qualification

The constraints $g_j(\bar{x})$ for $j \in J$ are pseudo-convex at $\bar{x}$.The constraints $h_i(\bar{x})$ for i=1,2,... , m are quasi-convex and quasi-concave. The gradients $\nabla h_i(\bar{x})$ for i=1,2,... , m are linearly independent .

In order to satisfy this constraint we actually have to prove that $g_j(x)$ for $j \in J$ is pseudo-convex cause we have no equality constraints in our problem. The proof is easy cause we have shown in **condition 1 that** g(x) is already convex so its is commonly known that it is pseudo-convex.

3.5:PROCEDURE OF MINIMIZATION

3.5.1: Basic Idea

The basic idea in Generalized Benders Decomposition **GBD** is the generation, at each iteration, of an upper bound and a lower bound on the sought solution of the MINLP model. The upper bound results from the primal problem, while the lower bound results from the master problem. The primal problem corresponds to initial problem with fixed
y-variables (i.e., it is in the x-space only), and its solution provides information about the upper bound and the Lagrange multipliers associated with the equality and the inequality constraints. The master problem is derived via nonlinear duality theory, makes use of the Lagrange multipliers obtained in the primal problem, and its solution provides information about the lower bound, as well as the next set of fixed y-variables to be used subsequently in the primal problem. As the iterations proceed, it is shown that the sequence of updated upper bounds is nonincreasing, the sequence of lower bounds is nondecreasing and that the sequences converge in a finite number of iterations.

3.5.2:Theoretical Development

This section presents the theoretical development of the Generalized Benders Decomposition **GBD.** The primal problem is analyzed first for the feasible and infeasible cases. Subsequently, the theoretical analysis for the derivation of the master problem is presented.

3.5.2.1:The Primal Problem

The primal problem results from fixing the y-variables to a particular 0-1 combination, which we denote as y^k where k stands for the iteration counter. The formulation of the primal problem $P(y^k)$, at iteration k is

$$\begin{aligned} &\operatorname{Min}_x f(x,y^k) \\ &\text{s.t} \quad h(x,y^k)=0 \\ &\qquad g(x,y^k) \leq 0 \\ &\qquad x \in X \subseteq R^n \end{aligned}$$

Remark 1 Note that due to conditions C_1 and C_3(i), the solution of the primal problem $P(y^k)$ is its global solution.

We will distinguish the two cases of (i) feasible primal, and (ii) infeasible primal, and describe the analysis for each case separately.

Case (i): Feasible Primal

If the primal problem at iteration k is feasible, then its solution provides information on x^k, $f(x^k, y^k)$, which is the upper bound, and the optimal multiplier vectors λ^k, μ^k for the equality and the inequality constraints. Subsequently, using this information we can formulate the Lagrange function as

$$L(x,y, \lambda^k, \mu^k)=f(x,y) +\lambda^{kT} h(x,y)+ \mu^{kT} g(x,y)$$

In our case we do have any equality constraints so the Lagrange function is

$$L(x,y, \lambda^k, \mu^k)=f(x,y) + \mu^{kT} g(x,y)$$

Case (ii): Infeasible Primal

If the primal is detected by the NLP solver to be infeasible, then we consider its constraints

$$g(x,y^k) \leq 0$$
$$x \in X \subseteq R^n$$

where the set X, for instance, consists of lower and upper bounds on the x variables.

To identify a feasible point we can minimize an L_1 or L_∞ sum of constraint violations.

L_1-minimization

L_1-minimization can be formulated as

$$\text{Min} \quad \sum_{i=1}^{p} a_i$$

$$\text{s.t} \quad g(x,y^k) \le a_i\,, \quad i=1,2,\ldots p$$
$$a_i \geq 0 \quad , \; i=1,2,\ldots p$$

Note that if $\sum^{p}_{i=1} a_i = 0$, then a feasible point has been determined.

Also note that by defining as

$$a^+ = \max(0,a) \text{ and}$$
$$g^+_i(x,y^k) = \max[0, g_i(x,y^k)]$$

the L_1-minimization problem is stated as

$$\text{Min} \quad \sum_{i=1}^{p} g^+_I$$

Similarly we have the L_∞-minimization problem

$$\min_{x \in X} \max_{1,2,..p} g^{+}_{i}(x,y^{k})$$

3.5.2.2:The Master Problem

The derivation of the master problem in the **GBD** makes use of nonlinear duality theory and is characterized by the following three key ideas:

i. Projection of problem(2.4.1)onto the y-space;
ii. Dual representation of V;
iii. Dual representation of the projection of problem (2.4.1)on the y-space;

In the sequel, the theoretical analysis involved in these three key ideas is presented.

i. Projection onto the y-space

Problem (2.4.1)can be written as

$$\min_y \inf_x f(x,y)$$

$$\text{s.t } g(x,y) \leq 0$$
$$x \in X$$

$$y \in Y=\{0,1\}^4$$

where the min operator has been written separately for x and y. Note that it is infimum with respect to x since for given y the inner problem may be unbounded.Let us define u(y) as

$$u(y)=\inf_x \ f(x,y)$$
$$\text{s.t } g(x,y) \leq 0$$
$$x \in X$$

We need also to define the set V as

$$V=\{y:g(x,y) \leq 0 \text{ for some } x \in X \}$$

ii. Dual Representation of V

The dual representation of **V** will be invoked in terms of the intersection of a collection of regions that contain it , and its is described in the following theorem of Geoffrion.
Assuming conditions C_1 AND C_2 a point $y \in Y$ belongs also to the set V if and only if it satisfies the system:

$$\mathit{Inf}\ L'(x,y,\lambda',\mu') \ , \ \lambda',\mu' \in \Lambda$$

$$\Lambda=\{\lambda' \in R^m \ , \ \mu' \in R^p : \mu' \geqq 0 \ , \Sigma^p \mu'=1 \}$$

i=1

iii. **Dual Representation of** u(y)

The dual representation of u(y) will be in terms of the point wise infimum of a collection of functions that support it and it is described in the following theorem.

$$u(y)=\inf_x \underset{x\in X}{f(x,y)\ \text{s.t}\ g(x,y)\leq 0} = [\sup_{\lambda,\mu\geq 0}\ \inf_{x\in X} L(x,y,\lambda,\mu)],\quad y\in Y\cap V$$

where $L(x,y,\lambda,\mu)=f(x,y)+\mu^T g(x,y)$

Remark : Note that the master problem is equivalent to (2.4.1).It involves however , an infinite number of constraints , and hence we will need to consider a relaxation of the master (e.g., by dropping a number of constraints) which will represent a lower bound on the original problem. Note also that the master problem features an outer optimization problem with respect to $y \in Y$ and inner optimization problems with respect to x which are in fact parametric in y.It is this outer-inner nature that makes the solution of even a relaxed master problem difficult.

3.6:Code Development using GAMS

Following the above steps we give an example of the procedure using given initial parameters .The form of pseudocode we will use is given below:

The Benders' Decomposition algorithm can be stated as:

```
{initialization}
y := initial feasible integer solution
LB := -$\infty$  UB := $\infty$
while UB - LB > $\epsilon$ do
{solve subproblem}
 $\min_u\{f^T y' + (b - By')^T u | A^T u \leq c, u \geq 0\}$
 if Unbounded then
Get unbounded ray u
Add cut (b - By)T u _ 0 to master problem
else
Get extreme point u
Add cut z _ fT y + (b - By)T u to master problem
UB := min{UB, fT y + (b - By)T u} end if
{solve master problem} miny{z|cuts, y 2 Y } LB := z
end while
```

The subproblem is a dual LP problem, and the master problem is a pure IP problem (no continuous variables are involved). Benders' Decomposition for MIP is of special interest when the Benders' subproblem and the relaxed master problem are easy to solve, while the original problem is not.

3.7:Simulation and Results

The methodology that we use in our code is the one as it is given before, we actually use a lower and an upper bound for each one of the four variables **$x_{1=}$** q_1=velocity deviation of aircraft 1

$x_{2=}$ p_1=heading angle deviation of aircraft 1

$x_{3=}$ q_2=velocity deviation of aircraft 2

$x_{4=}$ p_2=heading angle deviation of aircraft 2

and we calcutate in how many iterations our method GBD can approximate the desired value. We must notice that the desired values are already known from previous researches on the subject of air traffic management and are taken as granted.

$x_{1=}$ q_1=velocity deviation of aircraft 1

The desired value is granted and it is $x_{1=}$ q_1= 0.210

Table 3. 6:Velocity deviation of aircraft 1.

Lower bound	Upper bound	Iterations
0.190	**0.230**	**1**
0.192	**0.230**	**2**
0.193	**0.228**	**3**
0.195	**0.225**	**4**
0.196	**0.224**	**5**
0.199	**0.219**	**6**
0.204	**0.218**	**7**
0.204	**0.215**	**8**
0.207	**0.211**	**9**
0.209	**0.210**	**10**
0.209	**0.210**	**11**

We notice that we reach the desired goal in eleven iterations.

$x_{2=}$ p_1=heading angle deviation of aircraft 1

The desired value is granted and it is $x_{2=}$ p_1= 0.071

Table 3. 7:Heading angle deviation of aircraft 1.

Lower bound	Upper bound	Iterations
0.060	**0.080**	**1**
0.060	**0.080**	**2**
0.062	**0.079**	**3**
0.063	**0.077**	**4**
0.065	**0.76**	**5**
0.067	**0.073**	**6**
0.069	**0.071**	**7**
0.070	**0.071**	**8**
0.071	**0.071**	**9**

We notice that we reach the desired goal in nine iterations.

$x_{3=}$ q_2=velocity deviation of aircraft 2

The desired value is granted and it is $x_{3=}$ q_2= 0.126

Table 3. 8:Velocity deviation of aircraft 2.

Lower bound	Upper bound	Iterations
0.110	**0.140**	**1**
0.113	**0.139**	**2**
0.114	**0.136**	**3**
0.115	**0.134**	**4**
0.115	**0.133**	**5**
0.117	**0.131**	**6**
0.117	**0.130**	**7**
0.118	**0.129**	**8**
0.121	**0.128**	**9**
0.122	**0.127**	**10**
0.124	**0.126**	**11**
0.125	**0.126**	**12**

We notice that we reach the desired goal in twelve iterations.

$x_{4=}$ p_2=heading angle deviation of aircraft 2

The desired value is granted and it is $x_{4=}$ p_2= 0.096

Table 3. 9:Heading angle deviation of aircraft 2.

Lower bound	Upper bound	Iterations
0.080	**0.110**	**1**
0.081	**0.108**	**2**
0.083	**0.106**	**3**
0.083	**0.103**	**4**
0.084	**0.101**	**5**
0.087	**0.100**	**6**
0.090	**0.099**	**7**
0.094	**0.097**	**8**
0.095	**0.096**	**9**
0.096	**0.096**	**10**

We notice that we reach the desired goal in ten iterations.

3.7.1:Conclusion

We conclude the previous results of the four variables we use in our cost function

Table 3. 10: Results of the four variables.

variables	iterations	goal
$x_{1=}$ q_1=velocity deviation of aircraft 1	**11**	**Achieved (nearly)**
$X_{2=}$ p_1=heading angle deviation of aircraft 1	**9**	**Achieved**
$X_{3=}$ q_2=velocity deviation of aircraft 2	**12**	**Achieved (nearly)**
$x_{4=}$ p_2=heading angle deviation of aircraft 2	**10**	**Achieved**

As we can notice the iterations required for achieving the desired goal are ranged between 9-12 which is a standard range for calculations of optimization using methods such as Generalized Bender's Decomposition but also Outer Approximation which is the most common method that is used for solving optimization problems. The calculation time was for x_1 586 secs at a Pentium 333 Mhz which is an important amount of time considering the amount of data the processor had to deal with.

Fortunately the convexity of the inequalities (constraints) provided us with excellent quality for the lower bounds .Also it gave us the capability of fastening the pase at the first part of Bender's Decomposition (primal problem) in order to achieve the bounds we have set.

Appendix A

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 (**$x_{1=}$** q_1=velocity deviation of aircraft 1) since the rest of the codes use a very similar approach and only the shines of cosines and sines and some other minor details are different.

Gams Code for velocity deviation of aircraft 1

```
Set i ”aircraft” /1* 2/;
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 /;
*X-coordinates in Km
Parameters x(i) / 1 108 2 54 /;
*Y-coordinates in Km
Parameters y(i) / 1 0 2 93.531 3 -9/;
*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 /;
*minimum safe distance in Km
```

```
msd=5.4;
*Big M
M=50;
*Consider only pairs (i,j) where i6=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)6=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 ), u(i);
Binary variable B1(i),B2(i),B3(i);
Lb = - 00

Up= + 00

while UB - LB > 0.520 do

*solve subproblem

minimize ( cos (q_i)alpha(i,j)-cos (q_j) alpha(i,j))

u(i)=0 ;

minimize ( cos (q_i)alpha(i,j)-cos (q_j) alpha(i,j)/sin (q_i)alpha(i,j)- sin (q_j)

alpha(i,j));

u(i)<0;

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;
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;
then
Ub= (min,*);

min (((u(i)+q(i))*cos(theta(i)+p(i))(u(j)+q(j))*cos(theta(j)+p(j))-
M*B1(i))+ ((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)))
else
z(i,j)..z(i,j)=tang(15.66-u(i))-tang(u(i)-14.4);
Lb=z(i,i)-z(j,j);
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, 2) 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 as well as the other parameters used in the VC and HAC model, exactly as described in the corresponding sections. We also define the constantparameters of circle radius and distance of final configuration point from exit point.
Next we define the value of the upper bound of our cost function and we solve the subproblem.
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 *!ij ;Aij ; lij* and *rij* 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.
We valuate the lower bound in case the cut of the upper bound is not feasible for the first part of the code.
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 *velocity1-velocity3* and *angle1-angle2* impose some bounds on the velocities and heading angles, while *A1-A6* are the main constraints. The program concludes with thecommands 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.

Bibliography

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

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

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

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

[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 Trafficb 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] *GAMS User Guide, www.gams.de, GAMS IDE 2.0.20.0. 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] *Garmin GPS Website*. *www.garmin.com.*