Alma Mater Studiorum University of Bologna

Ph.D. in Control System Engineering and Operational Research

MAT/09

XXIII Cycle

Algorithms for UAS insertion in civil air space

Carlo Alfredo Persiani

Coordinator
Prof. Paolo Toth

Relators
Prof. Paolo Toth
Prof. Francesca De Crescenzo

Final exam 2011

Acknowledgments

To my family and to my wife.

Bologna, 2011

Carlo Alfredo Persiani.

Keywords

Combinatorial Optimisation, Integer Programming, Heuristic algorithms ,Travelling Salesman Problem, Time Dependence, Unmanned Air Vehicle, Air Traffic Management .

Abstract

One of the most interesting challenge of the next years will be the Air Space Systems automation. This process will involve different aspects as the Air Traffic Management, the Aircrafts and Airport Operations and the Guidance and Navigation Systems. The use of UAS (Uninhabited Aerial System) for civil mission will be one of the most important steps in this automation process. In civil air space, Air Traffic Controllers (ATC) manage the air traffic ensuring that a minimum separation between the controlled aircrafts is always provided. For this purpose ATCs use several operative avoidance techniques like holding patterns or rerouting. The use of UAS in these context will require the definition of strategies for a common management of piloted and piloted air traffic that allow the UAS to self separate. As a first employment in civil air space we consider a UAS surveillance mission that consists in departing from a ground base, taking pictures over a set of mission targets and coming back to the same ground base. During all mission a set of piloted aircrafts fly in the same airspace and thus the UAS has to self separate using the ATC avoidance as anticipated. We consider two objective, the first consists in the minimization of the air traffic impact over the mission, the second consists in the minimization of the impact of the mission over the air traffic. A particular version of the well known Travelling Salesman Problem (TSP) called Time-Dependant-TSP has been studied to deal with traffic problems in big urban areas. Its basic idea consists in a cost of the route between two clients depending on the period of the day in which it is crossed. Our thesis supports that such idea can be applied to the air traffic too using a convenient time horizon compatible with aircrafts operations. The cost of a UAS sub-route will depend on the air traffic that it will meet starting such route in a specific moment and consequently on the avoidance maneuver that it will use to avoid that conflict. The conflict avoidance is a topic that has been hardly developed in past years using different approaches. In this thesis we purpose a new approach based on the use of ATC operative techniques that makes it possible both to model the UAS problem using a TDTSP framework both to use an Air Traffic Management perspective. Starting from this kind of mission, the problem of the UAS insertion in civil air space is formalized as the UAS Routing Problem (URP). For this reason we introduce a new structure called Conflict Graph that makes it possible to model the avoidance maneuvers and to define the arc cost function of the departing time. Two Integer Linear Programming formulations of the problem are proposed. The first is based on a TDTSP formulation that, unfortunately, is weaker then the TSP formulation. Thus a new formulation based on a TSP variation that uses specific penalty to model the holdings is proposed. Different algorithms are presented: exact algorithms, simple heuristics used as Upper Bounds on the number of time steps used, and metaheuristic algorithms as Genetic Algorithm and Simulated Annealing. Finally an air traffic scenario has been simulated using real air traffic data in order to test our algorithms. Graphic Tools have been used to represent the Milano Linate air space and its air traffic during different days. Such data have been provided by ENAV S.p.A (Italian Agency for Air Navigation Services).

Contents

Chapter 1

Introduction

As frequently happens, technologies that demonstrate their validity in military context are subsequently used in civil applications. In past years, the use of UAS (Uninhabited Aerial System) in military missions has proved their capability and versatility in different applications: the D3 (Dull, Dirty, Dangerous) paradigm emphasizes the ability of UAS to operate autonomously in complex scenarios, reducing both the necessity of human intervention and its related risks. There are many civil applications that can be performed by the UAS: for example, crop spraying or active volcano monitoring as "dirty" missions, crowd and traffic monitoring as "dull" missions and search and rescue (SAR),fire monitoring and fire fighting as "dangerous" missions. The D3 paradigm can be translated in civil contexts, achieving several advantages like reducing risk for human crew, increasing mission efficiency and, not less important, reducing operative and personnel costs. As a matter of fact the costs aspect is very important and will increase its importance in future years as air traffic and its related costs increase. The forecasts of air traffic growth are significant, and the crisis that struck the economic system in 2009 underlines how this growth cannot leave aside the aspect of cost reduction. In addition, the automation of some UAS missions in civil air space can be considered as a first step in a more extensive and integrated automation process that, in a not too distant future, will probably see the automation of all civil air transport. This process will involve different aspects, such as Air Traffic Management, Aircraft and Airport Operations and Guidance and Navigation Systems. The use of the UAS for civil missions will be one of the most important steps in this context.

UAS insertion in civil non-segregated airspace addresses quite a number of issues, such as the level of priority that unmanned air traffic should have relative to manned air traffic, the coordination between the UAS controller and air traffic control, the level of automation of the UAS, and its sensing and avoidance capacity. The management of potential conflicts between the UAS and manned air traffic is a challenging issue that is often considered as an instance of the general Conflict Detection and Resolution (CDR) problem. This problem aims to detect and solve the conflict between two vehicles and has been consistently investigated in past years.

Since the first concepts concerning Free Flight were envisaged, both the industry and the academia research communities have paid attention to the CDR problem. In order to study the feasibility of self separation, many prototype tools such as the Autonomous Operations Planner (AOP), Future ATM Concepts Evaluation Tool (FACET) [1] developed at NASA and the Airborne Separation Assurance System (ASAS) [2] developed at the National Aerospace

Laboratory (NLR) in the Netherlands have been proposed. All these tools implement CDR algorithms. The algorithms used in the Autonomous Mediterranean Free Flight (AMFF) simulations have proved to be effective with limitations for dense traffic conditions [3]. AMFF is a state based conflict detection and resolution concept with a level of automation such that the pilot follows the automatically generated conflict resolution messages by steering the aircraft. Farley et al. [4] assessed the performance of a conflict resolution algorithm developed as part of the Automated Airspace Concept [5] for conflicts detected in any phase of flight - including arrival merging operations - in a simulation environment designed to represent the complexity, variety and volume of current and future air traffic operations. The literature subsequently expanded to include approaches based on genetic algorithms [6] [7] [8] and arrival-time constraints. In this thesis we present the problem of the UAS mission management in controlled airspace. As an example of the first UAS employment, a surveillance mission is considered: it consists in overflying and photographing a set of target points at a scheduled altitude, self separating from the piloted traffic. We investigate such problem using an off line approach. We thus consider the air space lay-out and the air traffic as input data for our problem. A similar problem have been presented in [21]. The presence of piloted and non-piloted air traffic in the same airspace requires that a minimum separation between them is defined and always provided. Priority politics also have to be defined to regulate the common management of these two types of airspace system users. Priority politics in non-segregated airspace will depend on the priority and objectives of the UAS mission and on the contingency of civil aircrafts. Avoiding collisions between aircraft and expediting and maintaining an orderly flow of air traffic are very important goals for ATM. Air Traffic Controllers (ATC) use different strategies for this purpose: the most important are radar vectoring, assignment of different flight levels, rerouting and assignment of holding over a point.

The methodology used in this thesis is based on a hybrid approach that merges ATM-operative technique with the Operational Research. This approach shows how the problem of finding the best order of visiting a set of target points and the problem of maintaining a minimum separation between air vehicles can be combined and modeled using a time extension of a classic routing problem, the Travelling Salesman Problem, called Time Dependent-TSP (TDTSP). Some Integer Linear Programming formulations are proposed and several exact and heuristic algorithms have been designed to solve this problem.

The TSP is a NP-Hard combinatorial optimization problem that finds the best way, in terms of a given objective, to visit a set of clients (time minimization, cost minimization, etc). The TSP has been widely discussed in past years; many exact and heuristic algorithms have been developed and tested for several practical problems. For a complete survey on the TSP see [9]. The TDTSP arises when the cost of the paths between the clients depends on time. The first TDTSP models were proposed in order to deal with traffic problems in large urban areas. In these cases the cost of crossing a path depends on the time of the day: some routes are affected by traffic peaks at a particular time of the day and the time-to-target depends on the traffic jam. The basic idea of this thesis is that this model is also useful to interpret air traffic. In fact, the cost of a UAS route between two target points depends on the air traffic that the UAS will meet, starting that path at a specific moment. Studies on TSP with time-dependent travel speed are rare in the literature. In 1981, Beasley [10] solved this problem, considering the case where two periods of the day present different travel times, adapting the savings algorithm. In 1992 Malandraki and Daskin [11] proposed a Mixed Integer Linear Programming formulation of the TDTSP and of the related vehicle routing problem TDVRP; they handled the travel time functions as a step function and presented several heuristic algorithms to solve these problems. In the same year, Hill and Benton [12] presented a model for estimating the average travel speed for the TDVRP; they also presented a simple greedy algorithm to solve this problem. In 1993, Gouveia et al. [13] presented a classification of the TDTSP, considering old and new formulations obtained from a quadratic assignment model for the TDTSP. In 1996, Malandraki and Dial [14] used a dynamic programming heuristic algorithm to solve the TDTSP with a given starting time from the depot. In 2000, Park [16] considered the bi-criteria TDVRP with time and area dependent travel speed. The minimization of total vehicle operation time and the minimization of total weighted tardiness were the conflicting objectives studied and a mixed integer linear programming formulation for the problem was presented. This model was characterized by the waiting time at nodes that occurs when a vehicle awaits the next time interval for more rapid movement. Park also presented a heuristic algorithm to solve this problem called the bi-criteria-saving algorithm. In 2004, Fleishmann et al. [15] described the derivation of travel time data from traffic information systems. They also tried to overcome the problem of the so-called non-passing property by a smoothed travel time function used for calculating arrival time given a departure time. The non-passing property arises in the TDVRP when an earlier departure time of a vehicle must be coupled with an earlier arrival time, and vice versa. Computational results were tested on traffic information obtained from the city of Berlin. In 2005, Chen et al. [17] solved the real-time TDVRP with time windows, using a heuristic algorithm that included methods for route construction; a technique to choose the optimal departure time was also developed. In 2007, Stecco et al. [20] proposed a branch and cut algorithm designed for a TDTSP used to model a production scheduling problem. In fact, as the set-up time between two jobs is a function of the completion time of the first job, a scheduling problem can be reformulated using a TDTSP approach. Stecco also introduced some families of valid inequalities used to strengthen the Linear Programming Relaxation. In 2008, Soler et al [18] considered the TDVRP with time window; they showed how it is possible to extend this problem into an Asymmetric Capacitated Vehicle Routing Problem in order to solve it with known exact or heuristic code. In 2009, Kuo et al. [19] proposed a Tabu Search to solve the TDVRP and the related goods assignment problem: a true case of a warehousing company was used to illustrate the method studied.

In this thesis real air traffic data from Milano Linate (LIML) Terminal Maneuvering Area (TMA) are used to test our algorithms. Computer Graphic tools are also used to analyze the results and to rapidly evaluate the performances of the proposed conflict resolution models. Aircraft position, altitude and flight path are represented as over-laid onto the environment virtual reconstruction. Air space data as VOR and Fix are synoptically represented within the photorealistic representation. The methodology that we propose is innovative for two basic reasons: the conflict resolution uses an intent-based ATM approach: many algorithms in the literature use state-based techniques in order to optimize the trajectory, instead we model the avoidances through maneuvers used by Air Traffic Controllers. Moreover the mission is modeled using a variation of a classic routing problem that makes it possible to consider traffic jams in air space as in urban areas.

Chapter 2

Air space modeling

As anticipated In this thesis we investigate the problem of UAS insertion in non-segregated or controlled airspace from an off line approach. For this reason, we consider the air space lay-out and the air traffic as input data for our problem. In this Chapter we introduce the main characteristics of the airspace and its main players. We also introduce some notation to model this mission environment.

2.1 The controlled air space

Civil and military flights conducted in accordance with the rules and provisions of ICAO are stated as GAT General Air Traffic (GAT). The Italian Air Force and ENAV S.p.A. provide the Air Navigation Services (ANS) in the Italian Air Space. These services are:

- Air Traffic Control Service (ATC)
- Flight Information Service (FIS)
- Alarm Service (ALRS)

A controlled air space is a "part" of the air space where the air traffic control service is provided both to IFR and VFR flights. A flight that follows instrumental rules for navigation is called IFR (Instrumental Flight Rules), while a flight that follows the visual rules is called VFR. Basically, IFR flights use radio-assistance such as VOR (VHF Omni directional Range) or NDB (Non Directional Beacon) for navigation. VFR flights use visual reporting points. ICAO regulations identify several classes of air space defined by a letter: A, B ,C, D, E, G. In the Italian Air Space only four categories are used: A, C, D, G.

- Air space A is a controlled air space where the ATC service consists in providing a minimum separation between the IFR flights while VFR flights are not allowed.
- Air space C is a controlled air space where the ATC service consists in providing a minimum separation between the IFR flights and between IFR and VFR. Traffic information is provided between conflicting VFR flights.

- Air space D is a controlled air space where the ATC service consists in providing a minimum separation between the IFR flights, while between IFR and VFR flights only traffic information is provided.
- Air space G is a non-controlled airspace. Only traffic information is provided between conflicting aircrafts.

The Terminal Maneuvering Area (TMA) is a particular A class air space typically used to manage congested areas near large airports. In Italy, one of the most important TMAs is Milano TMA. An aircraft flying into the TMA follows the standard route or vectors provided by the Air Traffic Controllers. A departing aircraft follows a SID (Standard Instrumental Departure) route, while an arriving aircraft follows a STAR (Standard Arrival Route) route. The first one links the airport with the en-route airspace, the second the en-route with the airport. Each route is identified by a set of navigation points defined by VOR, NDB or Fix Points.

Let us introduce some notation used to model the air space environment. If we consider a defined airspace, the set M_a represents the set of the navigation points used by the aircraft. In that airspace, a set A of n aircraft fly during the period considered (see following chapters). Each aircraft $a \in A$ is characterized by a route $R_a = \{m_1 ... m_{ra}\}$ defined by a succession of $r_a \in N^+$ navigation points. l_m^a and t_m^a represent respectively the altitude and the time of the aircraft a over its routing point m. We define the set $V_a = \cup_{a \in A} R_a$ as the set of aircraft routes. In addition, we define the graph $G_a = \{M_a, V_a\}$ as the Aircraft Graph. This graph is direct and asymmetric and makes it possible to represent the mission environment. Each aircraft that flies in the modeled airspace during the mission is moving over the Aircraft Graph.

2.2 The Air Traffic Controllers

The air navigation services are provided to the pilots by the Air Traffic Controllers (ATCs). One of ATCs' most important goal is to maintain and expedite an orderly flow of the air traffic. In fact, they ensure a minimum separation between the aircraft under control. ICAO regulations define different types of separations between aircrafts:

- Geographical Separation.
- Vertical Separation.
- Horizontal Separation.
- VOR/DME separation.
- Radar Separation.

According to the air space classification, at least one of these separations must exist between each pair of aircrafts in the controlled air space. A Geographical Separation between two or more aircrafts occurs when they overfly points that are considered (by appropriate studies conducted by the aeronautical authority regulator that in Italy is ENAC: Ente Nazionale Aviazione Civile) sufficiently far away from each other. A vertical separation between two aircrafts occurs when their distance on the vertical direction is greater than or equal to a minimum vertical distance defined by the aeronautical authority. Typically this distance is 1000 ft for GAT or 2000 ft for State flights. VOR/DME (Distance Measuring Equipment) separations are applied in non radar airspace or in the event of Radar failure. They are based on the position (radial of the VOR and distance from the DME) of the aircraft related to such radio assistance. Radar separation is provided through radar vectors assigned by the ATCs. To provide the minimum separation, the ATC uses different techniques: rerouting, holding patterns, speed control and vertical or horizontal avoidance. The conflicting aircraft are instructed by the ATC to perform one of these maneuvers.

2.3 The UAS

The UAS technologies were conceived in a military context. As a reference for our thesis we use the General ATOMICS MK2 Reaper called Predator B.

Figure 2.1: General Atomics MK2 Reaper: Predator B.

Its most important performances are:

- Cruise Speed: 120-170 kts
- Max Speed: 260 kts
- Endurance: 14 hours
- Sevice Ceiling: 50000 ft

As an example of civil UAS employment, a surveillance mission is considered. The mission consists in departing from a ground base, taking pictures over a set M (Mission Way Points) of m targets at a given mission altitude and coming back to the same airfield. Between each pair of targets a direct route is available; let V be the set of all possible sub-routes between such points and thus let us indicate (i, j) as the generic sub-route. The cost of a sub-route can be approximated using the flight time once the distance between the points and the UAS mission parameters P_{uas} are known. These parameters depend on the UAS performance and consists in the speed v_{uav} (best range speed or best endurance speed, cruise speed), the rate of climb $\tilde{r_c}$ and the rate of descent $\tilde{r_d}$ that the UAS would have used in the same sub-routes in segregated air space. We introduce the UAS mission parameters P_{uav} because we want to evaluate how the mission changes in controlled air space "independently" from the objective of that mission (minimum time, minimum consumption,). Using such parameters, the flight time necessary to reach j from i is $f_{i,j}$. Due to the different performance during the climb and descent phase, we presume that the cost from target point i to target point j is different than the cost from j to i . The set of mission targets and the set sub-routes define the UAS graph $G_u = \{M_u, V\}$ that is a directed and asymmetric graph.

The problem of finding the cheapest route that visits the target points in segregated airspace (no civil air traffic) corresponds to the problem of finding the Hamiltonian circuit on the graph G_u and can be modeled through the Asymmetric Traveling Salesman Problem (ATSP). Thus, if the same mission had been performed in segregated air space, the UAS route would have consisted in the ATSP route. It is also important to note that this route represents a Lower Bound on the value of the same mission in non segregated airspace. Finally as priority politic we consider that the UAS has a low priority respect to the other airspace users. In other words it has to change its route to avoid the air traffic and not vice versa. The only exception is represented by the case in which the UAS starts its holding over a target point. In such case we suppose that the UAS can maintain its holding until necessary.

Chapter 3

Mission modeling

3.1 The Conflict Graph

During this mission in controlled air space n piloted aircrafts fly into the same environment following the authorized routes: the UAS has to self separate from the piloted air traffic, maintaining a minimum separation (5NM or 1000ft). For this purpose, some ATM operative techniques are used: rerouting or holding over a mission target at the altitude required by the mission. A holding of the UAS can cause problems to the ATC depending on its position with respect to the airport and to the arriving and departing routes (SIDs and STARs). We consider two different objectives:

- Objective 1: Planning the UAS route that minimizes the air traffic impact over the mission.
- Objective 2: Planning the UAS route that minimizes the impact of the mission over the air traffic.

The first goal consists in finding the UAS route that minimizes the mission endurance, maintaining the minimum separation from the piloted traffic and using the same parameters P_{uav} that it would have used in segregated air space. On the other hand, if we consider the impact of the mission over the ATC we have to consider that certain holding patterns can cause problems to the ATC system as anticipated. Thus, the mission endurance is weighted according to the position of the holdings or of the routes with respect to the airspace lay-out.
Independently from the goal, this problem has two decisional dimensions: the choice of the best order of visiting a set of way points and the choice of the avoidance techniques that have to be used. The first problem corresponds to the well known Travelling Salesman Problem (TSP), while the second pertains to a UAS path planning problem that has been investigated in the literature using several approaches. Our thesis supports the idea that these two problems can be combined in a TDTSP framework, considering that the avoidance of the piloted air traffic in a sub-route between two mission way points depends on the moment at which the UAS starts to fly between them. In other words, the cost of a path between two mission way points varies in time. Starting a sub-route at a specific moment could implicate a conflict and

thus require appropriate management (rerouting or holding), while starting at another moment could be conflict free and thus the UAS can proceed directly between its targets. Note that rerouting is not exactly an avoidance maneuver; it is intrinsically in the TSP formulation. The temporal dimension is introduced by dividing the temporal horizon into time steps. If t_{END} is an upper bound on the mission duration (for example 300 minutes) obtained by the UAS endurance and Δt is the duration of a time step, the set $T = \{1, 2, ..t_{max}\}$ is the set of time steps where $t_{max} = t_{END}/\Delta t$ represents the number of time steps used. To calculate the function of the time $w_{(i,j)}(t), T \rightarrow R$ that represents the weight (as an estimation of the time) of the sub route (i, j) started by the UAS at time step $t \in T$, let us introduce the "Conflict Graph".

As the TSP is associated with a graph, the UAS mission environment can be modeled using a "Conflict Graph". We define a Conflict Graph a "five-tupla" $G = \{G_u, A, G_a, S, T\}$. S is the set of possible conflicts: a conflict $s \in S$ between the UAS u sub-route $(i, j) \in V, V \in G_u$ and an aircraft a route $R_a \in G_a$ occurs if at least one point exists over (i, j) such that its distance from R_a is less than 1000 ft on the vertical plane or 5 NM on the horizontal plane. In this case, the point over (i, j) that is closest to R_a is defined as the Conflict Point p_{cnf}. The time step in which the aircraft is overflying the closest point to p_{cnf} on its route is the Conflict Time Step t_{cnf}. Note that if the UAS and the aircraft sub-route are coincident or parallel (for example the same or opposite route) p_{cnf} degenerates in a segment and t_{cnf} in a time interval (set of time steps). If no conflict occurs, the UAS flies direct from $i - th$ to $j - th$ target. As anticipated in the previous chapter, G_u is an asymmetric and direct graph and thus, to model the conflict, it is necessary to discriminate the case of the UAS moving from i to j from the case of the UAS moving from j to i .

The UAS route R_u over the Conflict Graph can be described by two vectors: the first (route vector) comprises $(m + 1)$ elements and contains the succession of targets (visiting order), the second (holding vector) comprises m elements and contains the holding in time steps over each target. The associated TSP solution of the problem (in segregated air space) can be described by the route vector that corrsponds to the TSP route and by the holding vector that comprises holdings equal to 0.

Over the involved sub-routes of the graph G_u, each conflict s defines a related Conflict Zone that is a sub-separation area defined by the UAS and the aircraft routes segments that are not separated; these segments cannot be occupied by the UAS and the aircraft together. To identify the Conflict Zone, the projection on the horizontal plane π_h of the two conflicting routes is used. Let us consider fiugre 3.1 where the segment $p_i - p_j$ represents the projection on π_h of the UAS sub-route (i, j) and the segment $p_a - p_b$ represents the projection on π_h of the aircraft conflicting route. t_a and t_b are the time steps in which the aircraft is overflying p_a and p_b respectively.

We define the point p_{1cu} as the projection on π_h of the first point over the UAS sub-route between p_i and p_{cnf} that presents a distance of 1000 ft in the vertical direction or 5NM in the horizontal direction, whichever happens first, moving from p_i to p_{cnf}. Similarly, the point p_{2cu} is the projection of the first point over the UAS path between p_j and p_{cnf} that presents a distance of 1000 ft in the vertical direction or 5NM in the horizontal direction, whichever happens first, moving from p_j to p_{cnf}. In the same way, the points p_{1ca} and p_{2ca} are defined over the aircraft a route; t_{1ca} and t_{2ca} are the time steps in which a overflies these points respectively. A "sub-separation stretch" over $p_i - p_j$ is identified between p_{1cu} and p_{2cu}, and another one over $p_a - p_b$ between p_{1ca} and p_{2ca}. Note that the length of these sub-separation

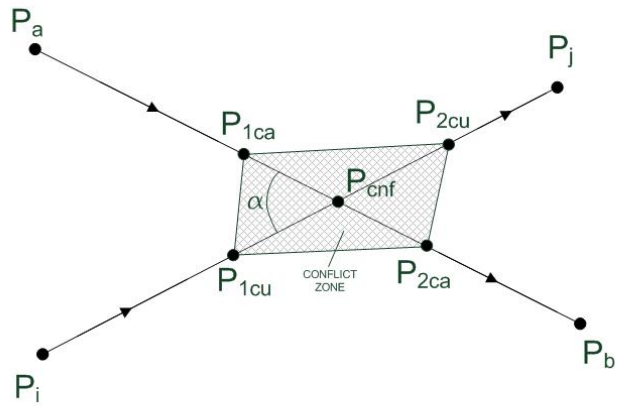


Figure 3.1: Conflict Zone

stretches depends on the angle between the routes projection on the horizontal plane and on the same angle on the vertical plane. α is the angle between the projection of the routes on the horizontal plane and describes whether the routes are converging or diverging. On the other hand, the angle on the vertical plane depends on the UAS and aircraft altitudes over the points and is related to the rate of climb/descent that the they have to set to comply with these altitudes. Obviously, if $\alpha = 90$ and if the UAS and the aircraft fly at the same altitude, the Conflict Zone is represented by a circle of radius 5NM, whereas for $\alpha = 0$ or $\alpha = 180$ the aircraft and the UAS present the same (or opposite) route. In this case the entire sub-route becomes a Conflict Zone.
Considering the UAS flight parameters P_{uas}, for each sub-route involved in a conflict, it is possible to identify the time steps that implicate a sub-separation. These time steps define the time interval for the conflict s, called Conflict Interval $X^s_{(i,j)}$ for the sub-route (i, j), in which a departure from i to j of the UAS implicates a sub-separation in the Conflict Zone and thus requires a holding of length $|X^s_{(i,j)}|$ time steps. Specifically, we indicate with $s(Ra, (i, j), Puav)$ the conflict condition and:

$$s(Ra, (i, j), Puav) = \begin{cases} |X^s_{(i,j)}| & \text{if a conflict occurs} \\ 0 & \text{otherwise} \end{cases}$$

For a given conflict, the Conflict Interval depends on the position and on the extension of the Conflict Zone related to the UAS route and on its performance. We identify three basic cases by which it is possible to model several sub-cases combining them.

3.1.1 Case 1: The Conflict Zone does not interest p_i or p_j and only one aircraft sub-route is involved.

This is the general case, and is reported in figure 3.1. First we consider the case of the UAS moving from p_i to p_j (sub-route (i, j)). The UAS last departing time step from p_i is such that it can arrive in p_{2cu} when the aircraft a is in the point p_{1ca} . If $\Delta t_{p_i - p_{1cu}}$ is the time step interval necessary to the UAS to fly from p_i to p_{1cu} maintaining P_{uas}, the Conflict Interval over p_i starts at time step:

$$t'_{p_i} = max\{0, t_a + \Delta t_{p_a - p_{1ca}} - \Delta t_{p_i - p_{2cu}}\} \tag{3.1}$$

Similarly, if the first UAS redeparting time step from the target point p_i is such that it can arrive at p_{1cu} when the aircraft a is at the point p_{2ca}. The Conflict Interval over p_i finishes at time step:

$$t''_{p_i} = max\{0, t_a + \Delta t_{p_a - p_{2ca}} - \Delta t_{p_i - p_{1cu}}\} \tag{3.2}$$

The Conflict Interval over p_j is determined in the same way considering the UAS flying from p_j to p_i (sub-route (j, i)).

$$t'_{p_j} = max\{0, t_a + \Delta t_{p_a - p_{1ca}} - \Delta t_{p_j - p_{1cu}}\} \tag{3.3}$$
$$t''_{p_j} = max\{0, t_a + \Delta t_{p_a - p_{2ca}} - \Delta t_{p_j - p_{2cu}}\} \tag{3.4}$$

3.1.2 Case 2: The Conflict Zone does not interest p_i or p_j and more than one aircraft sub-route is involved

Figure 3.2 reports an intersection between the UAS path $p_i - p_j$ and an aircraft route such that more than one aircraft sub-route is involved: the aircraft direction changes in the Conflict Zone.

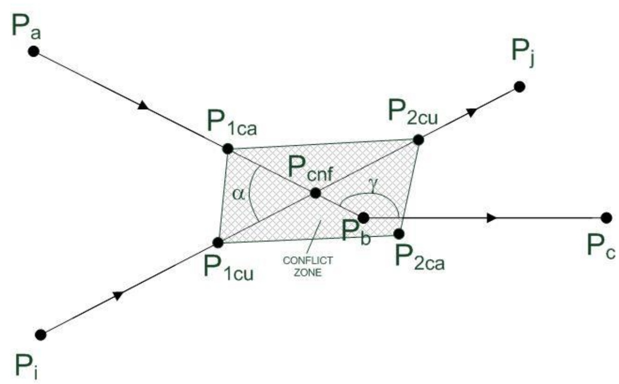


Figure 3.2: Conflict Zone Case 2

Thus the sub-separation zone on the aircraft route could comprise more than one sub-separation stretch. Consequently, the sub-separation stretch on the UAS route depends on the angle between the two aircraft sub-routes: the points p_{1ca} and p_{2ca} could be defined using the distance from different aircraft sub-routes. In Figure 3.2 γ represents the angle between $p_a - p_b$ and $p_b - p_c$. The point p_{2ca} lies on $p_b - p_c$ and the point p_{2cu} is defined through the distance from the sub-route $p_b - p_c$ and the UAV path $p_i - p_j$.

3.1.3 Case 3: The Conflict Zone involves p_i or p_j or both points

This case requires a further consideration because in this kind of conflict more than one UAS sub-route is involved. In fact, due to the combinatorial approach used to model the problem, the direction from which the UAS will arrive at p_i or at p_j is not known. In this way a departing time step from another mission target could exist such that a conflict occurs if the UAS arrives at p_i or at p_j when the aircraft is in the conflict zone. In the same way, the next sub-route of the UAS is not known, so it is not possible to define the conflict zone as previously reported. An example is provided in Figure 3.3 the Conflict Zone involves p_j . To overcome this problem, if the UAS is moving from p_i to p_j (sub-route (i, j)), the Conflict Zone is modeled using a cylinder of 2000ft height and radius of 5 NM centered at p_cnf ; the point p_{2cu} is defined as the intersection between the Conflict Zone and the direction of $p_i - p_j$.

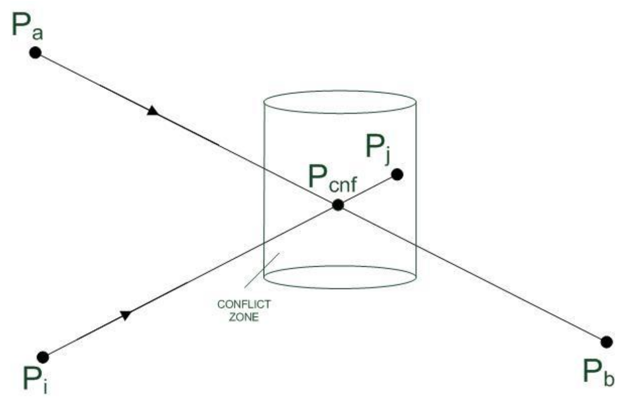


Figure 3.3: Conflict Zone Case 3a

Whatever the UAS direction in the next path, the mentioned conflict defines a Conflict Interval over p_i. This interval can be evaluated using the equations reported for case 1 considering the new definition of p_{2cu}. On the other hand, if the UAS is moving from p_j to p_i (sub-route (j, i)) as reported in Figure 3.4, it is necessary to consider that the conflict zone over p_j defines a Conflict Interval also over each point $q \in M, q \neq i, q \neq j$. In this case, in fact, it is more proper to speak about a visit of p_j that could, in a defined time step, implicate a conflict. Thus all departures from the entering sub-routes (q, j) are involved and require the definition of a Conflict Interval $X^s_{(q,j)}$ and a related holding over q.

In this way, whatever the UAS direction in the previous sub-route, the conflict is modeled. Considering the example of Figure 3.4, for a generic $q \in M, q \neq i, q \neq j$, the point p_{2cu} is the first point over the UAS sub-route between p_q and p_j that presents a distance of 1000 ft in the vertical direction or 5NM in the horizontal direction, whichever happens first, moving from p_q to p_j. A Conflict Interval is identified for each $q \in M, q \neq i, q \neq j$ using the equation described for case 1 with the new definition of p_{2cu} applied to each q. The same procedure can be applied if the Conflict Zone involves p_i using the appropriate changes. Finally, if the entire UAS sub-route (i, j) is involved in the Conflict Zone, a Conflict Interval is identified for each sub-route involved that has an extreme point in i or j, using the procedure described above. In addition, a Conflict Interval is associated with both i and j that corresponds to the time steps in which the aircraft is passing in that Conflict Zone. As an example, this conflict situation is verified when the aircraft and the UAS have the same/opposite route or when

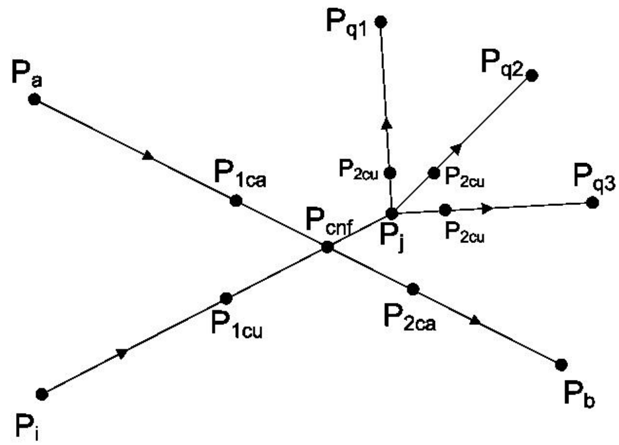


Figure 3.4: Conflict Zone Case 3b

they are parallel but not separated.

As anticipated, many conflict cases can be derived from the combination of these basic cases. For example, let us consider the case in which the conflict zone involves more than one aircraft sub-route and p_i or p_j. We must also point out that if a conflict can not be modeled using this basic cases or an appropriate combination of them, we consider the entire UAS sub-route involved in the conflict. The related Conflict Interval over i, over j and over the all q connected with them is determined by the time steps in which the aircraft is passing. Finally, it should be noted that, by using this model, more than one Conflict Interval could be associated with a UAS sub-route.

3.2 The arc cost function

The cost of a sub-route (i,j) function of the departing time step can be described considering that in each time step $t \in T$ the UAS has basically two options: if no conflict occurs ($t \notin X^s_{i,j}, \forall s \in S$) the UAS can proceed directly from i to j and thus the cost corresponds to the flight time $f_{i,j}$, while in the event of conflict ($t \in X^s_{i,j}$) the cost consists of the sum of two components: the remaining holding from t until the first no conflicting time step and the flight time $f_{i,j}$. Thus we define the set $Xt^s_{i,j}$ as the subset of time steps of $X^s_{i,j}$ from t to the last element of $X^s_{i,j}$.
In the following pseudo-code, we present a simple Algorithm that summarizes the entire procedure of conflict detection and estimation of the $w_{i,j}(t)$ for each and $(i,j) \in V$.

Finally, figure 3.5 reports an example of $w_{i,j}(t)$. In the event of arriving in a time step $t \in X^s_{i,j}$ a holding must be performed. The UAS can then continue its sub-route. Note since now that the main difference of the two objectives previously presented consists in this aspect. In fact, considering the case of all holding patterns that have the same cost (objective 1), the UAS should leave that target as soon as possible. This does not happen if the holding patterns have different costs (objective 2). In fact, a longer wait in some holding patterns can mean a shorter and a more expensive pattern.

Cost estimation Algorithm

BEGIN
$\forall a \in A$
$\forall (i,j) \in V$
if $s(R_a, (i,j), P_{uas}) > 0$ **then**
$\forall t \in T$
if $(t \in X^s_{i,j})$ **then** $w_{i,j}(t) = |Xt^s_{i,j}| + f_{i,j}$
else $w_{i,j}(t) = f_{i,j}$
else $\forall t \in T \ w_{i,j}(t) = f_{i,j}$
END

Table 3.1: Cost estimation Algorithm Pseudocode

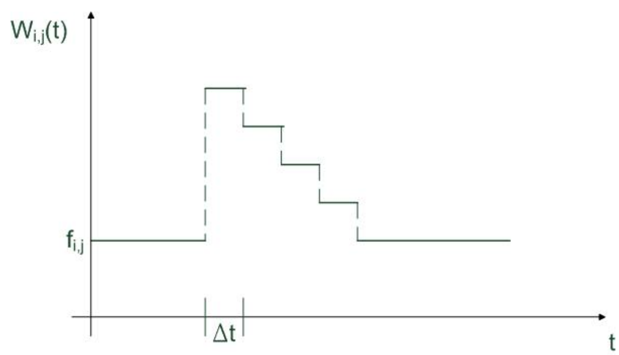


Figure 3.5: The arc cost function

For that reason we define another set of time steps: $X_{i,j} = \cup_{s \in S} X^s_{i,j}$. This set defines the forbidden departure time steps from i for the sub-route (i,j). In other words, whatever the objective of the problem, the UAS cannot leave i for j in any time step $t \in X_(i,j)$.

3.3 The UAS Routing Problem

We define the UAS Routing Problem URP as follows: Let $G = \{G_u, A, G_a, S, T\}$ be the mission conflict graph, at each vertex (target point) $i \in M, M \in G_u$, a holding pattern k_i is associated. Let β_i be the cost of a time step in the holding pattern associated with i and α the cost of a time step during the flight phase. Moreover let $f_{i,j}$ be the flight time from i to j, maintaining the flight parameters P_{uas}. For each UAS route (i,j) let $X_{i,j}$ be the set of time steps in which a departure from i to j implicates a conflict with an aircraft $a \in A$. The objective consists in finding a Hamiltonian Circuit and an associated Holding over each target whose global cost is minimum. As anticipated, note that if we want to minimize the impact of the ATC over the mission we can set $\alpha = 1$ and $\beta_i = 1, i \in M$. Otherwise the impact of the mission over the ATC is considered.

Chapter 4

TDTSP Integer Linear Programming formulations in literature

4.1 Introduction

Two types of TDTSP exist in the literature. The first one is characterized by the arc cost that depends on its position in the circuit, the second one by the arc cost that depends on the time in which is performed. Our thesis is based on this kind of TDTSP. As anticipated, its literature is quite rare. In this chapter, this literature is analyzed. The most important ILP formulations of the TDTSP and on the TDVRP are reported. Moreover, the algorithm used and the instance solved are analyzed. For each author we introduce the same notation used in his/her paper, these notations have to be considered applied only in the related paragraph.

4.2 Malandraki and Daskin

Malandraki and Daskin present the TDTSP as a special case of the TDVRP. They present an ILP formulation of such problems, then they analyze the related properties and propose some heuristic algorithms. The TDVRP is defined as follows: a vehicle fleet of fixed capacities serves customers of fixed demands from a central depot. Customers are assigned to the vehicles and the vehicles routed so that the total time of the routes is minimized. The travel time between two customers or between a customer and the depot depends on the distance between the points and time of day. The TDTSP is a special case of the TDVRP in which only one vehicle of infinite capacity is available.

n: number of nodes including the depot
M : number of time interval considered for each link
K: number of vehicles
$c_{i,j}^{m}$: travel time from node i to j if starting at i during time interval m;$c_{i,i} = \infty, \forall i, m$
c_i: service time at node i, $c_i = 0 \forall i = 1, n + 1 ..., n + K$

$T^m_{i,j}$: upper bound for time interval m for link (i, j)
t: starting time from the depot node 1
b_k:weight (or volume) capacity of vehicle k
d_i:weight (or volume) to be collected to customer i; $d_i = 0 \forall i = 1, n + 1..., n + K$
B_1: a large number
B_2: a large number
$B = max_k b_k$: capacity of largest vehicle
L_i: earliest time that the salesman can arrive at node i
U_i: latest time that the salesman can arrive at node i

Variables:

$$x^t_{(i,j)} = \begin{cases} 1 & \text{if any vehicle travels directly from node } i \text{ to node } j \text{ starting from } i \text{ during time interval } m \\ 0 & \text{otherwise} \end{cases}$$

t_j = departure time of any vehicle from node j

w_j = weight (or volume) larger than or equal to that carried by a vehicle when departing from node j

The ILP formulation for the TDVRP is the following:

$$\min \sum_{k=1}^{K} t_{n+k} \tag{4.1}$$

Subject To:

$$\sum_{i=1,i\neq j}^{n}\sum_{m=1}^{M} x_{i,j}^{m} = 1 \qquad (j = 2, ...n + K) \tag{4.2}$$

$$\sum_{j=2,j\neq i}^{n+K}\sum_{m=1}^{M} x_{i,j}^{m} = 1 \qquad (j = 2, ...n) \tag{4.3}$$

$$\sum_{j=2}^{n}\sum_{m=1}^{M} x_{i,j}^{m} = K \tag{4.4}$$

$$t_1 = t \tag{4.5}$$

$$t_j - t_i - B_1 x_{i,j}^{m} \geq c_{i,j}^{m} + c_j - B_1 \qquad (i = 1, ..., n; j = 2, ...n; i \neq j; m = 1, ...M) \tag{4.6}$$

$$t_i + B_2 x_{i,j}^{m} \leq T_{i,j}^{m} + B_2 \qquad (i = 1, ..., n; j = 2, ...n + K; i \neq j; m = 1, ...M) \tag{4.7}$$

$$t_i - T_{i,j}^{m} - 1 \geq 0 \qquad (i = 1, ..., n; j = 2, ...n + K; i \neq j; m = 1, ...M) \tag{4.8}$$

$$L_i + c_i \leq t_i \leq U_i + c_i \qquad (i = 1, ..., n + K) \tag{4.9}$$

$$w_j - w_i - B\sum_{m=1}^{M} x_{i,j}^{m} \geq d_j - B \qquad (i = 1, ..., n; j = 2, ...n + K; i \neq j;) \tag{4.10}$$

$$w_i = 0 \tag{4.11}$$

$$w_{n+k} \leq b_k \qquad (k = 1, ...K) \tag{4.12}$$

$$x_{(i,j)}^{t} \in \{0, 1\}\forall i, j, m \tag{4.13}$$

$$t_i \geq 0 \forall i \tag{4.14}$$

$$w_i \geq 0 \forall i \tag{4.15}$$

The TDTSP formulation can be obtained by this formulation as a special case in which $K = 1$ and the capacity constraints are omitted . The objective function minimizes the total route time of all vehicles (considering the travel time, the service time and the waiting time at all nodes). Constraints 4.2 and 4.3 ensure that each costumer is visited exactly once, constraints 4.4 ensure that exactly K vehicles are used. Constraint 4.5 define t as the starting time from the depot for all vehicles. Constraints 4.6 compute the departure time at node j. The objective function ensures that such constraints apply with equality when $x_{i,j}^{m} = 1$ except in the case when waiting at i decreases the objective function value. B_1 can be set equal to the total route time of a set of feasible vehicles plus the $max_{(i,j,m)} c_{i,j}^{m}$ plus $max_j c_j$. The temporal constraints 4.7 and 4.8 ensure that the proper parallel link m is chosen between nodes i and j according to the departure time from node i. B_2 can be set equal to the latest possible return time of the vehicle. Constraints 4.9 impose the time windows defined in term of arrival time. Constrains 4.10 to 4.12 impose the capacity restriction. This formulation does not require SEC because constraints 4.10 and 4.5 operate as SE constraints. To solve the TDTSP, Malandraki and Daskin propose a Nearest Neighbor Algorithm and a Cutting Plane Heuristic. This algorithm use strong SE constraints and strong temporal constraints that are iteratively added to the LP relaxation of the problem. We report the inequalities that correspond to the strong constraints: the flow chart of the algorithm is reported in the paper.

Strong SE Inequalities:

$$\sum_{i \in S} \sum_{j \in S} \sum_{m=1}^{M} \leq |S| - 1 \tag{4.16}$$

Strong Temporal Inequalities:

$$x^m_{i,j} + \sum_{k=2, k \neq j}^{n+1} \sum_{p \in A^m_{i,j}} x^p_{j,k} \leq 1 \tag{4.17}$$

for every link (i, j) in period m

$$\sum_{i=1, i \neq j}^{n} \sum_{p \in B^p_{j,k}} x^m_{i,j} + x^p_{j,k} \leq 1 \tag{4.18}$$

for every link (j, k) in period p
where:

$A^m_{i,j}$: period p for every link (j, k) such that $\{T^p_{j,k} < T^{m-1}_{i,j} + c^m_{i,j} + c_j\}$ or $\{T^{p-1}_{j,k} > T^m_{i,j} + c^m_{i,j} + c_j + DIFF\}$

$B^p_{j,k}$: period m for every link (i, j) such that $\{T^p_{j,k} < T^{m-1}_{i,j} + c^m_{i,j} + c_j\}$ or $\{T^{p-1}_{j,k} > T^m_{i,j} + c^m_{i,j} + c_j + DIFF\}$
and

$DIFF = max\{0, c^{p-1}_{j,k} - c^p_{j,k}\}$

The algorithms were tested over instances whose size is up to 2 or 3 periods and 25 nodes. The percentage gap of the Cutting Plane Algorithm with respect to the best known solution varies from 15 to 35 percent.

4.3 Stecco, Cordeau and Moretti

The authors consider a production scheduling problem with sequence-dependent and time-dependent setup times on a single machine. The setup time between two jobs is a function of the completion time of the first job, thus the problem can be formulated as a TDTSP where the travel time between two nodes is a function of the departure time from the first node. Three different ILP formulation are proposed, we report only the strongest one. Moreover, some families of valid inequalities are introduced in order to solve the problem through a Branch-and-Cut algorithm.

Let us introduce some notation:
n:number of nodes. Two dummies node are introduced and denoted by 0 and $n+1$.
A:set of arcs.
p_i: processing time at node i
$C(t) = [c_{i,j}(t_i)]$: time dpendent matrix representing the travel time on arc $(i,j) \in A$.
K: set of relevant time interval.
$c^k_{i,j}$: represents the travel time from node i to j if starting at i during time interval k.
$I^k_{i,j}$: as the upper bound of the time interval k for arc (i, j).
d: denote the length of a planning period.

Variables:

$$x^k_{(i,j)} = \begin{cases} 1 & \text{if node } j \text{ is visited immediately after node } i \text{ and the vehicle leaves node } i \text{ during interval } k \\ 0 & \text{otherwise} \end{cases}$$

$t_{i,j}$ = departure time from node j if it is visited immediately after node i

$h_{i,j}$ = number of the period in which node j is visited if it is visited immediately after node i.

The ILP formulation for the TDTSP is the following:

$$\min \sum_{i=1}^{n} t_{i,n+1} \tag{4.19}$$

Subject To:

$$\sum_{i \in P} \sum_{k \in K} x_{i,j}^k = 1 \qquad \forall j \in S \tag{4.20}$$

$$\sum_{j \in S} \sum_{k \in K} x_{i,j}^k = 1 \qquad \forall i \in P \tag{4.21}$$

$$\sum_{j \in S} t_{i,j} \geq \sum_{l \in P} t_{l,i} + \sum_{j \in S} \sum_{k \in K} (c_{i,j}^k + p_j) x_{i,j}^k \qquad (i = 1, 2, ...n) \tag{4.22}$$

$$\sum_{j=1}^{n} t_{0,j} \geq t_0 + \sum_{j=1}^{n} \sum_{k \in K} (c_{0,j}^k + p_j) x_{0,j}^k \tag{4.23}$$

$$\sum_{l \in P} t_{l,i} \geq \sum_{j \in S} \sum_{k \in K} I_{i,j}^{k-1} x_{i,j}^k + d \sum_{m \in P} h_{m,i} \qquad (i = 1, 2, ..., n) \tag{4.24}$$

$$t_0 \geq \sum_{j=1} \sum_{k \in K} I_{0,j}^{k-1} x_{0,j}^k \tag{4.25}$$

$$\sum_{l \in S} t_{l,i} \leq \sum_{j \in S} \sum_{k \in K} I_{i,j}^k x_{i,j}^k + d \sum_{m \in P} h_{m,i} \qquad (i = 1, 2, ..., n) \tag{4.26}$$

$$t_0 \leq \sum_{j=1} \sum_{k \in K} I_{0,j}^k x_{0,j}^k \tag{4.27}$$

$$t_{i,j} \leq M_1 \sum_{k \in K} x_{i,j}^k \qquad \forall i \in P, j \in S \tag{4.28}$$

$$h_{i,j} \leq M_2 \sum_{k \in K} x_{i,j}^k \qquad \forall i \in P, j \in S \tag{4.29}$$

$$t_0 \geq t \tag{4.30}$$

$$h_{i,j} \in N \qquad \forall i \in P, j \in S \tag{4.31}$$

$$x_{i,j}^k \in \{0, 1\} \qquad \forall i \in P, j \in S, k \in K \tag{4.32}$$

The objective function minimizes the departure time from node $n+1$ as the sum of the transition times between i and $n+1$. There is only one variable $x_{i,n+1}^k$ equal to one and only one i such that $t_{i,n+1} > 0$ (constraints 4.28), thus only one $t_{i,n+1}$ variable can be positive. Constraints 4.20 and 4.21 ensure that each node is visited exactly once. Constraints 4.22 compute the departure time from the node j visited after i. It is necessary to add the transition time between i and j to such departure time. Constraint 4.23 computes the departure time from a node j that is visited immediately after node 0. Constraints 4.26 and 4.29 ensure that the appropriate interval k is chosen between i and j according to the departure time from node i. Constraints 4.28 and 4.29 ensure that $t_{i,j} >= 0$ and $h_{i,j} >= 0$ if and only if one of the $x_{i,j}^k = 1$ and that otherwise $t_{i,j} = 0$ and $h_{i,j} = 0$ otherwise. M_1 and M_2 can be set as any upper bound on the value of $\sum_{i=1}^{n} t_{i,n+1}$ and $\sum_{i=1}^{n} t_{i,n+1}/d$ respectively.

The authors consider a reference period of length d (d corresponds to one day): the travel time function on arc (i, j) has the same structure in every period. Moreover, in this period there are three intervals: $|K| = 3$. The authors present some families of valid inequalities that can be added to the previous formulation in order to strengthen its linear relaxation. The first group of inequalities are bounds on the interval and are defined for nodes whose travel time function is made up of two or three intervals. The inequalities of this family are defined by introducing particular sets of nodes. Such sets are defined using specific function defined ad hoc. The second group of inequalities are bounds on the starting time and period. As the starting time t is an input of the problem, it is possible to set a bound on $t_{i,j}$ and $h_{i,j}$ variables knowing the interval in which the starting time t takes place for each node.

A Brunch-and-Cut algorithm was implemented to solve instances of different sizes: from 5 to 50 nodes considering a time horizon of one period and 3 intervals. Computational results show that this algorithm is able to solve instances up to 50 nodes and one period of 3 intervals in an acceptable computing time.

Chapter 5

ILP Formulations for URP

5.1 Notation

We resume the notation used:

M: set of Mission Targets
$m = |M|$: number of Mission Targets
A: set of piloted Aircrafts
$a \in A$: generic Aircraft
$n = |A|$: number of Aircrafts
v_{uav} : UAS speed
t_{UB}: Upper Bound on the mission duration
Δt: duration of the time step
$t_{max} = t_{UB}/\Delta t$: number of time steps
$T = \{0...t_{max}\}$: Set of Time Steps
R_u : UAS Route
R_a : Aircraft a Route
V : set of UAS possible routes
$f_{i,j}$: flight time from i to j at speed v_{UAS}
S: set of all conflicts (sub separation)
$s \in S$: generic conflict
p_{cnf}: point over the UAV route closest to the conflicting aircraft route
t_s: conflict time step (time step in which the aircraft overfly p_{cnf})
$X^s_{i,j}$: subset of time step involved in the conflict s for subroute i, j
$Xt^s_{i,j}$: holding at time step t, cardinality of $X^s_{i,j}$ from t
$w_{i,j}(t)$: time to reach j starting from i at time step t ($w_{i,j}(t) = f_{i,j} + Xt^s_{i,j}$)
$G_m = \{V, M\}$: Mission Graph
$G = \{G_m, S, A, W\}$: Conflict Graph
H: set of Hamiltonian circuit: hamiltonian route
$H_s \subseteq H$: set of Hamiltonian circuit that have at least a conflict
h_s:generic element of H_s
$X_{i,j}$: set of time steps of forbidden departure
k_{min} min holding duration
k_{max} max holding duration

5.2 Mathematical Formulation

Two decision variables are used to render the time dimension of the problem and to model the holding over the target points:

Variables:

$$x^t_{(i,j)} = \begin{cases} 1 & \text{if the UAS starts to fly from node i to node j at time step t} \\ 0 & \text{otherwise} \end{cases}$$

$$y^t_i = \begin{cases} 1 \text{ the visit of node i starts at time step t} \\ 0 \text{ otherwise} \end{cases}$$

The objective function is written as follow :

$$\min \quad \alpha(\sum_{t=0}^{t_{max}} \sum_{i \in M} \sum_{j \in M} f_{(i,j)} x^t_{(i,j)}) + \sum_{i \in M} \beta_i (\sum_{t=0}^{t_{max}} (\sum_{j \in M} t x^t_{(i,j)} - t y^t_i)) \tag{5.1}$$

The first term represents the total flight time between the target points; the second describes the holding duration over each target point as a difference between the arrival and the departure time step. The coefficients α and β_i make it possible to weigh the different phases of flight and holding: if rerouting is the preferred option with respect to holding, β_i can be increased appropriately. If $\alpha = \beta = 1$ the objective function minimizes the total mission duration in terms of time step. To obtain the mission duration in time units it is necessary to consider the duration of a generic time step Δt. The objective function is subject to the following constraints:

$$\sum_{t=0}^{t_{max}} \sum_{(i,j)\in\delta^-(i)} x_{(i,j)}^t = 1 \qquad \forall i \in M \tag{5.2}$$

$$\sum_{t=0}^{t_{max}} \sum_{(i,j)\in\delta^+(i)} x_{(i,j)}^t = 1 \qquad \forall i \in M \tag{5.3}$$

$$\sum_{t=0}^{t_{max}} y_i^t = 1 \qquad \forall i \in M \tag{5.4}$$

$$\sum_{(j,i)\in\delta^-(i), t-f_{i,j}\geq 0} x_{(j,i)}^{t-f_{(j,i)}} \leq y_i^t \qquad \forall i \in M, i \neq 0, t = 0, ..., t_{max} \tag{5.5}$$

$$\sum_{(i,j)\in\delta^+(i)} x_{(i,j)}^t \leq \sum_{t^*=0}^{t} y_i^{t^*} \qquad i \in M, t = 0, ..., t_{max} \tag{5.6}$$

$$y_0^0 = 1 \tag{5.7}$$

$$x_{(i,j)}^t = 0 \qquad \forall (i,j) \in V, \forall t : t + f_{(i,j)} \geq t_{max} \tag{5.8}$$

$$\sum_{t=0}^{t_{max}} (\sum_{j\in M} t x_{(i,j)}^t - t y_i^t) \geq k_{min} \qquad \forall i \in M \tag{5.9}$$

$$\sum_{t=0}^{t_{max}} (\sum_{j\in M} t x_{(i,j)}^t - t y_i^t) \leq k_{max} \qquad \forall i \in M \tag{5.10}$$

$$x_{(i,j)}^t = 0 \qquad \forall t \in X_{i,j}, \forall (i,j) \in V \tag{5.11}$$

$$x_{(i,j)}^t \in \{0,1\} \qquad (i,j) \in V, t = 0, ..., t_{max} \tag{5.12}$$

$$y_i^t \in \{0,1\} \qquad i \in M, t = 0, ..., t_{max} \tag{5.13}$$

$$k_{min} \in N^+ \tag{5.14}$$

$$k_{max} \in N^+, k_{max} \geq k_{min} \tag{5.15}$$

Constraints 5.2 and 5.3 are the assignment constraints; constraints 5.4 ensure that only one visit beginning exists for each target. Then the so called "Temporal Constraints" follow: 5.5 and 5.6. These Constraints joined with the Initial Condition 5.7 act as Subtour Elimination Constraints. The first one ensures that if the UAS starts to visit the target point i at time step t, it had departed from the previous target at an appropriate time step compatible with the flight time. The second one imposes that if the UAS leaves the target point i at time step t, the beginning of the visit of i must occur in a previous or in the same time step. These groups of constraints make it possible to build the Hamiltonian circuit over the mission target points as in the ATSP problem. Constraints 5.8 forbids departures after the last time step.

The Temporal Constraints can also be written to render the time coherence on the arcs:

$$x^t_{(i,j)} \leq y_j^{t+f_{(i,j)}} \qquad \forall (i,j) \in V, t = 0, ..., t_{max} \tag{5.16}$$

$$x^t_{(i,j)} \leq \sum_{t^*=0}^{t} y_i^{t^*} \qquad \forall (i,j) \in V, t = 0, ..., t_{max} \tag{5.17}$$

The minimum separation constraint is modeled using constraints 5.11. Each conflict $s \in S$ defines a set of involved time steps $X^s_{i,j}$ over the subroute (i,j). $X_{i,j}$ is set a of time steps in which the UAS will be involved in the conflict if it starts to fly from i to j. Thus they are the time steps of forbidden departure. Constraints 5.9 and 5.10 define a lower and an upper bound on the duration of the holding. Moreover the constraints 5.9 ensure that the holding over a target is consistent with the UAS performances: once the decision of a holding is taken, at least κ_{min} time steps will be necessary to complete it. κ_{min} can vary from 0 to a maximum value according to the UAS performances. As an alternative it is possible to set κ_{min} equal to 0, solve the problem and replace the holding using different avoidance maneuvers such as control speed. For example, a holding smaller than Δt can be avoided by reducing the UAS speed in the related subpath; instead a holding larger than Δt can be reassigned, if possible, to points that are in a less sensitive zone. This holding replanning and assignment require a further processing of the results obtained and will be the subject of future developments.

5.3 Problem Characteristics

The value of the optimal solution of the TSP associated with the graph G_u represents a lower bound on the value of the optimal solution of the TDTSP defined by the formulation 5.1-5.15. In fact, the inequalities 5.2 to 5.7 represent a formulation of the TSP based on time extended variables. This formulation is weaker than the Dantzing-Fulkerson-Johnson formulation of the TSP. This is due to the fact that the value of a variable in the LP-relaxation of the problem is scattered among the time dimensions. The value of the solution defined by following the TSP route and performing the appropriate holding represents an upper bound on the value of the optimal solution of the TDTSP. We call such solution HTSP (Heuristic based on TSP route), beacuse it represents a heuristic solution of the problem.

The TDTSP is an asymmetric problem. In fact, if the flight time of sub-route (i,j) is the same as that of subroute (j,i), this would not implicate that two tours that traverse the same sub-route but in opposite directions have the same total route time. Moreover, the $k-opt$ exchange or insertion heuristics cannot be easily extended to the TDTSP. The $k-opt$ exchange heuristic tries to improve a solution by exchanging k arcs while retaining feasibility. When this method is applied to the symmetric TSP, only the costs of exchanging the arcs is taken into account. Instead, considering the TDTSP, the flight time of several arcs (sub-routes) may change because the transversal direction changes and because the starting time (and consequently the air traffic) changes. For that reason this approach becomes expensive computationally. In the same way, the application of an insertion heuristic is expensive because the flight times of the sub-route subsequent to the sub-route inserted may change.

Moreover, some properties of the Euclidean TSP cannot be extended to the TDTSP. First the

optimum tour can intersect itself as does not occur in the TSP. Moreover, the convex hull is not well defined for the Euclidean TDTSP. Since the positions of the node change, the convex hull may differ from period to period.
Another characteristic of the problem consists in the properties of the objective considered. As anticipated, if we consider the minimization of the mission impact over the ATC, the holdings weight is the same because for the UAS it is not important where it has to wait but only how long. This implies that in this case, once a sequence of targets is fixed, the best way to visit them consists in waiting as little as possible over each target of the sequence. This means that if a departure time step is available when the UAS overflies a target, it continues along its route. Otherwise the UAS waits until the first available departure time step. In other words, the route cost depends only on the sequence of the targets. As a direct consequence, the algorithms studied for this objective work on the route vector. This does not happen when the holdings are weighted as we will see further on in the thesis. In fact, for a given route, to find the best way of visiting that ordered sequence represents another optimization problem. The main difference with the previous case consists in the fact that in this case an available departure time step may not correspond to a convenient departure. In fact, a waiting over a cheaper holding could be performed as an alternative to an immediate departure and a consequently more expensive wait in another holding. Based on this assumption, the problem was reformulated using a different approach such as the TSP with penalty. In fact, this approach makes it possible to model the idea that the cost of the route depends on the sequence of the targets.

5.4 Valid Inequalities

As anticipated the SEC inequalities are not necessary in the time dependant formulation reported before because the constraints necessary to render the time evolution act as SEC. Nevertheless the following inequalities can be added to the formulation 5.1-5.15 in order to strengthen its linear relaxation:

$$\sum_{t=0}^{t_{max}} \sum_{(i,j)\in\delta^+(Sub)} x^t_{(i,j)} \geq 1 \qquad 2 \leq |Sub| \leq |V| - 2, \forall Sub \subseteq V \tag{5.18}$$

These inequalities are a time extension of the well known SEC of the TSP Dantzing-Fulkerson-Johnson formulation. The Patberg-Rinaldi procedure can be conveniently used to separate such inequalities. .

We also report other tested inequalities that, according to our computational experience, do not strengthen the linear relaxation of the problem. All these inequalities can be separated by enumeration.

$$x^{t}_{(i,j)} + \sum_{k \in M} \sum_{t=0}^{t+f_{(i,j)}-1} x^{t}_{(j,k)} \leq 1 \qquad \forall i \in M, \forall j \in M/0, \forall t \in T : t + f_{i,j} \leq t_{max} \quad (5.19)$$

$$x^{t}_{(i,j)} + \sum_{t=0}^{t+f_{(i,j)}-1} y^{t}_{j} \leq 1 \qquad \forall i \in M, \forall j \in M/0, \forall t \in T \quad (5.20)$$

$$x^{t^*}_{(i,j)} + \sum_{k \in M} \sum_{t=t^*}^{t_{max}} x^{t}_{(k,i)} \leq 1 \qquad \forall i \in M, \forall j \in M/0, \forall t^* \in T \quad (5.21)$$

$$x^{t*}_{(i,j)} + \sum_{t=t^*+1}^{t_{max}} y^{t}_{i} \leq 1 \qquad \forall i \in M, \forall j \in M/0, \forall t^* \in T \quad (5.22)$$

The first inequalities state that a departure from i to j in time step t is not compatible with any departure from j before $t + f_{i,j}$. In the same way, the second state that a departure from i to j in time step t is not compatible with any beginning of the visit of j before $t + f_{i,j}$. Moreover, the third inequalities state that a departure from i to j in time step t is not compatible with an arrival in i after t. Finally the last inequalities state that a departure from i to j in time step t is not compatible with any beginning of the visit of i after t.

5.5 Branch and Cut Algorithm

A branch-and-cut algorithm was developed based on the formulation 5.1-5.15. Before starting the algorithm, a preprocessing phase is performed. This phase consists of two steps: computation of an Upper Bound and elimination of the "infeasible departure" time steps. The Upper bound t_{ub} is found using one of the heuristic algorithms presented in the following chapter. Since such algorithms require a very short computing time, the Upper Bound may also be found as the smallest between their solutions. The Upper Bound makes it possible to consistently reduce the time dimension. In fact, t_{max} is an Upper Bound defined using the UAS endurance. This endurance may be quite long with respect to the mission duration.

An "Infeasible Departure " time step is a time step in which a departure cannot take place. For example, considering the generic target i , if t_i^{sp} is the first time step in which it is possible to arrive at i, no departure from i to any other target can take place before t_i^{sp}. The shortest path time from the airport (target 0) to a target i makes it possible to define its related t_i^{sp}. The Dijkstra algorithm is an effective algorithm that finds the shortest path between two targets. Using this algorithm, the shortest path from the airport to each target is found. For each target the variables related to a departure before the shortest path arrival are removed from the problem. Due to the complexity $O(mlog(m))$of the Dijkstra algorithm this step of the preprocessing phase also requires a very short computing time.

Once the preprocessing phase has been completed, the algorithm starts solving the LP relaxation of the problem and generating valid inequalities 5.18. If its solution is integer, an optimal solution has been found. Otherwise an enumeration tree is constructed and an attempt to generate violated valid inequalities at each node is performed. These inequalities are in exponential number, at each node of the tree we use the Padberg-Rinaldi Separation

Procedure to find violated inequalities. Once found, the inequalities are added to the linear program. The Padberg-Rinaldi separation procedure is able to find many time-extended constraints such as our inequalities. Because all the inequalities described before are valid for the original formulation, the inequalities added at any node of the tree are valid for all other nodes. Thus, whenever the bound of the LP is computed at a given node of the tree, the linear program incorporates all cuts generated so far. The branch and cut-algorithm was implemented in C++ by using ILOG CPLEX 12.1.

5.6 Problem Reformulation: TSP with Penality

To avoid the weakness of the time dependent formulation, the problem is reformulated as a TSP with Penalties (called TSPP) associated with the routes. These penalties represent the holding necessary to avoid the conflict. Let us introduce some notation. Let H be the set of all the Hamiltonian circuits that represent an UAS route. Let H_s be the subset of H defined by the routes that have at least one conflict. h is the generic element of H_s and i_c^h is the conflict target of h. The conflict target is the last target before the first conflict, that is the target over which a holding is performed to avoid the conflict. $h' \subseteq h$ is the sub-route until the conflict that is comprised of all the targets until the successor of i_c^h. It is also necessary to include also such point because a conflict is related to an arc and not only to a target. $K_{i_c^h}$ is the holding duration in time steps. Note that h' is comprised by a set of non conflicting targets where $w_{i,j}(t + P_i) = f_{i,j}$ and the arc of the conflict where $w_{i_c^h,j}(t_c) > f_{i,j}$. Moreover the duration of the holding over i_c^h is $K_{i_c^h} = w_{i_c^h,j}(t_c) - f_{i,j}$

Two decision variables are used:

Variables:

$$x_{(i,j)} = \begin{cases} 1 & \text{if the UAS uses the subroute (i,j)} \\ 0 & \text{otherwise} \end{cases}$$

p_i=Holding over i:penality

The objective function is written as follow :

$$\min \quad \alpha\left(\sum_{i \in M} \sum_{j \in M} f_{(i,j)} x_{(i,j)}\right) + \sum_{i \in M} \beta_i p_i \tag{5.23}$$

Subject to:

$$\sum_{(i,j)\in\delta^-(i)} x_{(i,j)} = 1 \qquad \forall i \in M \tag{5.24}$$

$$\sum_{(i,j)\in\delta^+(i)} x_{(i,j)} = 1 \qquad \forall i \in M \tag{5.25}$$

$$\sum_{(i,j)\in\delta^+(Sub)} x_{(i,j)} \geq 1 \qquad 2 \leq |Sub| \leq |V| - 2, \forall Sub \subseteq V \tag{5.26}$$

$$\sum_{(i,j)\in h'} t_{max}(1 - x_{i,j}) + p_{i_c^h} \geq K_{i_c^h} \qquad \forall h' \in H_s \tag{5.27}$$

$$\sum_{(i,j)\in h'} -t_{max}(1 - x_{i,j}) + p_{i_c^h} \leq K_{i_c^h} \qquad \forall h' \in H_s \tag{5.28}$$

$$x_{(i,j)} \in \{0, 1\} \tag{5.29}$$

$$p_i \in N_k \tag{5.30}$$

$$N_k = \{0...k_{max}\}, N_k \subseteq N^+ \tag{5.31}$$

The first three constraints are the TSP constraints of the D-F-J formulation. We define TSP-relaxation of the formulation 5.23-5.31 the problem defined by 5.24-5.26 and by 5.29-5.31. The last two constraints relate to the route and the penalties using the big M method. Unfortunately such constraints are in exponential number. To separate them it is necessary to distinguish the case of the minimization of the mission impact over ATC and the case of the minimization of the ATC impact over the mission.

5.7 Min ATC impact over the mission Algorithm

In this case the cost of the holdings is the same and thus the target over which a holding is performed is not important in terms of objective function. The algorithm, called MIM algorithm proposed starts from the TSP route, finds its first conflict (if exists) and the related holding. If no conflict exists the TSP route represents the best solution. Otherwise the appropriate penalty constraint is added and the problem is re-solved. This procedure is iteratively repeated until no conflict exists. The pseudocode of the MIM algorithm is reported in the table 5.1.

5.8 Holding Assignment Problem

In this case the weight of a holding can be different with respect to the others according to the position of this holding in the airspace. For that reason a further consideration is necessary relating to the holding $k_{i_c^h}$ found in the previous approach. This holding represents the wait over the last target point before the conflict. If the holdings are not weighted it is not important where this wait is performed. Instead, in the case of weighted holdings it is necessary to

Min ATC impact over Mission Algorithm

t: time
$h =$: current circuit
$p =$: current holding(penalty)
$NOCNF$: conflict flag
$\overline{P}$: current Problem
i_c: conflict node
k_{hold}: holding necessary to avoid the conflict

Output:
h^*: route vector
p^*: holding vector

BEGIN
Initalize: $\overline{P}$ =**TSP-relaxation of TSPP,** $t = 0$, $NOCNF = 1$, $k_{hold} = 0$ **;**
do:
$NOCNF = 0$, $h^* = \{\}$, $p^* = \{\}$;
Solve $\overline{P}$ **and get** h **and** p ;
for $(i = 0; i \leq m; i++)$

if $(w_{h(i),h(i+1)}(t) = f_{h(i),h(i+1)})$ **then** $t = t + f_{h(i),h(i+1)}$, $h^* = h^* \cup \{h(i), h(i+1)\}$, $p^*(i) = 0$

else $NOCNF = 1$, $i_c = h(i)$, $k_{hold} = w_{h(i),h(i+1)}(t) - f_{h(i),h(i+1)}$, $h^* = h^* \cup \{h(i), h(i+1)\}$, **break for;**

if $(NOCNF = 1)$ **then add constraints to** $\overline{P}$ **such that** $h' = h^*$,$i_c^h = i_c$ and $k_{i_c^h} = k_{hold}$;

while $(NOCNF = 1)$

END

Table 5.1: MIM Algorithm Pseudocode

find the best way to carry out this wait over the targets that precede the conflict. For that reason it is necessary to solve another problem: the Holding Assignment Problem.

Let us introduce the problem and some notation. Let m_h the number of targets in h', i is the generic target point of h'. β_i is the cost of a holding time step over the target point i and $k_{i_c^h}$ is the holding duration in time steps. The goal of the HAP is to find the best way to assign $k_{i_c^h}$ holding time steps over the different target points of h'(except over the last one). We define K as a set of $k_{i_c^h}$ holding time steps. Let be $k \in K$ the generic holding time step. Unfortunately it is necessary to consider the presence of further conflicts that can occur by changing the holdings in the previous parts of the route. In fact, by reassigning the holdings, the departure and the arrival time steps of each target point that precedes the conflict point, ic changes. Thus, these time steps could correspond to conflicting time steps. In this case the UAS has to wait at least until a new departure time step that does not involve a conflict is available. For this purpose a penalty is added in a manner similar to the TSPP. We define Q' as the set of all possible assignment of $k_{i_c^h}$ holding time steps over m_h points such that at least a further conflict occurs. A generic assignment $q' \in Q'$ is represented by the appropriate pairs (k, i) that correspond to the decision variables. If $s \in S$ is the further conflict in which the UAS occurs reassigning the holding and $X^s_{i,i+1}$ is the set of time step involved in that conflict, we indicate with $Xt^s_{i,i+1}$ the subset of $X^s_{i,i+1}$ that is made up of the time steps comprised from t to the end of set $X^s_{i,i+1}$.

We introduce two decision variables:

Variables:

$$z_{(k,i)} = \begin{cases} 1 & \text{if the } k-th \text{ holding time step of } K \text{ is assigned to target point } i \\ 0 & \text{otherwise} \end{cases}$$

p'_i=Further holding over i: penality

$$\min \quad \sum_{k\in K}\sum_{i\in h'} \beta_i z_{(k,i)} + \sum_{i\in h'} p'_i \tag{5.32}$$

Subject to:

$$\sum_{i\in h'} z_{(k,i)} = 1 \qquad \forall k \in K \tag{5.33}$$

$$\sum_{k\in K} z_{(k,i)} \leq k_{max} \qquad \forall i \in h' \tag{5.34}$$

$$\sum_{(k,i)\in q'} B(1 - z_{(k,i)}) + p'_i \geq Xt^s_{i,i+1} \qquad \forall q' \in Q', \forall i \in h' \tag{5.35}$$

$$\sum_{(k,i)\in q'} -B(1 - z_{(k,i)}) + p'_i \leq Xt^s_{i,i+1} \qquad \forall q' \in Q', \forall i \in h' \tag{5.36}$$

$$z_{(k,i)} \in \{0, 1\} \tag{5.37}$$

$$p'_i \in N^+ \tag{5.38}$$

The first constraints ensure that each holding time step is assigned; constraints 5.34 represent the capacity constraint related to the upper bound on the holding duration. Constraints 5.35 5.36 define the penalties by relating them with the reassignment. B is a large number that can be set to max T.

5.9 Solving HAP

Constraints 5.35, 5.36 are in exponential number and thus the following algorithm makes it possible to separate them using the same approach presented for the Min ATC impact over the mission Algorithm. We define the AP-relaxation of the HAP the problem defined by 5.32,5.38 and by 5.37-5.38

HAP Algorithm

h':part of circuit
$i_c \in h'$:last target before the conflict
k'_h: holding over i_c
$m_{h'}$: number of targets in h'
t: time
z^*: holding vector such that $z^*(i) = \sum_{k \in K} z_{k,i}$
$NOCNF$:conflict flag
$\overline{P}$: current Problem
q^*:actual assignment(set of (k,i) such that $z_{k,i} = 1$ in the related solution)
k_f:further holding

Output:
p^*: holding vector

BEGIN
  Initalize: $\overline{P}$ =**AP-relaxation of HAP,** $t = 0$, $NOCNF = 1$, $k_f = 0$ **;**
  do:
    $NOCNF = 0$, $z^*(i) = \{0...0\}$,$p^* = \{0...0\}$;
    Solve $\overline{P}$ **and get** z^*,q^* **and** $p'_i \forall i$;
    for $(i = 0; i < m_{h'}; i++)$
      if $(w_{h'(i),h'(i+1)}(t + z^*(i) + p'_i) = f_{h'(i),h'(i+1)})$ **then** $t = t + f_{h'(i),h'(i+1)}$, $p^*(i) = z^*(i)$
      else $NOCNF = 1$, $k_f = w_{h(i),h(i+1)}(t) - f_{h(i),h(i+1)}$, **break for;**
    if $NOCNF = 1$ **then add constraints to** $\overline{P}$ **such that** $q' = q^*$, $Xt^s_{i,i+1} = k_f$, $p'_i = p_i$;
  while $(NOCNF = 1)$
END

Table 5.2: HAP Algorithm Pseudocode

5.10 Min mission impact ove ATC Algorithm

The mission impact over the ATC Algorithm, called MAM algorithm is obtained by the MIM algorithm by solving at each iteration the HAP. In table 5.3 is reported its pseudocode.

Min Mission impact over ATC

Variables:
t: time
$h = \{\}$: current circuit
$p = \{\}$: current penalty
$NOCNF$:conflict flag
$\overline{P}$: current Problem
k_{hold}: holding necessary to avoid the conflict

Output:
h^*: route vector
p^*: holding vector

Initalize: $\overline{P}$ =**TSP-relaxation of TSPP,** $t = 0$**,** $h = \{\}$,$p = \{0...0\}$**,** $NOCNF = 1$**,** $h^* = \{\}$**,** $p^* = \{\}$**,** $k_{hold} = 0$ **;**

do:

$NOCNF = 0$, $h^* = \{\}$, $p^* = \{\}$;

Solve $\overline{P}$ **and get** h **and** p ;

for $(i = 0; i \leq m; i + +)$

if $(w_{h(i),h(i+1)}(t) = f_{h(i),h(i+1)})$ **then** $t = t + f_{h(i),h(i+1)}$, $h^* = h^* \cup \{h(i), h(i+1)\}$, $p^*(i) = 0$

else $NOCNF = 1$, $i_c = h(i)$, $k_{hold} = w_{h(i),h(i+1)}(t) - f_{h(i),h(i+1)}$, $h^* = h^* \cup \{h(i), h(i+1)\}$, **break for;**

if $NOCNF = 1$ **then:**
Solve HAP with $h' = h^*$ **and** $k_{h'} = k_{hold}$ and get p^*
for each $p^*(i) > 0$ **add constraints to** $\overline{P}$ **such that** $h' = h^*$, $i_c^h = i$ and $k_{i_c^h} = p^*(i)$;

while $(NOCNF = 1)$

Table 5.3: MAM Algorithm Pseudocode

Chapter 6

Heuristic Approach for URP

6.1 Local Search

First we introduce some notation and we define the neighborhood of a solution. Let R_u be the set of possible solution of the URP. As anticipated, each solution $r_u \in R_u$ is described by two vectors, the route vector h and the holding vector p. Moreover let $\varphi(R_u - > R)$ be the objective function. We define two neighborhoods of r_u according to the different objectives considered: minimization of the ATC impact over the mission and minimization of the impact of the mission over the ATC.
In the first case the Neighborhood of r_u, $N_1(r_u)$, is the set of solutions $\hat{r_u}$ such that $\hat{h}$ and $\hat{p}$ are obtained as follow. The route vector $\hat{h}$ is obtained from h by performing an exchange $1:1$ between the first target involved in the conflict and its successors. If the conflict node is the last node before a conflict (where it is possible to perform a holding to avoid such conflict), the first target involved in a conflict is its successor. The holding vector $\hat{p}$ is obtained by following the route defined by its related route vector and waiting over each target for the first departure time step available that is conflict free. This means that in case of conflict the holding duration is defined by the first departure available, otherwise the holding is 0.
In the second case (minimization of the impact of the mission over the ATC) the Neighborhood of r_u, $N_2(r_u)$,is defined as follow. The route vector is defined as in the first case, instead the holding vector is defined solving the related HAP. Precisely it is obtained by following the route defined by the route vector and once a conflict is found the related holding is quantified and the related HAP is solved.

A Local Search (Best Improvement) is described in table 6.1.

The algorithm in case of minimization of the impact of the mission over the ATC is obtained by the previous by replacing N_1 with N_2.

6.2 Nearest neighbor

The nearest neighbor algorithm was implemented for the case of the minimization of the impact of the ATC over the mission. As in the previous case, two vectors are used. The

Local Search Algorithm

t: time
r_u: current route,defined by h and p
h route vector ($m+1$ elements)
p holding vector (m elements)
r'_u: generic route of the neighborhood
r_{TSP}:TSP route
r^b_u:best route

BEGIN
  Initalize: $r_u = r^b_u = r_{TSP}$**,** $t = 0$, $FIND = TRUE$**,** $CNF = FALSE$**;**
  do:
    $FIND = FALSE$;
    $r^b_u = r_u$;
    for $(i = 0; i < m; i++)$;
      if $(w_{h(i),h(i+1)}(t + p(i)) = f_{h(i),h(i+1)})$ **then** $t = t + f_{h(i),h(i+1)}$;
      else $CNF = TRUE$, $i_c = i$ **break for;**
    if $CNF = TRUE$ **then**
      for each $r'_u \in N^1(r_u)$ **do:**
        if $\varphi(r_u) \leq \varphi(r^b_u)$ **then** $r^b_u = r'_u$
    if $r^b_u <> r_u$ **then** $FIND = TRUE$
  while $(FIND = TRUE)$;
END

Table 6.1: Local Search Pseudocode

first represents the route and the second the holding. Initially these vectors are void; the algorithm starts initializing the route vector with the airport that represents the starting and ending target. The solution is then iteratively built, searching for the nearest target, that is the target that requires the minimum time to be reached considering the flight time and the holding time. Once a target is inserted in the route vector, the related holding to reach it is inserted in the holding vector. The algorithm is reported in the pseudocode reported in table 6.2.

Nearest Neighbor Algorithm

t: time
r_u: current route,defined by h and p h route vector ($m+1$ elements)
p holding vector (m elements)
it: ieration
r_u^b: best route

BEGIN
  Initalize: $h(0) = 0$**,**$p(0) = 0$**,** $it = 0$**,** $t = 0$**;**
  do:
    find k^* **such that** $w_{h(i),k^*}(t) = min\{w_{h(i),k^*}(t), k \notin h\}$;
    if $w_{h(i),k^*}(t) > f_{h(i),k^*}$ **then** $p(it) = w_{h(i),k^*}(t) - f_{h(i),k^*}$**;**
    else $p(it) = 0$**;**
    $it = it + 1$;
    $t = t + w_{h(i),k^*}(t)$;
    $h(it) = k^*$;
  while $(it < (m-1))$;
END

Table 6.2: Nearest Neighbor Pseudocode

Chapter 7

Metaheuristic Approach for URP

The algorithms presented in this chapter consider only the case of minimization of the ATC impact over the mission.

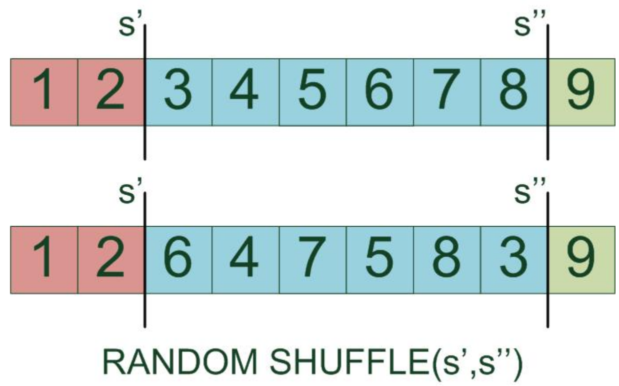
7.1 Genetic Algorithm

The algorithm is initialized by randomly generating a first population of η_p UAS routes R_u. Each route can be represented by two chromosomes : the first one consists of m+1 elements and represents the sequence of targets ("route chromosome"), the second one of m elements and represents the holdings ("holdings chromosome"). Since only the objective of the minimization of the ATC over the mission is considered, the holding chromosome depends on the route chromosome. In fact, for a given route, its related holding chromosome is automatically identified by following that route, and performing a conflict check at each target. If a conflict occurs, the related holding consists in the wait until the first available departure time step. The same consideration cannot be extended if we had considered the objective of the minimization of the mission impact over the ATC.

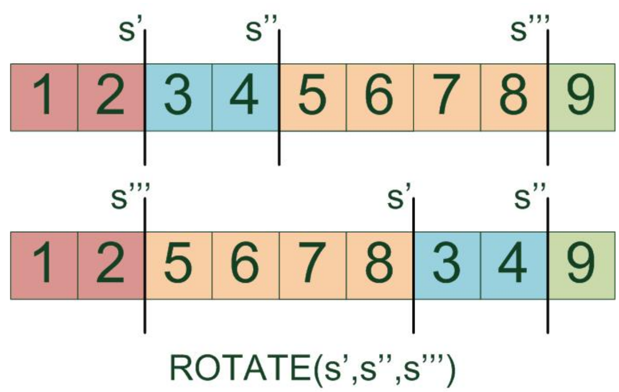
In this way, the fitness of each route can be simply evaluated considering that for a given sequence of targets, the mission duration is the sum of the value that the function $w_{i,j}(t)$ assumes following that sequence.
After the initialization, the population is sorted by decreasing fitness. The evolution to the next population then starts. This process is repeated until the STOP criterion is satisfied. It consists in a bound on the times when the evolution is repeated or in a bound on the improvement of the last η_C generations .

The evolution to the next population is carried out by dividing the previous sorted population into two parts. The first part is considered "good" and thus it is not changed. Instead the second one is rebuilt by combining the chromosomes of the first half. For this purpose, the criterion used is such that the $i-th$ individual is used to rebuild the $(i+(\eta p/2))-th$ individual.
A typical problem of the evolutionary operators for TSP is that infeasible routes can be produced (for example it can contain a sub-tour). Thus the route chromosome is evolved using the following mutation operators that make it possible to obtain feasible solutions: Random Shuffle, Rotate and Reverse.

Random Shuffle(s', s'') consists in a random exchange of the genes comprising the two cuts of



the chromosome s' and s''.



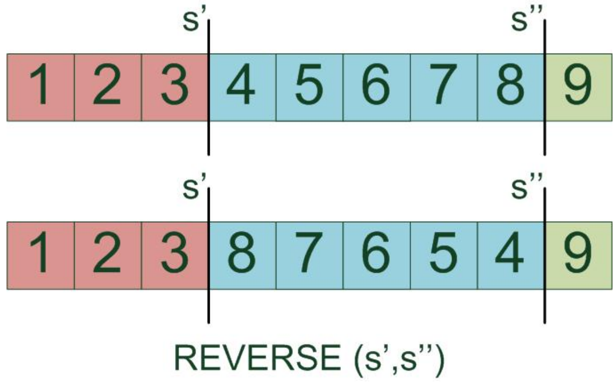
Rotate(s', s'', s''') exchanges the parts of the route chromosome between s' and s'' with the part between s'' and s'''.

Finally, Reverse(s', s'') simply reverses the part of the chromosome between s' and s''.

The Genetic Algorithm is described in the pseudocode reported in table 7.1.

7.2 Simulated Annealing

The algorithm starts by a randomly generated UAS route r_u. The annealing process then starts from an initial temperature T_0 and terminates at a temperature T_{min}. At each iteration the temperature is reduced according to a defined cooling rate. The generation of the new route r'_u is performed as follows. Two indices of the route vector of r_u are randomly chosen and inverted. In this case too, the holding vectors is obtained by following the route defined in the route vector and performing the appropriate holdings (we consider only the minimization of the ATC impact over the mission). The fitness of the new route is found as in the Genetic Algorithm. This fitness is compared with the fitness of the route r_u. If an improvement is developed, the solution is updated. Otherwise, depending on the current temperature, the new route r_u can be refused or accepted. One of the peculiarities of the simulated annealing algorithms is that they can accept not only improving solution in order to perform a better



Genetic Algorithm

η_p: Number of individual in the population
η_t: Number of generation
$imp(\eta_c)$: improvement in last η_c generations
gen: generation

BEGIN
  Generation of the first population of η_p UAS routes
  Sort population by decreasing fitness, $gen = 1$, $STOP = FALSE$;
  while $(STOP = FALSE)$ **do**
    Evolve the population
    Sort the population by decreasing fitness;
    $gen = gen + 1$;
    if$(gen \leq \eta_t)$ **or** $imp(\eta_c) \leq 0.01$ **then** $(STOP = TRUE)$
  EndWhile;
END

Table 7.1: Genetic Algorithm Pseudocode

exploration of the solution space. The Simulated Annealing Algorithm is described in the pseudocode in table 7.2.

Simulated Annealing Algorithm

T_0: Initial temperature
c_r: Cooling Rate
T_{min}: Final Temperature
T: Temperature

BEGIN
  Generation of initial route r_u **with fitness** $\varphi(r_u)$**,** $T = T_0$;
  while $T > T_0$ **do**
    Generation of r'_u **from** r_u;
    $\Delta_\varphi = \varphi(r'_u) - \varphi(r_u)$;
    Generation of a random number $rand$ **in** $[0, 1]$;
    if($\Delta_\varphi \leq 0$ **or(** $\Delta_\varphi \geq 0$ **and** $e^{-\Delta_\varphi / rand} > rand$**)) then** $r_u = r'_u$**;**
    $T = T \cdot c_r$;
  EndWhile;
END

Table 7.2: Simulated Annealing Algorithm Pseudocode

Chapter 8

Milano Linate Air Traffic Scenario

8.1 The mission

We tested our algorithms on a real air traffic scenario: the TMA (Terminal Manoeuvring Area) of Milano Linate (ICAO code LIML), an important airport in the North of Italy with an average of 450 (air)movements per day. In this Controlled Air Space, Navigation Points such as Radio-Assistance and Fix Points (radial and distance by radio assistance) are reported; SID (Standard Instrumental Departure) routes and STAR (STandard arrival Route) of the airport are modeled using graphic tools. Our simulator is presented in the figure 8.1.

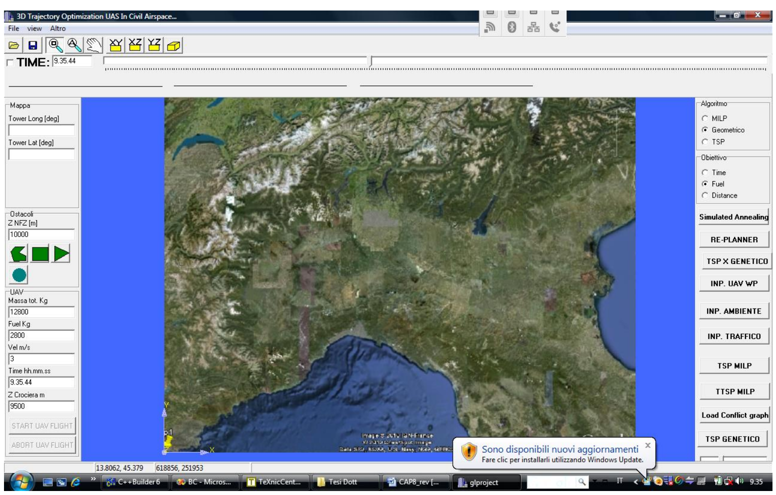


Figure 8.1: Air Traffic Simulator

It is possible to recognize the satellite chart that represents the North-West of Italy. On the right of the chart, some buttons make it possible to manage the simulator. It is possible to insert the Air Navigation Points and the Air Traffic Data. Figure 8.2 shows the Air Navigation Points used. To model some Radar Vectoring techniques, fictitious points are inserted. For example, the point LIN does not exist as a navigation point on Aeronautical Charts, nevertheless it makes it possible to model the Radar Downwind and the Radar Base of the west Radar circuit of Linate.

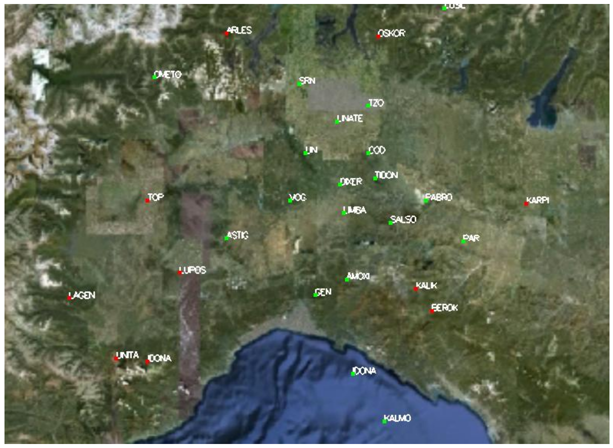


Figure 8.2: Air Navigation Points

The air traffic data and the Radar Tracks were provided by ENAV S.p.A. They represent the position and the altitude of departing, arriving and overflying aircrafts during five different days. The days considered were: the 11th of November 2009 from 05:30 until 13:30 UTC, the 22-nd of August 2010 from 04:30 to 12:30 UTC, the 25th of August 2010 from 04:30 to 12:30 UTC, the 25th of August 2010 from 12:30 to 20:30 UTC and the 26th of August from 04:30 to 12:30 UTC. The aircraft routes are represented by green segments between two Navigation Points. The position of an aircraft is represented by a red square that follows its route. The aircraft name and its altitude are represented through a related label. The position and the altitude are updated at each time step. Figure 8.3 shows an example of our model of the air traffic over Linate Area.

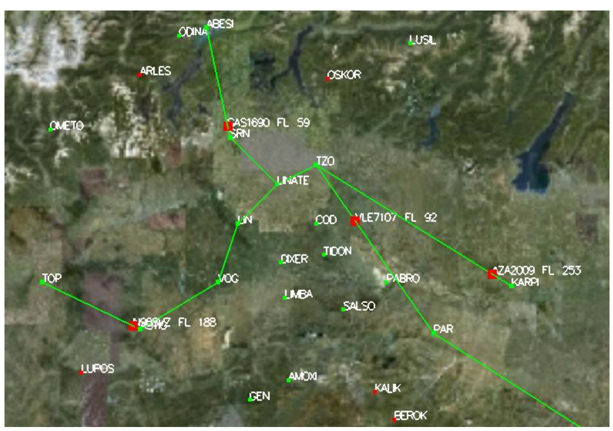


Figure 8.3: Air Traffic Simulation

This is also an example of the Aircraft Graph that is the tool used to model the air traffic during the mission. Moreover, using the button on the right it is possible to insert the UAS target points. The position of these points is generated as follows: one part (60 percent) is randomly generated, while another part (40 percent) corresponds to the Air Navigation Points. We considered two types of mission: a medium-short and a medium-long. The first one consists in visiting up to 20 targets in a range of 20NM (Nautical Miles) from Linate. The second one consists in visiting up to 40 targets in a range of 80NM from Linate. Note that in this case it was necessary to also model the air traffic of the Controlled Zones near Linate, such as Milano Malpensa, Bergamo, Torino and Genova. The costs of the holdings are assigned using the following considerations: if 1 is the cost of the flight phase ($\alpha = 1$), a ground holding (in the airport) cost is equal to 2. Instead, the airborne holdings have a decreasing cost according to their distance from the airport. The basic idea is that a UAS in holding near the airport requires a more difficult management than a UAS holding far from the airport because that zone is more sensitive since the traffic is more concentrated .

As example of the simulation, the 20 targets mission Medium-Long of the 11 November 2009 is reported in the following figures. In Figure 8.4 it is shown the UAS route (red line) if the mission would have performed in segregated air space. In this case the time required to the UAS to overfly all 20 targets is 162 minutes (TSP optimal solution).
In Figure 8.5 is reported the same mission in controlled air space obtained using the MIM Algorithm. The duration of the mission in controlled air space is 170 minutes: a rerouting make it possible to avoid the piloted air traffic.
An interesting example of UAS conflict resolution in such mission is reported in Figures 8.6 - 8.9 that have to be observed in sequence.

Figure 8.4: TSP

The point AMOXI, LIMBA and DIXER are lined up to Runway 36 of Linate, arriving aircraft coming from South follow this route. Looking the pictures in sequence it is possible to

Figure 8.5: The route in Controlled Air Space

recognize an arrival sequence AZA2036, ACL324 and AZA2032. The UAV route includes the path between DIXER and LIMBA in opposite direction. This path is yet performed by the UAV between ACL324 and AZA2032 without separation minima infringement.

The tables 8.1-8.11 report the main characteristics of the instance considered. The name of an instance is comprised by: the ICAO code of the Airport (LIML), the date and the mission departure time. The other characteristics are: the number of targets visited(NWP), the value of the optimal solution of the associated TSP route (OBJ TSP) in segregated air space (Lower Bound on the value of the optimal solution in Controlled Air Space), the number of aircrafts simulated (N acf), the number of conflicts over the TSP route(N cnf TSP), the value of the optimal solution of the associated TSP route (OBJ HTSP) in controlled airspace (Upper Bound on the value of the optimal solution). Finally it is reported the percentage gap between these bounds (GAP). As explained before, the value of the optimal solution of the TSP in controlled air space (HTSP) is obtained by following the TSP route and by performing the appropriate holdings to avoid the conflicts until the first available departure time step. Note that the conflicts over the TSP route are only a subset of all the possible conflict that exist between the UAS Graph and the Aircraft Graph.
Finally in tables 8.12 and 8.13 are reported the TSP route of the missions. All the algorithm have been tested on a 3.0 GHz Pentium 4 computer with 1 GB of memory.

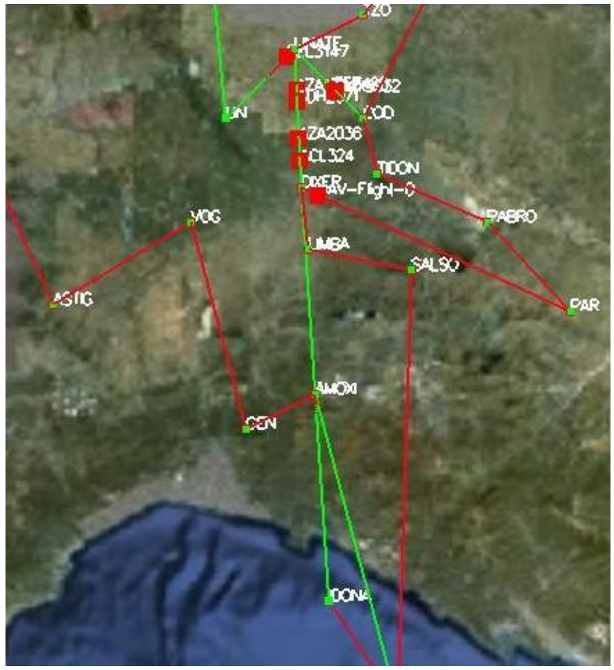


Figure 8.6: Example of Avoidance (1)

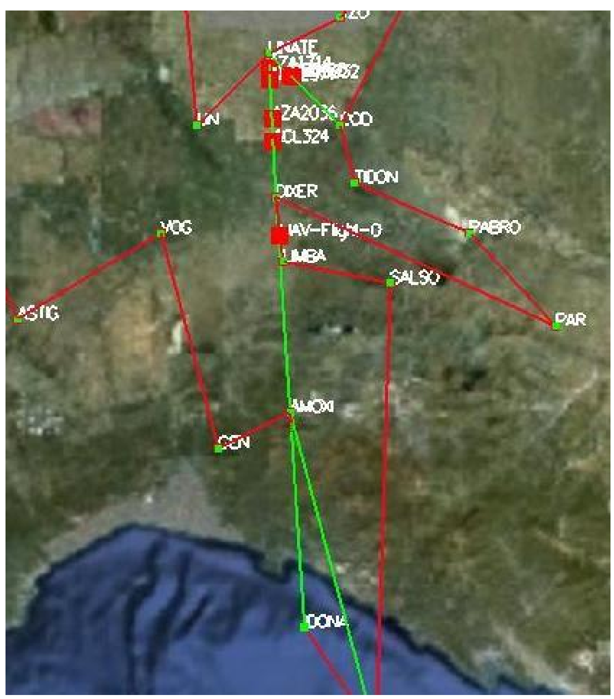


Figure 8.7: Example of Avoidance (2)

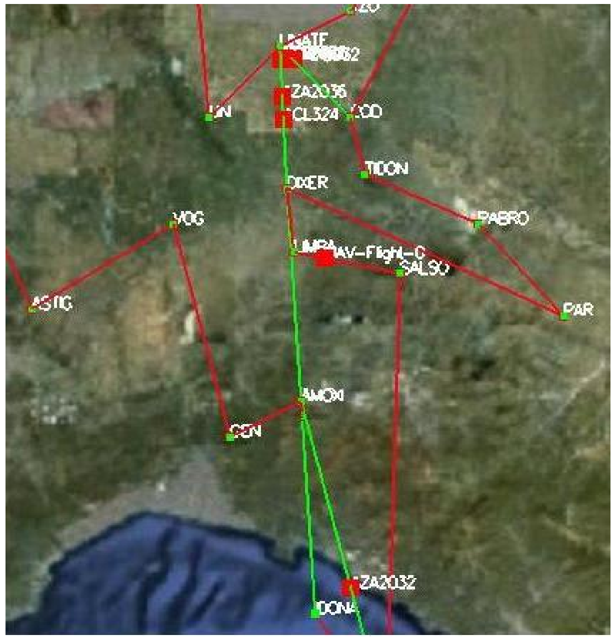


Figure 8.8: Example of Avoidance (3)

Figure 8.9: Example of Avoidance (4)

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	8	52	30	1	63	17.46
LIML 22/08 04:30	8	52	27	3	67	22.38
LIML 25/08 04:30	8	52	37	3	63	17.46
LIML 25/08 12:30	8	52	33	2	67	22.38
LIML 26/08 04:30	8	52	36	2	63	17.46

Table 8.1: 8 Targets Mission, Medium-Long Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	10	55	32	2	68	19.11
LIML 22/08 04:30	10	55	28	3	68	19.11
LIML 25/08 04:30	10	55	40	3	63	12.69
LIML 25/08 12:30	10	55	34	3	80	31.25
LIML 26/08 04:30	10	55	38	3	68	19.11

Table 8.2: 10 Targets, Medium-Long Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	15	98	68	1	117	16.23
LIML 22/08 04:30	15	98	61	2	119	17.64
LIML 25/08 04:30	15	98	74	2	100	2
LIML 25/08 12:30	15	98	71	2	117	16.23
LIML 26/08 04:30	15	98	72	1	106	7.54

Table 8.3: 15 Targets, Medium-Long Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	20	162	123	4	189	14.28
LIML 22/08 04:30	20	162	117	4	175	7.42
LIML 25/08 04:30	20	162	131	4	192	15.62
LIML 25/08 12:30	20	162	127	4	220	26.36
LIML 26/08 04:30	20	162	125	3	204	20.58

Table 8.4: 20 Targets, Medium-Long Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	25	176	211	5	190	7.36
LIML 22/08 04:30	25	176	204	2	204	13.72
LIML 25/08 04:30	25	176	216	4	192	8.33
LIML 25/08 12:30	25	176	212	6	223	21.07
LIML 26/08 04:30	25	176	214	3	204	13.72

Table 8.5: 25 Targets, Medium-Long Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	30	179	213	5	214	16.35
LIML 22/08 04:30	30	179	207	4	207	13.52
LIML 25/08 04:30	30	179	220	3	203	11.82
LIML 25/08 12:30	30	179	214	5	221	19
LIML 26/08 04:30	30	179	216	4	205	12.68

Table 8.6: 30 Targets, Medium-Long Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	35	191	229	6	225	15.11
LIML 22/08 04:30	35	191	226	4	236	19.06
LIML 25/08 04:30	35	191	271	4	216	24.5
LIML 25/08 12:30	35	191	245	7	253	11.57
LIML 26/08 04:30	35	191	250	2	228	16.22

Table 8.7: 35 Targets, Medium-Long Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	40	199	247	9	228	12.71
LIML 22/08 04:30	40	199	232	6	262	24
LIML 25/08 04:30	40	199	282	6	220	9.54
LIML 25/08 12:30	40	199	257	10	226	11.94
LIML 26/08 04:30	40	199	261	4	232	14.22

Table 8.8: 40 Targets, Medium-Long Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	10	29	32	2	43	32.55
LIML 22/08 04:30	10	29	28	0	29	0
LIML 25/08 04:30	10	29	40	5	42	30.95
LIML 25/08 12:30	10	29	34	1	34	14.7
LIML 26/08 04:30	10	29	38	2	58	50

Table 8.9: 10 Targets, Medium-Short Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	15	31	68	3	38	18.42
LIML 22/08 04:30	15	31	61	1	37	16.21
LIML 25/08 04:30	15	31	74	1	34	8.82
LIML 25/08 12:30	15	31	71	2	38	18.42
LIML 26/08 04:30	15	31	72	1	37	16.21

Table 8.10: 15 Targets, Medium-Short Mission

Instance Name	NWP	ObjTSP	N acf	N cnf TSP	ObjHTSP	GAP%
LIML 11/11 05:30	20	41	123	1	52	21.15
LIML 22/08 04:30	20	41	117	3	49	16.32
LIML 25/08 04:30	20	41	131	3	63	34.92
LIML 25/08 12:30	20	41	127	0	41	0
LIML 26/08 04:30	20	41	125	3	67	38.8

Table 8.11: 20 Targets, Medium-Short Mission

M-S MISSION	TSP ROUTE
10 TARGET	0 5 2 6 8 9 7 4 3 1 0
15 TARGET	0 1 10 12 2 11 6 8 9 7 5 4 13 14 3 0
20 TARGET	0 3 14 18 13 17 19 1 10 12 2 11 6 15 8 9 5 16 7 4 0

Table 8.12: Medium-Short Mission TSP route

M-L MISSION	TSP ROUTE
8 TARGET	0 3 4 5 7 6 2 1 0
10 TARGET	0 3 4 8 9 7 6 5 2 1 0
15 TARGET	0 3 4 13 14 11 12 7 6 10 9 8 5 2 1 0
20 TARGET	0 2 5 10 6 7 16 12 15 11 9 8 4 13 14 17 18 3 19 1 0
25 TARGET	0 2 5 10 6 21 7 16 12 15 11 9 8 4 13 14 20 17 23 18 3 22 19 24 1 0
30 TARGET	0 2 5 10 28 6 21 7 16 12 15 11 9 29 8 4 25 13 14 20 17 23 18 27 3 22 19 26 24 1 0
35 TARGET	0 31 1 24 26 22 19 21 7 28 6 10 16 12 15 11 9 8 29 5 2 34 32 4 25 13 14 20 17 23 18 27 30 3 33 0
40 TARGET	0 31 1 35 24 26 22 36 37 19 38 21 7 28 6 10 16 12 15 11 9 8 29 5 2 34 32 4 25 13 14 20 17 23 18 27 39 30 3 33 0

Table 8.13: Medium-Long Mission TSP route

Chapter 9

Results of Algorithms based on TSPP

The tables below (from 9.1 to 9.5) report the results of the MIM Algorithm. This algorithm is based on the reformulation of the problem through a TSP with Penalty (TSPP). Using this algorithm it is possible to solve all the instances, both medium-short and medium-long missions. For each mission is reproted the value of the objective function OBJ, the CPU time and the number of iterations IT. It corresponds to the number of penalty constraints added. These iterations vary consistently according to the traffic condition. In some instances only two iterations are necessary, while in other instances up to three thousand iterations are required.
In tables 9.7, 9.8, 9.9, and 9.10 are reported the resulting routes. Note that the $i-th$ holding in the holding vector correspond to the holding over the target $i-th$ and not to the $i-th$ element of the route vector.

The tables 9.6 and 9.11 report some results of MAM Algorithm: only the 20 targets and 25 targets missions are considered. Note that the Min impact over the ATC Algorithm requires a longer computing time due to the HAP that is solved at each iteration. Thus only the results for the instances up to 25 targets are reported. Note that the different costs of the holdings produce the following consequences. If the Min impact over the mission Algorithm finds an optimal solution without holdings, this solution is valid also for the Min impact over the ATC Algorithm. Otherwise this algorithm searches for a rerouting solution that implies no holdings or a solution with holdings over the cheaper targets.

Instance Name	8 TARGETS			10 TARGETS		
	OBJ	CPU	IT	OBJ	CPU	IT
LIML 11/11 05:30	58	34.85	112	62	28.4	105
LIML 22/08 04:30	56	3.615	37	57	0.293	5
LIML 25/08 04:30	54	0.822	14	55	0.147	2
LIML 25/08 12:30	59	20.061	89	64	62.694	182
LIML 26/08 04:30	56	6.474	64	57	0.815	18

Table 9.1: Min impact over the mission: 8 and 10 Targets, Medium-Long Mission

Instance Name	15 TARGETS			20 TARGETS		
	OBJ	CPU	IT	OBJ	CPU	IT
LIML 11/11 05:30	101	1.871	19	170	928.29	710
LIML 22/08 04:30	102	1.78	17	165	4.12	21
LIML 25/08 04:30	99	0.121	2	166	11.762	35
LIML 25/08 12:30	105	81.11	145	171	695.557	541
LIML 26/08 04:30	99	0.41	5	164	1.231	12

Table 9.2: Min impact over the mission: 15 and 20 Targets, Medium-Long Mission

Instance Name	25 TARGETS			30 TARGETS		
	OBJ	CPU	IT	OBJ	CPU	IT
LIML 11/11 05:30	185	946.88	689	189	4518.1	1146
LIML 22/08 04:30	184	144.916	135	188	802.97	546
LIML 25/08 04:30	183	25.812	23	185	45.12	81
LIML 25/08 12:30	185	894.59	543	192	254959	2392
LIML 26/08 04:30	183	122.858	121	188	922.95	629

Table 9.3: Min impact over the mission: 25 and 30 Targets, Medium-Long Mission

Instance Name	35 TARGETS			40 TARGETS		
	OBJ	CPU	IT	OBJ	CPU	IT
LIML 11/11 05:30	196	326.359	181	209	123327.4	2931
LIML 22/08 04:30	194	29.204	48	206	18408.3	949
LIML 25/08 04:30	198	11546.3	771	206	17140.3	937
LIML 25/08 12:30	194	11.078	22	204	113.985	97
LIML 26/08 04:30	197	2485.02	450	206	49235.2	867

Table 9.4: Min impact over the mission: 35 and 40 Targets, Medium-Long Mission

Instance Name	10 TARGETS			15 TARGETS			20 TARGETS		
	OBJ	CPU	IT	OBJ	CPU	IT	OBJ	CPU	IT
LIML 11/11 05:30	30	1.003	12	32	0.125	2	41	0.194	2
LIML 22/08 04:30	29	0.094	1	34	27.492	62	43	96.07	115
LIML 25/08 04:30	29	0.596	8	34	17.417	49	42	2.079	9
LIML 25/08 12:30	31	8.371	44	33	1.391	9	41	0.13	1
LIML 26/08 04:30	32	32.35	99	34	30.272	66	42	0.493	3

Table 9.5: Min impact over the mission: 10,15 and 20 Targets, Medium-Short Mission

Instance Name	20 TARGETS			25 TARGETS		
	OBJ	CPU	IT	OBJ	CPU	IT
LIML 11/11 05:30	171	4486	346	186	3961.88	456
LIML 22/08 04:30	165	4.12	21	185	1111.916	235
LIML 25/08 04:30	166	11.762	35	184	130.812	64
LIML 25/08 12:30	172	3545	352	185	894.59	543
LIML 26/08 04:30	164	1.231	12	183	122.858	121

Table 9.6: Min impact over the ATC: 20 and 25 Targets, Medium-Long Mission

Instance Name	ROUTE 25 TARGETS
LIML 11/11 05:30	0 3 1 2 5 10 6 21 7 16 12 15 11 9 8 4 13 14 20 17 23 18 22 19 24 0
LIML 22/08 04:30	0 1 24 19 22 2 5 10 6 21 7 16 12 15 11 9 8 4 13 14 20 17 23 18 3 0
LIML 25/08 04:30	0 4 13 14 20 17 23 18 3 22 19 24 1 2 5 10 6 21 7 16 12 15 11 9 8 0
LIML 25/08 12:30	0 3 4 13 9 2 5 8 10 6 21 7 16 12 11 15 14 20 17 23 18 22 19 24 1 0
LIML 26/08 04:30	0 3 4 8 5 10 6 21 7 16 12 15 11 9 13 14 20 17 23 18 22 19 24 1 2 0
Instance Name	ROUTE 30 TARGETS
LIML 11/11 05:30	0 4 25 8 29 2 5 10 28 6 21 7 16 12 15 11 9 13 14 20 17 18 23 27 3 22 19 26 24 1 0
LIML 22/08 04:30	0 4 25 13 14 20 17 23 18 27 3 22 19 26 24 1 2 29 5 10 28 6 21 7 16 12 15 11 9 8 0
LIML 25/08 04:30	0 4 25 13 14 20 17 23 18 27 3 19 22 26 24 1 2 5 10 28 6 21 7 16 12 15 11 9 8 29 0
LIML 25/08 12:30	0 29 8 4 25 13 14 20 17 23 18 27 3 19 22 26 24 1 2 5 10 28 6 21 7 16 12 15 11 9 0
LIML 26/08 04:30	0 4 25 13 14 20 17 23 18 27 3 22 19 26 24 1 2 29 5 10 28 6 21 7 16 12 15 11 9 8 0
Instance Name	ROUTE 35 TARGETS
LIML 11/11 05:30	0 33 32 4 25 13 8 34 2 5 29 9 10 28 6 21 7 16 12 11 15 14 20 17 23 18 27 30 3 22 19 26 24 1 31 0
LIML 22/08 04:30	0 32 4 25 13 14 20 17 23 18 27 3 30 33 31 1 24 26 22 19 21 7 28 6 10 16 12 15 11 9 8 29 5 2 34 0
LIML 25/08 04:30	0 32 4 25 13 14 20 17 23 18 27 3 30 33 31 1 24 26 22 19 21 7 28 6 10 16 12 15 11 9 8 29 5 2 34 0
LIML 25/08 12:30	0 32 4 25 13 14 20 17 23 18 27 30 3 33 31 22 19 26 24 1 2 5 10 28 6 21 7 16 12 15 11 9 29 8 34 0
LIML 26/08 04:30	0 32 4 25 13 14 20 17 23 18 27 3 30 33 31 1 24 22 26 19 21 7 28 6 10 16 12 15 11 9 8 29 5 2 34 0
Instance Name	ROUTE 40 TARGETS
LIML 11/11 05:30	0 2 34 33 25 13 14 20 17 23 18 27 39 30 3 31 35 24 26 22 36 37 19 38 6 21 7 28 10 16 12 15 11 9 8 1 29 5 4 34 0
LIML 22/08 04:30	0 34 2 8 29 5 10 28 6 7 21 38 19 37 36 22 26 24 35 1 31 33 3 30 39 27 18 23 17 20 14 15 16 12 11 9 13 25 4 32 0
LIML 25/08 04:30	0 32 4 25 13 14 20 17 23 18 27 3 39 30 33 31 1 35 24 26 22 36 37 19 38 21 7 6 28 10 16 12 15 11 9 8 29 5 2 34 0
LIML 25/08 12:30	0 32 4 25 13 14 20 17 23 18 27 39 30 3 33 31 1 35 24 26 22 36 37 19 38 6 21 7 28 10 16 12 15 11 9 8 29 5 2 34 0
LIML 26/08 04:30	0 2 34 8 29 5 10 28 6 7 21 38 19 37 36 22 26 24 35 1 31 33 3 30 39 27 18 23 17 20 14 15 16 12 11 9 13 25 4 32 0

Table 9.7: Min impact over the mission: Routes for 25,30,35 and 40 Targets, Medium-Long Mission

Instance Name	ROUTE 8 TARGETS	HOLDING
LIML 11/11 05:30	0 1 7 6 5 2 4 3 0	4 0 0 0 0 0 0 0
LIML 22/08 04:30	0 1 2 6 7 5 4 3 0	0 0 0 0 0 0 2 0
LIML 25/08 04:30	0 1 2 6 7 5 4 3 0	0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 4 3 1 7 6 5 2 0	0 4 0 0 0 0 0 0
LIML 26/08 04:30	0 1 2 6 7 5 4 3 0	0 0 0 0 0 0 2 0
Instance Name	ROUTE 10 TARGETS	HOLDING
LIML 11/11 05:30	0 3 1 2 5 6 7 9 8 4 0	0 0 0 0 0 0 0 0 7 0
LIML 22/08 04:30	0 1 2 5 6 7 9 8 4 3 0	0 0 0 0 0 0 2 0 0 0
LIML 25/08 04:30	0 1 2 5 6 7 9 8 4 3 0	0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 2 5 6 7 9 8 4 3 1 0	0 0 5 3 0 0 0 0 0 0
LIML 26/08 04:30	0 1 2 5 6 7 9 8 4 3 0	0 0 0 0 0 0 2 0 0 0
Instance Name	ROUTE 15 TARGETS	HOLDING
LIML 11/11 05:30	0 4 8 9 13 14 11 12 7 6 10 5 2 1 3 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 1 3 4 8 9 13 14 11 12 7 6 10 5 2 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 3 4 8 9 13 14 11 12 7 6 10 5 2 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 3 1 2 9 13 14 12 11 7 6 10 5 8 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 3 4 8 9 13 14 11 12 7 6 10 5 2 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Instance Name	ROUTE 20 TARGETS	HOLDING
LIML 11/11 05:30	0 3 18 17 14 13 4 5 10 6 7 16 12 15 11 9 8 2 1 19 0	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 4 2 5 10 6 7 16 12 15 11 9 8 13 14 17 18 3 19 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 3 17 18 19 1 2 5 10 6 7 16 12 15 11 9 8 13 14 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 3 1 19 18 17 14 13 8 2 5 10 6 7 16 12 15 11 9 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0
LIML 26/08 04:30	0 4 8 2 5 10 6 7 16 12 15 11 9 13 14 17 18 3 19 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Table 9.8: Min impact over the mission: Routes and Holdings for 8,10,15 and 20 Targets, Medium-Long Mission

Instance Name	HOLDING 25 TARGETS
LIML 11/11 05:30	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0
LIML 25/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 1 0 0 0 0 0 0
LIML 25/08 12:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Instance Name	HOLDING 30 TARGETS
LIML 11/11 05:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
LIML 22/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0
LIML 25/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0
LIML 26/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0
Instance Name	HOLDING 35 TARGETS
LIML 11/11 05:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 0 1 0 0 0 0 0 0 0 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Instance Name	HOLDING 40 TARGETS
LIML 11/11 05:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Table 9.9: Min impact over the mission: Holdings for 25,30,35 and 40 Targets, Medium-Long Mission

Instance Name	ROUTE 10 TARGETS	HOLDING
LIML 11/11 05:30	0 3 1 2 6 8 9 5 7 4 0	0 0 0 1 0 0 0 0 0 0
LIML 22/08 04:30	0 5 2 6 8 9 7 4 3 1 0	0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 3 1 2 6 8 9 7 5 4 0	0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 3 1 2 6 8 9 7 5 4 0	0 1 0 1 0 0 0 0 0 0
LIML 26/08 04:30	0 1 3 4 9 8 7 6 2 5 0	0 1 0 0 0 0 0 0 0 0
Instance Name	ROUTE 15 TARGETS	HOLDING
LIML 11/11 05:30	0 4 13 14 3 1 10 12 2 11 6 8 9 7 5 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 1 10 12 2 11 6 8 9 7 5 3 14 13 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 1 10 12 2 11 6 8 9 7 5 4 13 14 3 0	0 0 0 0 0 0 0 0 0 0 0 0 0 3 0
LIML 25/08 12:30	0 5 1 10 12 2 11 6 8 9 7 4 13 14 3 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 1 10 12 2 11 6 8 9 7 5 3 14 13 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Instance Name	ROUTE 20 TARGETS	HOLDING
LIML 11/11 05:30	0 3 14 18 13 17 19 1 10 12 2 15 11 6 8 9 7 16 5 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 3 18 13 4 14 17 19 1 10 12 2 11 6 15 16 9 8 7 5 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 5 7 9 8 16 11 6 15 2 12 10 1 19 17 3 18 13 14 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 3 18 13 14 17 19 1 10 12 2 11 6 15 7 5 16 9 8 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 3 14 18 13 17 19 1 10 12 2 15 11 6 8 9 5 16 7 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

Table 9.10: Min impact over the mission: Ruotes and holdings for 10,15 and 20 Targets, Medium-Short Mission

Instance Name	ROUTE 20 TARGETS	HOLDING
LIML 11/11 05:30	0 3 8 9 11 15 12 16 7 6 10 5 2 1 19 18 17 14 13 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 4 2 5 10 6 7 16 12 15 11 9 8 13 14 17 18 3 19 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 3 17 18 19 1 2 5 10 6 7 16 12 15 11 9 8 13 14 4 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 3 4 13 14 17 18 19 1 2 5 6 7 10 16 12 15 11 9 8 0	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 4 8 2 5 10 6 7 16 12 15 11 9 13 14 17 18 3 19 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Instance Name	HOLDING 25 TARGETS	HOLDING
LIML 11/11 05:30	0 3 4 13 14 20 17 23 18 22 19 24 1 2 5 10 6 21 7 16 12 11 15 9 8 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 1 24 19 22 5 2 10 6 21 7 16 12 15 11 9 8 4 13 14 20 17 23 18 3 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 1 24 19 22 3 18 23 17 20 14 13 4 8 5 10 6 21 7 16 12 15 11 9 2 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 3 4 13 9 2 5 8 10 6 21 7 16 12 11 15 14 20 17 23 18 22 19 24 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 3 4 8 5 10 6 21 7 16 12 15 11 9 13 14 20 17 23 18 22 19 24 1 2 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Table 9.11: Min impact over the ATC: Routes and Holdings for 20 and 25 Targets, Medium-Long Mission

Chapter 10

Results of BC Algorithm

10.1 Min impact over the mission

The tables 10.1 and 10.2 show the main characteristics of the TDTSP formulation 5.1-5.15 of some instances. The columns NvarX and NvarY report the number of $x^t_{(i,j)}$ and y^t_i variables in the problem after the preprocessing phase. In the same way, the columns NCT1 and NCT2 report the number of temporal constraints in the model after the preprocessing. Note that due to the elimination of the variables some involved constraints are also eliminated. The column LP1 shows the value of the optimal solution of the LP-relaxation of the formulation 5.1-5.15. By comparing this value with the value of the associated TSP optimal solution it is possible to underline how, unfortunately, the bound obtained is weak. Column LP2 reports the value of the optimal solution of the LP-relaxation obtained by adding the inequalities 5.18. In this case the bound increases and overcomes the bound provided by the optimal solution of the TSP. Columns CPU LP1 and CPU LP2 show the computing time necessary to solve the LP-relaxation in both cases.

Table 10.3 and 10.4 reports the results of the Branch and Cut algorithm with the generation of the cuts. A maximum of 1200 seconds of CPU time was allowed for the solution of each instance. In this table we report the number of cuts (Cuts), the best upper bound obtained (UB), the CPU time in seconds (CPU), the number of nodes (Nodes) explored in the branch and cut tree and the best lower bound (LB) found within the CPU time if an instance could not be solved to optimality. The asterisk on the best upper bound obtained indicates that the algorithm was not able to find an integer solution within the CPU time. In this case the Upper Bound is represented by the Upper Bound provided by the heuristic algorithms (NN or HTSP). In the column GAP is reported the percentage gap between the UB and the LB.

Instance Name	NWP	NvarX	NvarY	NCT1	NCT2	LP1	CPU LP1	LP2	CPU LP2
LIML 11/11 05:30	8	278	1667	270	408	40	0.054	53	0.022
LIML 22/08 04:30	8	272	1620	264	385	39	0.053	54	0.02
LIML 25/08 04:30	8	279	1370	271	388	39	0.043	53	0.024
LIML 25/08 12:30	8	315	1754	307	459	39	0.067	53	0.024
LIML 26/08 04:30	8	257	1544	249	390	39	0.076	53	0.024
Instance Name	NWP	NvarX	NvarY	NCT1	NCT2	LP1	CPU LP1	LP2	CPU LP2
LIML 11/11 05:30	10	441	3412	431	609	47	1.037	56	0.036
LIML 22/08 04:30	10	442	3413	432	602	46	0.097	56	0.029
LIML 25/08 04:30	10	407	2698	397	561	46	0.105	55	0.028
LIML 25/08 12:30	10	457	3248	447	639	46	0.093	55	0.034
LIML 26/08 04:30	10	426	3244	416	607	46	0.098	55	0.031
Instance Name	NWP	NvarX	NvarY	NCT1	NCT2	LP1	CPU LP1	LP2	CPU LP2
LIML 11/11 05:30	15	1247	13997	1232	1605	86	0.553	98	0.114
LIML 22/08 04:30	15	1301	14113	1286	1628	87	0.579	99	0.106
LIML 25/08 04:30	15	1067	10785	1052	1353	88	0.409	98	0.092
LIML 25/08 12:30	15	1203	12981	1188	1596	86	0.434	98	0.101
LIML 26/08 04:30	15	1096	11350	1081	1438	88	0.504	98	0.086
Instance Name	NWP	NvarX	NvarY	NCT1	NCT2	LP1	CPU LP1	LP2	CPU LP2
LIML 11/11 05:30	20	2768	41465	2748	3527	135	1.713	163	0.345
LIML 22/08 04:30	20	2614	39775	2594	3240	136	1.389	163	0.305
LIML 25/08 04:30	20	2982	43733	2962	3592	135	1.747	163	0.355
LIML 25/08 12:30	20	2687	39335	2667	3461	135	1.55	163	0.311
LIML 26/08 04:30	20	2968	45182	2948	3747	135	2.647	163	0.44

Table 10.1: 8,10,15 and 20 Targets, Medium-Long Mission: TDTSP formulation characterisitcs

Instance Name	NWP	NvarX	NvarY	NCT1	NCT2	LP1	CPU LP1	LP2	CPU LP2
LIML 11/11 05:30	10	221	1869	211	264	26	0.047	29	0.025
LIML 22/08 04:30	10	222	1799	212	258	26	0.062	29	0.022
LIML 25/08 04:30	10	248	1861	238	307	26	0.047	29	0.026
LIML 25/08 12:30	10	225	1660	215	264	26	0.047	29	0.02
LIML 26/08 04:30	10	437	3891	427	548	26	0.125	29	0.04
Instance Name	NWP	NvarX	NvarY	NCT1	NCT2	LP1	CPU LP1	LP2	CPU LP2
LIML 11/11 05:30	15	414	5289	399	483	28	0.187	32	0.04
LIML 22/08 04:30	15	420	5291	405	490	28	0.343	32	0.042
LIML 25/08 04:30	15	402	4206	387	460	28	0.172	32	0.034
LIML 25/08 12:30	15	383	4522	368	459	28	0.125	32	0.032
LIML 26/08 04:30	15	402	5130	387	490	28	0.187	32	0.039
Instance Name	NWP	NvarX	NvarY	NCT1	NCT2	LP1	CPU LP1	LP2	CPU LP2
LIML 11/11 05:30	20	673	11123	653	743	37	0.375	41	0.066
LIML 22/08 04:30	20	652	10989	632	770	37	0.438	42	0.064
LIML 25/08 04:30	20	731	11264	711	868	36	0.266	42	0.068
LIML 25/08 12:30	20	670	9685	650	743	37	0.265	41	0.059
LIML 26/08 04:30	20	634	10697	614	770	37	0.25	42	0.144

Table 10.2: 10,15 and 20 Targets, Medium-Short Mission: TDTSP formulation characterisitcs

Instance Name	NWP	UB	CPU	Nodes	LB	Cuts	Gap%
LIML 11/11 05:30	8	58	27.993	583	58	9	0.00
LIML 22/08 04:30	8	56	3.711	68	56	3	0.00
LIML 25/08 04:30	8	54	3.61	13	54	5	0.00
LIML 25/08 12:30	8	59	17.245	797	59	11	0.00
LIML 26/08 04:30	8	56	2.958	31	56	4	0.00
Instance Name	NWP	UB	CPU	Nodes	LB	Cuts	Gap%
LIML 11/11 05:30	10	62	540.139	11225	62	29	0.00
LIML 22/08 04:30	10	57	3.773	4	57	7	0.00
LIML 25/08 04:30	10	55	4.456	1	55	2	0.00
LIML 25/08 12:30	10	64	86.561	1612	64	19	0.00
LIML 26/08 04:30	10	57	10.131	26	57	8	0.00
Instance Name	NWP	UB	CPU	Nodes	LB	Cuts	Gap%
LIML 11/11 05:30	15	114	1200.11	692	98	41	14.03
LIML 22/08 04:30	15	104	1200	2907	100	31	3.84
LIML 25/08 04:30	15	99	399.122	1061	99	24	0.00
LIML 25/08 12:30	15	117	1200.06	821	98	35	16.23
LIML 26/08 04:30	15	*122	1200.06	1173	98	34	8.16
Instance Name	NWP	UB	CPU	Nodes	LB	Cuts	Gap%
LIML 11/11 05:30	20	*189	1200.17	105	163	21	15.95
LIML 22/08 04:30	20	*175	1200.16	42	163	22	7.36
LIML 25/08 04:30	20	*192	1200.2	1	163	8	17.79
LIML 25/08 12:30	20	*187	1200.19	10	163	15	14.72
LIML 26/08 04:30	20	*201	1200.06	25	163	18	23.31

Table 10.3: 8,10,15 and 20 Targets, Medium-Long Mission: results of the Branch and Cut algorithm

Instance Name	NWP	UB	CPU	Nodes	LB	Cuts	Gap%
LIML 11/11 05:30	10	30	3.219	18	30	8	0.00
LIML 22/08 04:30	10	29	0.89	0	29	5	0.00
LIML 25/08 04:30	10	29	3.312	14	29	5	0.00
LIML 25/08 12:30	10	31	5.406	52	31	8	0.00
LIML 26/08 04:30	10	32	290.328	3158	32	24	0.00
Instance Name	NWP	UB	CPU	Nodes	LB	Cuts	Gap%
LIML 11/11 05:30	15	32	12.219	18	32	9	0.00
LIML 22/08 04:30	15	34	691.453	2750	34	36	0.00
LIML 25/08 04:30	15	34	16.141	163	34	18	0.00
LIML 25/08 12:30	15	33	55.938	319	33	25	0.00
LIML 26/08 04:30	15	34	23.281	65	34	13	0.00
Instance Name	NWP	UB	CPU	Nodes	LB	Cuts	Gap%
LIML 11/11 05:30	20	41	70.516	9	41	7	0.00
LIML 22/08 04:30	20	43	987.03	678	43	32	0.00
LIML 25/08 04:30	20	47	1200.13	1139	42	74	10.63
LIML 25/08 12:30	20	41	20.375	0	41	6	0.00
LIML 26/08 04:30	20	42	293.125	440	42	29	0.00

Table 10.4: 10,15 and 20 Targets, Medium-Short Mission: results of the Branch and Cut algorithm

10.2 Min impact over the mission

The tables 10.5, 10.6 and 10.7 show an example of the results for the minimization of the mission impact over the ATC. The example is provided by the 10 targets Medium-Long Mission. Tables 10.5 and 10.6 are similar to tables 10.1 and 10.3 respectively. The route of the instance LIML11/11 05:30 uses three holdings to avoid the air traffic: two minutes over target 5, one minute over target 7 and four minutes over target 8. Instead, in case of minimization of the impact over the ATC, each holding minute has a cost. Thus the UAS replans its route changing the targets visit order. The same happens for the instance LIML25/08 12:30. Instead in the instances LIML22/08 04:30 and LIML26/08 04:30 the UAS shifts its holdings in the most convenient target. Such target is the airport: the ground holding is preferred to the airborne holding. Generally it is important to note that in case of minimization of the impact over the ATC the UAS tries to avoid expensive holdings in two ways: rerouting or ground holding.

Instance Name	NWP	NvarX	NvarY	NCT1	NCT2	LP1	CPU LP1	LP2	CPU LP2
LIML 11/11 05:30	10	441	3412	431	609	47	0.078	56	0.033
LIML 22/08 04:30	10	442	3413	432	602	46	0.078	56	0.032
LIML 25/08 04:30	10	407	2698	397	561	46	0.079	55	0.027
LIML 25/08 12:30	10	457	3248	447	639	46	0.079	55	0.032
LIML 26/08 04:30	10	426	3244	416	607	46	0.093	55	0.031

Table 10.5: 10 Targets, Medium-Long Mission:results

Instance Name	NWP	UB	CPU	Nodes	LB	Cuts	Gap%
LIML 11/11 05:30	10	62	827.438	13662	62	51	0.00
LIML 22/08 04:30	10	59	38.875	809	59	21	0.00
LIML 25/08 04:30	10	55	1.109	0	55	1	0.00
LIML 25/08 12:30	10	67	249.078	3848	67	40	0.00
LIML 26/08 04:30	10	59	125.687	3682	59	30	0.00

Table 10.6: 10 Targets, Medium-Long Mission: Min impact over ATC

Min ATC impact over mission		
Instance Name	Route	Holding
LIML 11/11 05:30	0 3 1 2 5 6 7 9 8 4 0	0 0 0 0 0 2 0 1 4 0
LIML 22/08 04:30	0 1 2 5 6 7 9 8 4 3 0	0 0 2 0 0 0 0 0 0 0
LIML 25/08 04:30	0 1 2 5 6 7 9 8 4 3 0	0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 2 7 6 5 9 8 4 3 1 0	0 0 0 4 0 1 0 0 0 0
LIML 26/08 04:30	0 1 2 5 6 7 9 8 4 3 0	0 2 0 0 0 0 0 0 0 0
Min mission impact over ATC		
Instance Name	Route	Holding
LIML 11/11 05:30	0 3 1 2 5 7 6 8 9 4 0	0 0 0 0 0 0 0 0 0 0
LIML 22/08 04:30	0 1 2 5 6 7 9 8 4 3 0	2 0 0 0 0 0 0 0 0 0
LIML 25/08 04:30	0 1 2 5 6 7 9 8 4 3 0	0 0 0 0 0 0 0 0 0 0
LIML 25/08 12:30	0 7 6 5 9 8 4 3 2 1 0	0 0 0 0 0 0 0 0 0 0
LIML 26/08 04:30	0 1 2 5 6 7 9 8 4 3 0	2 0 0 0 0 0 0 0 0 0

Table 10.7: 10 Targets, Medium-Long Mission: Objectives comparison

Chapter 11

Results of Heuristic Approach

The tables below show the results of the heuristic algorithms proposed. These algorithms are used basically for two reasons. First they represent an upper bound on the value of the optimal solution of the problem and thus make it possible to reduce its time dimension. Moreover the computing times of these Nearest Neighbor and Local Search algorithms are very small(for that reason they have not been reported). To obtain a better Upper Bound it is therefore possible to use both algorithms considering the best value obtained. In each table is reported: the value of the solution of the Nearest Neighbor Algorithm(OBJ NN), its percentage gap (GAP NN) with the value of the optimal solution, the value of the solution of the Local Search Algorithm(OBJ LS), the number of exchange 1:1 performed (Move LS) and the percentage gap (GAP LS) with the value of the optimal solution.

Instance Name	8 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	59	1.72	63	0	8.62
LIML 22/08 04:30	57	1.78	64	1	14.28
LIML 25/08 04:30	56	3.7	58	1	7.4
LIML 25/08 12:30	66	11.86	67	0	13.55
LIML 26/08 04:30	57	1.78	63	0	12.5

Table 11.1: 8 Targets, Medium-Long Mission

Instance Name	10 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	86	38.7	67	1	8.06
LIML 22/08 04:30	80	40.35	64	2	12.28
LIML 25/08 04:30	90	63.63	63	0	14.54
LIML 25/08 12:30	73	14.06	70	3	9.37
LIML 26/08 04:30	85	49.12	67	1	17.54

Table 11.2: 10 Targets, Medium-Long Mission

Instance Name	15 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	128	26.73	105	1	3.96
LIML 22/08 04:30	122	19.6	111	3	8.82
LIML 25/08 04:30	126	27.27	100	0	1.01
LIML 25/08 12:30	127	20.95	110	4	4.76
LIML 26/08 04:30	122	23.23	104	1	5.05

Table 11.3: 15 Targets, Medium-Long Mission

Instance Name	20 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	197	15.88	189	0	11.17
LIML 22/08 04:30	202	22.42	175	0	6.06
LIML 25/08 04:30	204	22.89	176	1	6.02
LIML 25/08 12:30	187	9.35	194	3	13.45
LIML 26/08 04:30	201	22.56	181	2	10.36

Table 11.4: 20 Targets, Medium-Long Mission

Instance Name	25 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	220	18.91	190	0	2.7
LIML 22/08 04:30	221	20.1	204	0	10.86
LIML 25/08 04:30	220	20.21	192	0	4.9
LIML 25/08 12:30	232	25.4	223	0	20.54
LIML 26/08 04:30	211	15.3	204	0	11.47

Table 11.5: 25 Targets, Medium-Long Mission

Instance Name	30 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	229	21.16	214	0	13.22
LIML 22/08 04:30	233	25.94	207	0	11.89
LIML 25/08 04:30	215	14.36	203	0	7.97
LIML 25/08 12:30	216	14.89	221	0	17.55
LIML 26/08 04:30	211	9.89	205	0	6.77

Table 11.6: 30 Targets, Medium-Long Mission

Instance Name	35 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	232	18.36	218	3	11.22
LIML 22/08 04:30	253	30.41	236	0	21.64
LIML 25/08 04:30	244	25.77	216	0	11.34
LIML 25/08 12:30	253	27.77	220	2	11.11
LIML 26/08 04:30	244	23.85	225	2	14.21

Table 11.7: 35 Targets, Medium-Long Mission

Instance Name	40 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	259	23.92	228	0	9.09
LIML 22/08 04:30	280	35.92	251	5	21.84
LIML 25/08 04:30	263	27.66	220	0	6.79
LIML 25/08 12:30	263	28.92	224	1	9.8
LIML 26/08 04:30	250	21.35	230	1	11.65

Table 11.8: 40 Targets, Medium-Long Mission

Instance Name	10 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	31	3.33	43	0	43.33
LIML 22/08 04:30	51	75.86	29	0	0
LIML 25/08 04:30	35	20.68	38	1	31.03
LIML 25/08 12:30	31	0	34	0	9.67
LIML 26/08 04:30	63	96.85	58	0	81.25

Table 11.9: 10 Targets, Medium-Short Mission

Instance Name	15 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	37	15.62	36	1	12.5
LIML 22/08 04:30	37	8.82	37	0	8.82
LIML 25/08 04:30	40	17.64	34	0	0
LIML 25/08 12:30	36	9.09	33	1	0
LIML 26/08 04:30	37	8.82	37	0	8.82

Table 11.10: 15 Targets, Medium-Short Mission

Instance Name	20 TARGETS				
	OBJ NN	GAP NN	OBJ LS	Move LS	GAP LS
LIML 11/11 05:30	42	2.43	41	1	0
LIML 22/08 04:30	43	0	46	1	6.97
LIML 25/08 04:30	48	14.28	50	1	19.04
LIML 25/08 12:30	42	2.43	41	0	0
LIML 26/08 04:30	43	2.38	42	3	0

Table 11.11: 20 Targets, Medium-Short Mission

Chapter 12

Results of Metaheuristic Approach

The tables below show the results of the metaheuristic algorithms proposed. Only the Medium-Long missions are considered starting from 15 targets. The parameters of the Genetic Algorithm are the follows: $\eta_p = 200$, $\eta_t = 7000$ and $\eta_c = 50$. The column OBJ GA of the tables below reports the value of the solution of the Genethic Algorithm, then in the column GAP GA is reported the percentage gap between such solution and the optimal solution. Finally in the column CPU GA is reported the computing time of the Genetic Algorithm.

The parameters used for the Simulated Annealing are the follows:$T_0 = 10000$, $c_r = 0.999$ and $T_{min} = 0.000001$. The column OBJ SA of the tables reports the value of the solution of the Simulated Annealing Algorithm for the related instance. Then as in the previous case it is reported the percentage gap (GAP SA) and the computing time (CPU SA).
Note that the computing times of these algorithms is not so short as the computing times of the heuristic algorithms. Nevertheless they remain compatible with future real time applications.

Instance Name	15 TARGETS					
	OBJ GA	GAP GA	CPU GA	OBJ SA	GAP SA	CPU SA
LIML 11/11 05:30	104	2.97	6.115	108	6.93	11.529
LIML 22/08 04:30	105	2.94	8.112	103	0.9	11.295
LIML 25/08 04:30	105	6.06	6.63	110	11.11	11.342
LIML 25/08 12:30	105	0.00	8.205	108	2.85	11.809
LIML 26/08 04:30	102	3.03	6.927	109	10.1	11.372

Table 12.1: 15 Targets, Medium-Long Mission

Instance Name	20 TARGETS					
	OBJ GA	GAP GA	CPU GA	OBJ SA	GAP SA	CPU SA
LIML 11/11 05:30	173	1.76	12.152	181	6.4	15.491
LIML 22/08 04:30	176	6.66	16.411	181	9.6	14.571
LIML 25/08 04:30	174	4.81	17.566	180	8.4	14.462
LIML 25/08 12:30	176	2.92	17.02	183	7.01	14.414
LIML 26/08 04:30	177	7.92	16.677	177	7.92	14.618

Table 12.2: 20 Targets, Medium-Long Mission

Instance Name	25 TARGETS					
	OBJ GA	GAP GA	CPU GA	OBJ SA	GAP SA	CPU SA
LIML 11/11 05:30	192	3.78	11.918	199	7.56	17.69
LIML 22/08 04:30	191	3.8	29.672	200	8.69	18.315
LIML 25/08 04:30	191	4.37	13.634	193	5.46	17.831
LIML 25/08 12:30	195	5.4	31.356	199	7.56	19.017
LIML 26/08 04:30	190	3.8	42.635	199	8.74	18.518

Table 12.3: 25 Targets, Medium-Long Mission

Instance Name	30 TARGETS					
	OBJ GA	GAP GA	CPU GA	OBJ SA	GAP SA	CPU SA
LIML 11/11 05:30	205	8.46	40.997	205	8.46	22.292
LIML 22/08 04:30	219	16.48	28.58	198	5.31	21.669
LIML 25/08 04:30	196	5.94	30.357	213	15.13	21.622
LIML 25/08 12:30	209	8.85	30.217	209	8.85	22.386
LIML 26/08 04:30	206	9.57	44.21	214	13.82	22.293

Table 12.4: 30 Targets, Medium-Long Mission

Instance Name	35 TARGETS					
	OBJ GA	GAP GA	CPU GA	OBJ SA	GAP SA	CPU SA
LIML 11/11 05:30	218	11.22	61.48	219	11.73	27.393
LIML 22/08 04:30	219	12.88	48.797	228	17.52	25.412
LIML 25/08 04:30	222	12.12	80.605	226	14.14	25.053
LIML 25/08 12:30	225	15.97	34.554	213	9.79	25.693
LIML 26/08 04:30	210	6.5	96.627	214	8.62	26.941

Table 12.5: 35 Targets, Medium-Long Mission

Instance Name	40 TARGETS					
	OBJ GA	GAP GA	CPU GA	OBJ SA	GAP SA	CPU SA
LIML 11/11 05:30	232	11.00	42.62	240	14.83	29.702
LIML 22/08 04:30	241	16.99	84.6	243	17.96	29.437
LIML 25/08 04:30	244	18.44	53.087	227	10.19	28.298
LIML 25/08 12:30	231	13.23	95.582	238	16.66	28.563
LIML 26/08 04:30	225	9.22	66.332	239	16.01	29.391

Table 12.6: 40 Targets, Medium-Long Mission

Chapter 13

Conclusions

In this thesis we investigated the problem of the insertion of a UAS into a Controlled Air Space. The approach presented is a hybrid approach that merges the Operational Research with operative ATM techniques. First the problem was defined as the UAS Routing Problem and two different objectives were proposed: the minimization of the ATC impact over the mission and the minimization of the mission impact over the ATC. The problem was then modeled in two different ways: the first one using a TDTSP formulation and the second using a TSP with Penalty formulation (TSPP). Several exact and heuristic algorithms were then proposed to solve instances of different dimensions. An air traffic simulator was implemented and real air traffic data of Milano Linate Control Area were used to test the algorithms. The algorithms present different performances useful for different purposes. The branch and cut algorithm is able to solve instances up to 20 targets for medium-short missions and 15 target for medium-long missions. It is also able to use the same approach to model the different objectives. The algorithms based on the TSPP model are able to solve all types of mission instances presented. They present short computing times with respect to the Branch and Cut Algorithm but are still not compatible with real time applications. In fact a real time application can be obtained by the present off-line application considering a traffic update every minute. For this purpose, algorithms with a computing time smaller than one minute are required. The heuristic algorithms proposed have short computing times but present a quite large gap with respect to the optimal solution value. Metaheuristic algorithms present a smaller gap and quite small computing times. They probably represent the best approach for future real time applications. Finally, future developments will see the study of tabu search algorithms and the study of a model that considers only rerouting as an avoidance option. Such a model in fact makes it possible to use valid inequalities that are not useful for the models presented due to the presence of the holdings. Another interesting development could be the application of the algorithms presented for traffic problems in urban areas.

List of Figures

List of Tables

Bibliography

[1] K. Bilimoria, B. Sridhar, G. Chatterji, K. Sheth, and S. Grabbe : *FACET: Future ATM concepts evaluation tool* Proceedings of the 3rd USA/Europe Air Traffic Management RD Seminar, Naples, Italy, June 2000.

[2] J. Hoekstra, R. Ruigrok, R. van Gent, J. Visser, B. Gijsbers, M. Valenti, W. Heesbeen, B. Hilburn, J. Groeneweg, and F. Bussink : *Overview of NLR free flight project 1997- 1999* Tech. Report NLR-CR-2000-227, National Aerospace Laboratory (NLR), May 2000.

[3] Blom, H.A.P., Obbink, B.K. and Bakker, G.J., 2009. : *Safety Risk Simulation of an Airborne Self Separation Concept of Operation* 7th AIAA-ATIO Conference, September 18-20, 2007, Belfast, Northern Ireland.

[4] Farley, T., Kupfer, M., Erzberger, H. : *A Simulation Evaluation Under High Demand Including Merging Arrivals* 7th AIAA-ATIO Conference, September 18-20, 2007, Belfast, Northern Ireland.

[5] Erzberger, H. : *The Automated Airspace Concept* Proceedings of the 4th USA/Europe Air Traffic Management RD Seminar, Santa Fe, New Mexico, 2001.

[6] Mondoloni, S., Conway, S. : *Airborne Conflict Resolution for Flow-Restricted Transition Airspace* AIAA-2001-4054, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2003.

[7] Vivona, R. A., Karr, D. A., Roscoe, D. A. : *Pattern-Based Genetic Algorithm for Airborne Conflict Resolution* AIAA-2006-6061, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2005.

[8] Karr, D.A., Vivona, R. A., Roscoe, D.A., Consiglio, M., : *Experimental Performance of a Genetic Algorithm for Airborne Strategic Conflict Resolution* Proceedings of the 8th USA/Europe Air Traffic Management Research and Development Seminar (ATM2009).

[9] P.Toth,D.Vigo et Al. : *The Travelling Salesman Problem and its Variations* Kluwer Academic Publishers

[10] J.E.Beasley : *Adapting the savings algorithm for varying inter-customer travel times* Omega 9 (1981), pp. 658659.

[11] Malandrki and Daskin : *Time depend vehicle routing problem: Formulations, properties and heuristic algorithms* Transportation Science 26 (3) (1992), pp. 185200.

[12] Hill and Benton : *Modelling intra-city time-dependent travel speeds for vehicle scheduling problems* Journal of the Operational Research Society 43 (4) (1992), pp. 343351.

[13] Glover and VoSS : *A classification of formulations for the (time-dependent)traveling salesman problem* European Journal of Operational Research 83 (1995), pp. 6982.

[14] Malandraki and Dial : *A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem,* European Journal of Operational Research 90 (1996), pp. 4555.

[15] B. Fleischmann, M. Gietz and S. Gnutzmann : *Time-varying travel times in vehicle routing* Transportation Science 38 (2) (2004), pp. 160173.

[16] Y.B.Park : *A solution of the bicriteria vehicle scheduling problems with time and areadepend travel speeds* Computers and Industrial Engineering 38 (2000), pp. 173187.

[17] Y.B.Park : *The real-time time dependent vehicle routing problem* Transportation Research Part E 42 (2006), pp. 383408.

[18] Soler, Albiach, Martinez : *A way to optimally solve a time-dependent Vehicle Routing Problem with Time Windows* Operations Research Letters 37 (2009) 37-42

[19] Kuo, Wang, Chuang : *Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds*Computer and indistrial Engineering, (2009) Transportation Research Part E 42 (2006), pp. 383408.

[20] Stecco, Cordeau, Moretti : *A branch-and-cut algorithm for a production scheduling problem with sequence-dependent and time-dependent setup times* Computers and Operations Research 35 (2008) 2635 2655

[21] Bagassi S., Bombardi T., Francia D., Persiani C.A. : *, 3D Trajectory Optimization for UAS Insertion in Civil Non-Segregated Airspace* AIAA-2009-5840, AIAA Modeling and Simulation Technologies Conference, Chicago, Illinois, Aug. 10-13, 2009.

[22] E. Balas : *Disjunctive programming: properties of the convex hull of feasible points* Discrete Applied Mathematics, 1998, 1-44

[23] A. Balakrishnan, T.L. Magnanti and R.T. Wong : *A dual ascent procedure for large-scale uncapacitated network design* Operations Research , 1989, 716-740

[24] E. Balas and W. Pulleyblank : *The perfectly matchable subgraph polytope of a bipartite graph* Networks, 1983, 495516

[25] F. Barahona and R. Anbil : *The algorithm: Producing primal solutions with a subgradient method* Mathematical Programming, 2000, 385-399

[26] J.J. Bartholdi, J.B. Orlin and H. Ratliff : *Cyclic scheduling via integer programs with circular ones* Operations Research, 1980, 1074-1085

[27] J.F. Benders : *Partitioning procedures for solving mixed variables programming problems* Numerische Mathematik, 1962, 238-252

[28] O. Bilde and J. Krarup : *Sharp lower bounds and efficient algorithms for the simple plant location problem* Annals of Discrete Mathematics, 1977, 79-97

[29] G.H. Bradley, P.L. Hammer and L.A. Wolsey : *Coefficent reduction for inequalities in 0-1 variables* Mathematical Programming, 1974, 263-282

[30] M.Conforti, M. Di Summa, F. Eisenbrand and L. A. Wolsey : *Network formulations of mixed integer programs* Mathematics of Operations Research (to appear)

[31] M. Conforti and L. A. Wolsey : *Compact formulations as unions of polyhedra* Mathematical Programming, 2008, 277-289

[32] G.B. Dantzig and P. Wolfe : *Decomposition principle for linear programs* Operations Research, 1960, 101-111

[33] J. Desrosiers, F. Soumis and M. Desrochers : *Routing with time windows by column generation* Networks, 1984, 545-565

[34] B.P. Dzielinski and R.E. Gomory : *Optimal programming of lot sizes* Operations Research, 1965, 874-890

[35] D. Erlenkotter : *A dual-based procedure for uncapacitated facility location* Operations Research, 1978, 992-1009

[36] M.L. Fisher : *The Lagrangean relaxation method for solving integer programming problems* Management Science, 1981, 1-18

[37] D. Villeneuve, J. Desrosiers, M.E. Lubbecke and F. Soumis : *On compact formulations for integer programs solved by column generation* Annals of Operations Research, 2005, 375-388

[38] F. Furini, E. Malaguti, R. Medina, A. Persiani, P. Toth : *A Column Generation Heuristic for the Two-Dimensional Cutting Stock Problem with Multiple Stock Size* DEIS, University of Bologna, OR-10-9, 2010

[39] Lin, L. and Gen, M. Huang, YF and Gen, M and Kim, KH, : *A random key-based genetic algorithm for AGV dispatching problem in FMS*, Proceedings of the 4th International Conference on Intelligent Logistics Systems, 2008, 318-330,

[40] Lin, L. and Shinn, S. W. and Gen, M. and Hwang, H. : *Network model and effective evolutionary approach for AGV dispatching in manufacturing system*, of Intelligent Manufacturing, 2006, 17, 4, 465-477,

[41] Mohring, R. H. and Kohler, E. and Gawrilow, E. and and Stenzel, B. : *Conflict-free Real-time AGV Routing*, Operations Research Proceedings 2004, 2005, 2004, series Operations Research Proceedings, 18-24, publisher Springer Berlin Heidelberg,

[42] Nishi, T. and Morinaka, S and Konishi, M. : *A distributed routing method for AGVs under motion delay disturbance*, Robotics and Computer-Integrated Manufacturing, 2007, 23, 5, 517-532,

[43] R. Stahlbock and S. Voss : *Operations research at container terminals: a literature update*, OR Spectrum, 2008, 30, 1-52,

[44] D. Steenken and S. Voss and R. Stahlbock : *Container terminal operation and operations research: a classiffication and literature review*, OR Spectrum, 2004, 26, 3, 3-49,

[45] I. Vis : *Survey of research in the design and control of automated guided vehicle systems*, European of Operational Research, 2006, 170, 677-709,

[46] H. Wren : *Automation - a case for the future*, Port Technology International, 2009, 42, 41-42,

[47] Y. Saanen : *Automated Container Handling*, Freight International, 2008, URL http://www.freight-int.com/categories/automated-container-handling/automated-container-handling.asp,

[48] D. Steenken and S. Voss and R. Stahlbock : *Container terminal operation and operations research: a classiffication and literature review*, OR Spectrum, 2004, 26, 3, 3-49,

[49] Bae, J. and Kim, K. : *A pooled dispatching strategy for automated guided vehicles in port container terminals*, International of Management Science, 2000, 6, 47-67,

[50] Koo, P. and Lee, W. and Koh, S. : *Vehicle dispatching for for container transportation in seaport container terminals*, Proceedings of the 7th international conference of computers and industrial engineering, 2004, address Jeju, Korea,

[51] Europe Container Terminals : *Euromax: a new standard in container handling*, Port Technology International, 2009, 41, 56-61,

[52] Vis, I. F. A. : *Survey of research in the design and control of automated guided vehicle systems*, European of Operational Research, 2006, 170, 677-709,

[53] Stahlbock, R. and Voss, S. : *Operations research at container terminals: a literature update*, OR Spectrum, 2008, 30, 1, 1-52,

[54] Lin, L. and Gen, M. Huang, YF and Gen, M and Kim, KH, : *A random key-based genetic algorithm for AGV dispatching problem in FMS*, Proceedings of the 4th International Conference on Intelligent Logistics Systems, 2008, 318-330,

[55] Calinescu, Gruia and Chakrabarti, Amit and Karloff, Howard J. and Rabani, Yuval, : *Improved Approximation Algorithms for Resource Allocation*, booktitle Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization, 2002, 401–414,

[56] M. Bartlett, A. M. Frisch, Y. Hamadi, I. Miguel, S. A. Tarim and C. Unsworth, : *The Temporal Knapsack Problem and its Solution*, Proceedings of the International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 2005, 34–48,

[57] Arkin, E. M. and Silverberg, E. B., : *Scheduling jobs with fixed start and end times*, Discrete Applied Mathematics, 18, 1, 1987, 1–8,

[58] Hall, N. G. and M. J. Magazine : *Maximizing the value of a space mission* EJOR, "78", "224-241", "1994"

[59] Chen, B. and Hassin, R. and M. Tzur : *Allocation of Bandwidth and Storage* "IIE Transactions", "24", "501-507", "2002"

[60] Vanderbeck, F. and L.A. Wolsey : *Reformulation and Decomposition of Integer Programs* 2010, "431–502"

[61] Fekete, S. and Schepers, J. : *New classes of fast lower bounds for bin packing problems* "Mathematical Programming", "91", "11-31", "2001"

[62] Gabrel, V. and Minoux, M. : *A scheme for exact separation of extended cover inequalities and application to multidimensional knapsack problems* "Operations Research Letters", "30", "252-264", "2002"