The Time Dependent Traveling Salesman Planning Problem in Controlled Airspace

Fabio Furini

LAMSADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris, France
fabio.furini@dauphine.fr

Carlo Alfredo Persiani

ENAV S.p.A, Italian Agency for Air Navigation Services Via Salaria 716 Roma, Italy
carlo.persiani@enav.it

Paolo Toth

DEI, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
paolo.toth@unibo.it

Abstract

The integration of Unmanned Aircraft Systems (UAS) into civil airspace is one of the most challenging problems for the automation of the Controlled Airspace, and the optimization of the UAS route is a key step for this process. In this paper, we optimize the planning phase of a UAS mission that consists of departing from an airport, flying over a set of mission way points and coming back to the initial airport. We assume that during the mission a set of piloted aircraft flies in the same airspace and thus the cost of the UAS route depends on the air traffic and on the avoidance manoeuvre used to prevent possible conflicts. Two Air Traffic Management techniques, i.e., rerouting and holding, are modelled in order to maintain a minimum separation between the UAS and the piloted aircraft. Heuristic algorithms are proposed for the solution of the considered problem, called the Time Dependent Traveling Salesman Planning Problem in Controlled Airspace (TDTSPPCA). A mathematical formulation based on a particular version of the Time Dependent Traveling Salesman Problem (TDTSP), which allows holdings at mission way points, is proposed for solving the TDTSPPCA, and applied to the UAS route planning phase to minimize the total operational cost. Another formulation, based on a Travelling Salesman Problem variant that uses specific penalties to model the holding times, is proposed and compared with the first one. Finally, computational experiments on real-world air traffic data from Milano Linate Terminal Manoeuvring are reported to evaluate the performance of the proposed models and of the heuristic algorithms.

Keywords: Integer Programming, Traveling Salesman Problem, Time Dependence, Unmanned Aerial Systems, Air Traffic Management

1. Introduction

The Traveling Salesman Problem (TSP) is one of the most investigated problems in Transportation Science and many applications and variants have been studied during the last few decades. The temporal extension of the TSP, called *Time Dependent Traveling Salesman Problem* (TDTSP), has been addressed to deal with time dependent problems as traffic operations management in congested urban areas. The basic idea is that the cost of a path between two locations depends on the time of the day: during peak hours, the traversing time could significantly change with respect to different hours of the day. The version of TDTSP arising in *Controlled Airspace* presents three main differences with respect to the version of TDTSP concerning applications in urban areas. First, the sampling interval must be compatible with the characteristics of the air traffic, i.e., the temporal discretization must consider minutes instead of hours. Secondly, the international regulation of the airspace defines a minimum *separation* that must be applied between aircraft to avoid conflicts. The third difference concerns the fact that, in Controlled Airspace, paths between locations are forbidden during specific time periods due to the presence of other aircraft. These important differences lead to different mathematical models and solutions techniques.

There are several applications of the TDTSP in Controlled Airspace: one of the most important is the route planning for automated aircraft. The use of *Unmanned Aircraft Systems* (UAS) in military missions has proved their capability and versatility in different applications. The so called "D3 paradigm" (Dull, Dirty, Dangerous) emphasizes the ability of UAS to operate autonomously in complex scenarios, reducing both the necessity of human intervention and the correlated risks. There are many civil applications that can also be performed by 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 operational and personnel costs. As a case study in our computational tests, we use the planning phase of a so called "Intelligence, Surveillance, and Reconnaissance" (ISR) mission for UAS. We choose this application since the UAS operations are at a higher level of automation than those associated with the piloted aircraft. However, the proposed methodology could be applied also to the piloted traffic when the level of automation in the traffic control increases.

The *Time Dependent Traveling Salesman Planning Problem in Controlled Airspace* (TDTSPPCA) considered in this paper consists of finding the best flying sequence of an aircraft over a set of fixed mission way points at a given altitude, aiming at minimizing the operational cost and, at the same time, ensuring a minimum separation with respect to the planned traffic. Two avoidance techniques are considered: the possibility of rerouting with respect to the optimal TSP sequence, and the possibility of holding the aircraft over a mission way point. These avoidance techniques are particularly effective in the planning phase as the one considered in this work.

Paper contributions.. We propose two new Mixed Integer Linear Programming (MILP) formulations of the Time Dependent Traveling Salesman Planning Problem in Controlled Airspace: the *Time-Based* model and the *Penalty-Based* model. Both formulations are capable of successfully dealing with real-world data, and specific algorithms are developed to speed up the solution time. Our methodology is based on a hybrid approach that merges *Air Traffic Management* (ATM) and *Operations Research* techniques. This approach shows that the problem of finding the best flying sequence to visit a set of mission way points in terms of total cost and the problem of maintaining a minimum separation between the aircraft can be combined using a temporal extension of the classical Traveling Salesman Problem (TSP). In other words, our goal is to solve the UAS ATM planning phase by proposing an approach able to find optimized conflict-free routes. The output of the planning phase can then be the input for further tactical phases as the trajectory optimization or the real-time reoptimization of the routes (see Section 8 for further details on this issue).
The paper is organized as follows. In Section 1.1 we describe the mission scenario, its constraints and the assumptions used. In Section 2 we describe the literature on related topics. In Section 3 we model the mission environment and the separation constraints. In Section 3.1 we formally address the Traveling Salesman Planning Problem in Controlled Airspace and, in Section 4 we present new heuristic algorithms. In Section 5, we introduce the *Time-Based* model and, in Section 6, the *Penalty-Based* model. Finally in Section 7, we report extensive computational experiments testing the proposed formulations and the heuristic algorithms on real air traffic data from Milano Linate (LIML) Terminal Maneuvering Area (TMA). In Section 8, we discuss possible extensions of the models in order to take into consideration additional real-world features of the problem and the conclusions are discussed in Section 9.

1.1. Aeronautical background

The Controlled Airspace is the part of the airspace where the air traffic control service is provided, i.e., the air traffic controllers ensure a minimum spatial and/or temporal separation between aircraft. For this aim, they control for each aircraft in their area of responsibility:

- route,
- altitude,

- speed,
- rate of climb/descent,
- heading.

A pilot plans his flight from an airport to another one requesting a route and a cruise level to the air traffic control. The air traffic control approves the request or re-plans it. The *route* of an aircraft is defined by a sequence of navigation points and altitudes. An *airway* is a predefined succession of navigation points. An aircraft can follow an airway or can proceed directly between two navigation points belonging to different airways if cleared by the air traffic control. The goal of the air traffic control is to ensure a safe and ordered flow of the air traffic in the Controlled Airspace. The controllers provide the legal minimum separation between aircraft using different techniques, the most used ones are:

- the vectoring,
- the speed control,
- the use of holdings,
- the rerouting,
- the use of different levels.

The use of a particular technique depends on several factors as:

- the airspace layout,
- the aircraft performances,
- the human factors,
- the pilot training.

The choices of the separation technique, of the routes and of the altitudes are strictly connected and affect the controlled air traffic and its evolution.

This is the ideal general context of a routing problem in Controlled Airspace. Moreover, there are many stochastic aspects that air traffic controllers have to face during the traffic management, e.g. the real traffic may differ from the planned one and the weather conditions affect the routes and the avoidances. Since considering these aspects all together goes behind the scope of the paper, we put ourself in the context of planning a conflict free route considering the following realistic assumptions.

1.2. Modeling assumptions

We investigate what is known in the literature as the *static case* of the problem.

- We assume that the air traffic the UAS may encounter is known in advance. We take into account the real air traffic of a simulation day in a specific Terminal Area (i.e., a sector of Controlled Airspace around big airports) using historical data regarding the aircraft routes. We simulate such a traffic plan for the UAS mission, i.e., we optimize the UAS planned route, as it had flown in such an environment.

- We consider the UAS as traffic with lower priority; this is a realistic assumption since some UAS missions in Controlled Airspace will require a flexible management of the automated aircraft.
  We assume that the conflict detection is performed in the planning phase using foreseen air traffic data and not from a real-time perspective. Further developments, especially regarding the dynamic case, will require the assumption that the UAS can detect other aircraft.

- The real air traffic is composed by the VFR (Visual Flight Rules) traffic, i.e., the aircraft that navigate by sight, and by the IFR (Instrumental Flight Rules) traffic, i.e., the aircraft that navigate using instruments. We consider only the IFR traffic.

- We model only some avoidance techniques, i.e., the rerouting and the use of holdings, obtaining a conflict free optimized route from a planning perspective. Such a route can be post-optimized in order to define the trajectory in a given scenario.

- Finally, we assume that once an holding has been planned, the UAS can complete it without additional separation conflicts (e.g., in case of conflict the altitude can be changed accordingly).

2. Related works

Given a list of points and the distances between each pair of them, the TSP consists of finding the shortest possible Hamiltonian circuit, i.e. the minimum distance route that visits each point exactly once. The TSP has been widely studied in the past years and many exact and heuristic algorithms have been developed to solve different routing problems. For a complete survey on the TSP see [9]. However, studies on TSP with time-dependent travel costs (TDTSP) are fairly rare in the literature. Two main kinds of TDTSP have been proposed in the literature. The first one is characterized by arc costs that depend on the position of each arc in the Hamiltonian circuit, the second one by arc costs that depend on the time at which each arc is visited. As far as the first kind of TDTSP is concerned, we refer the interested reader to the seminal work of [17].

The second kind of TDTSP arose to deal with traffic jams in congested urban areas: during peak periods the cost of a path between two clients becomes higher than in normal conditions. We follow the second stream of research as this model is also useful to interpret air traffic using a convenient time horizon compatible with aircraft operations. In fact, the cost of a UAS route between two mission way points depends on the air traffic met by the UAS, starting a path at a specific time. Considering a time interval of one minute and a mission duration of some hours many exact algorithms proposed in the literature are not able to provide solutions in an acceptable computing time. [3] proposed a heuristic algorithm for the TDTSP, where two periods of the day present different travel costs, adapting the classical savings algorithm developed by Clark and Write for the Vehicle Routing Problem. [11] proposed a Mixed Integer Linear Programming (MILP) formulation of the TDTSP and of the related time dependent vehicle routing problem TDVRP; they handled the travel cost function as a step function and presented several heuristic algorithms to solve these problems.

In [8] a classification of the TDTSP is presented, considering old and new formulations obtained from a quadratic assignment model for the TDTSP. [12] used a dynamic programming heuristic algorithm to solve the TDTSP with a given starting time from the depot. [15] considered the bi-criteria TDVRP with time and area dependent travel speed. The minimization of the total vehicle operation time and the minimization of the total weighted tardiness were the conflicting objectives studied, and a MILP 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 called the bi-criteria-saving algorithm to solve this problem. [7] 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 using a smoothed travel time function for calculating the 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 reported on instances based on traffic information obtained from the city of Berlin. [4] 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. [2] considered a version of TDTSP with time windows proposing an exact solution through a graph transformation. Such a problem is first transformed into an Asymmetric Generalized TSP and then into a Graphical Asymmetric TSP in order to apply exact known algorithms for the Mixed General Routing Problem. [19] proposed a branch and cut algorithm designed for a TDTSP used to model a production scheduling problem. In fact, since the set-up time between two jobs is a function of the completion time of the first job, the considered scheduling problem can be reformulated using a TDTSP approach. [19] also introduced some families of valid inequalities used to strengthen the Linear Programming Relaxation of the proposed Integer Linear Programming (ILP) formulation. [18] considered the TDVRP with time windows; they showed how to transform this problem into an Asymmetric Capacitated Vehicle Routing Problem in order to solve it with known exact or heuristic algorithms. [10] proposed a Tabu Search algorithm to solve the TDVRP and the related goods assignment problem: a real-world case of a warehousing company was used to illustrate the studied method. In [1], a study of the polytope associated with a particular TDTSP formulation is presented and a branch-cut-and-price approach is applied to instances from the TSPLIB. Finally, a routing problem for UAS in controlled airspace has been presented in [16], where the authors propose a Genetic Algorithm designed to manage a UAS mission in civil airspace considering all the avoidance techniques previously mentioned.

3. Airspace and traffic separation modeling

The mission environment is modeled considering its key features: the air space layout, the aircraft and their routes, the UAS and the ATC avoidance techniques. The time is discretized, i.e. we consider the time as a succession of time steps separated by an interval chosen accordingly to the precision necessities. If t_{max} is an upper bound on the mission duration obtained by the UAS endurance and Δt is the duration of a time step, the set $T = \{0, 1, 2, \ldots, n_{ts}\}$ is the set of time steps where $n_{ts} = t_{max}/\Delta t$ represents the number of time steps used.

We consider a set A of n aircraft flying during the period under investigation and an airspace defined by the set N_A of q navigation points used by the aircraft. Each navigation point $p \in N_A$ is characterized by a geographical position. Each aircraft $a \in A$ is characterized by a route $r_a \subseteq N_A$ defined by a succession of navigation points. The altitude and the time of the aircraft a passing through the navigation point p are called respectively l_p^a and t_p^a, moreover we define the set V_A as the set of all the pairs of navigation points and the set $R_A = \cup_{a \in A} r_a$ as the set of all aircraft routes. Finally we introduce the complete graph $G_A = \{N_A, V_A\}$ as the "Airspace Graph", i.e. the graph that defines the aircraft routes. This graph is directed and represents the airspace environment. Each aircraft, which flies in the modeled airspace during the mission, is moving over the Aircraft Graph G_A.

The surveillance mission consists of departing from an airport (mission way point 0), taking pictures over a set $M \setminus 0$ of $(m - 1)$ mission way points and coming back to the airport. Each mission way point $i \in M$ is characterized by a geographical position and an altitude. We assume that between each pair of mission way points a direct route is available; let V_U be the set of all pairs of mission way points and we denote a specific pair with (i, j). We introduce the complete graph $G_U = \{M, V_U\}$ as the "UAS Graph" ,i.e. the graph that defines the UAS routes.

A holding pattern is available over each mission way point: in other words the UAS can overfly and, if necessary, can perform one or more orbits over a mission way point. We assume that the cost of a route can be approximated using the flight time, once the distances between the mission way points and the UAS mission parameters are known. These parameters (called P_{uas}) depend on the UAS performance and consist of the speed v_U (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 routes in segregated air space. Using such parameters, we define the flight time f_{ij} necessary to reach the mission way point j from the mission way point i. Due to the different performance during the climb and descent phases, we assume that the time from i to j is different from the time from j to i. Finally, since we assumed the UAS as the airspace user with lower priority, if a conflict between an aircraft and the UAS exists, the UAS has to wait or to change its route and not vice versa (see [13] for further details on this topic). According to the air space classification, a minimum separation must exist between each pair of aircraft (UAS included); two ATC separation techniques are modeled and applied to the UAS planned route: the rerouting or the holding over a mission way point.

Let S be the set of all possible conflicts: a conflict $s \in S$ between the UAS sub-route $(i, j) \in V_U$, and an aircraft route r_a occurs if at least one point exists on the UAS sub-route 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, for the conflict s we define the Conflict Point p_c^s as the point closest to r_a. The time step in which the aircraft is overflying the closest point to p_c^s on its route is defined as the Conflict Time Step t_c^s. The avoidance of a conflict is modeled by forbidding the departure of the UAS from the mission way point i to the mission way point j in the time steps involved in such conflict. For a given pair of UAS sub-route (i, j), and aircraft route r_a, a time step is involved in a conflict s if a UAS departure in such time step would determine an insufficient separation. Let us indicate with X_{ij}^s ($X_{ij}^s \subseteq T$) the conflict interval as the interval defined by the time steps involved in the conflict s over the sub-route (i, j).

The identification of the time steps involved in a conflict for a UAS sub-route can be performed using simple geometric considerations: we use the conflict detection geometric algorithm presented in [16] where it is also reported a complete analysis of all possible conflict cases. We report in Figure 1 the general case of conflict. The Conflict Zone consists of a sub-separation area defined by the UAS and the aircraft route segments that are not separated from each other. Each conflict has an associated Conflict Zone: the

Conflict Zone can not be occupied by the UAS and an aircraft at the same time. Once checked that the vertical separation of 1000ft does not exist between all points of the conflicting routes, their projection on the horizontal plain is used to identify the Conflict Zone. The segments (p_i, p_j) and (p_a, p_b) represent the projection on the plane of the UAS sub-route (i, j) and of the aircraft route (a, b), respectively. t_a and t_b are the time steps in which the aircraft is overflying p_a and p_b respectively. The point p_{1cu} is defined as the projection of the first point over the UAS sub-route between p_i and p_c^s that presents a distance from the aircraft route of 1000 ft in the vertical direction or 5 Nautical Miles (NM) in the horizontal direction, whichever happens first, moving from p_i to p_c^s. Similarly, the point p_{2cu} is the projection of the first point over the UAS path between p_j and p_c that presents a distance from the aircraft route of 1000 ft in the vertical direction or 5NM in the horizontal direction, whichever happens first, moving from p_c^s to p_j. Analogously the points p_{1ca} and p_{2ca} are defined over the route of aircraft a; the time steps in which aircraft a overflies such points are, respectively, t_{1ca} and t_{2ca}. The Conflict Zone is defined by the points p_{1cu}, p_{1ca}, p_{2cu} and p_{2ca} and it is comprised by two ”sub-separation stretch”: the first one is identified over the UAS route between p_{1cu} and p_{2cu} and the second one over the aircraft route between p_{1ca} and p_{2ca}. If the UAS is moving from p_i to p_j and the aircraft from p_a to p_b, the Conflict Interval over p_i starts after the last departing time step available before the conflict. That is the departing time step from p_i such that the UAS can arrive in p_{2cu} when the aircraft a is in the point p_{1ca}. The holding starts after the time step t'_{p_i}:

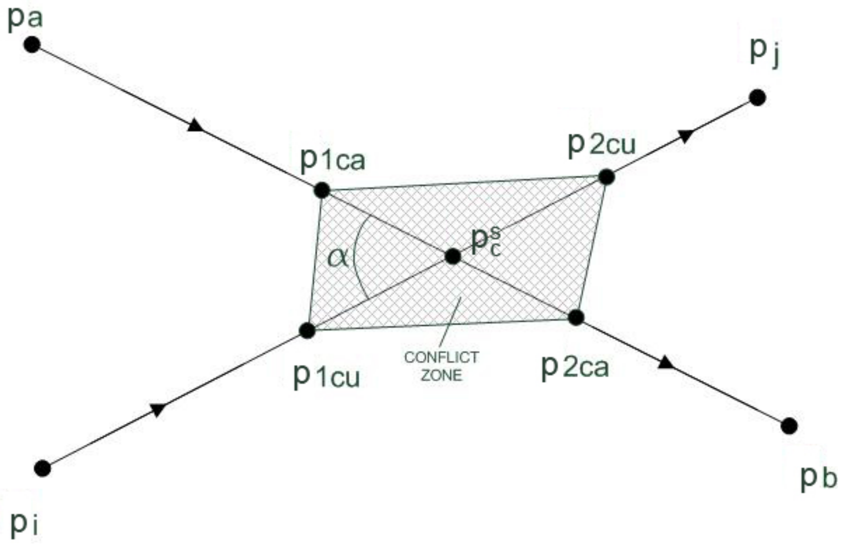


Figure 1: Conflict Zone

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

Where $\Delta t_{p_i,p_{2cu}}$ is the time step interval necessary for the UAS to fly from p_i to p_{2cu} maintaining P_{uas}, and $\Delta t_{p_a,p_{1ca}}$ is the time step interval necessary to the aircraft to fly from p_a to p_{1ca}. In the same way, the Conflict Interval over p_i finishes before the first re-departing time step available after the conflict. That is the departing time step from p_i such that the UAS can arrive in p_{1cu} when the aircraft a is in the point p_{2ca}. The holding finishes before the time step t''_{p_i}:

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

Where $\Delta t_{p_a,p_{2ca}}$ is the time step interval necessary for the aircraft to fly from p_a to p_{2ca}, and $\Delta t_{p_i,p_{1cu}}$ is the time step interval necessary for the UAS to fly from p_i to p_{1cu}. The dimension of the Conflict Zone depends on the angle α (that defines if the routes are diverging or converging) and on the altitudes of the UAS and the aircraft on the mission way points. If $\alpha = 90$ and the altitudes on the points are the same, the Conflict Zone consists of a circle. In case $\alpha = 180$ (opposite routes) or $\alpha = 360$ (same routes) and the altitudes on the points are the same, the entire sub-route becomes a Conflict Zone during the aircraft passage. Using the notation reported above it is possible to write the conflict interval X^s_{ij} defined by the time steps involved in the conflict s over the UAS sub-route (i, j):

$$X^s_{ij} =]t'_{p_i}, t''_{p_i}[\tag{3}$$

Considering all the conflicts over the UAS sub-route (i, j), we indicate with X_{ij} the set of the all time steps involved, i.e. the time steps in which a departure from mission way point i to the mission way point j produces a conflict with an aircraft $a \in A$:

$$X_{ij} = \cup_{s \in S} X^s_{ij} \tag{4}$$

Considering that the air traffic and the related avoidance techniques used by the UAS in a sub-route (i, j) depend on the time at which the UAS starts to fly that sub-route, the problem combines two decisions: the choice of the best order of visiting a set of mission way points and the choice of the avoidance techniques that have to be used. These two dimensions can be combined in a Time Dependent Asymmetric Travelling Salesman Problem where the cost of a path between two mission way points varies in time. Starting a sub-route at a specific time could produce a conflict and thus require appropriate management, i.e. rerouting or holding.

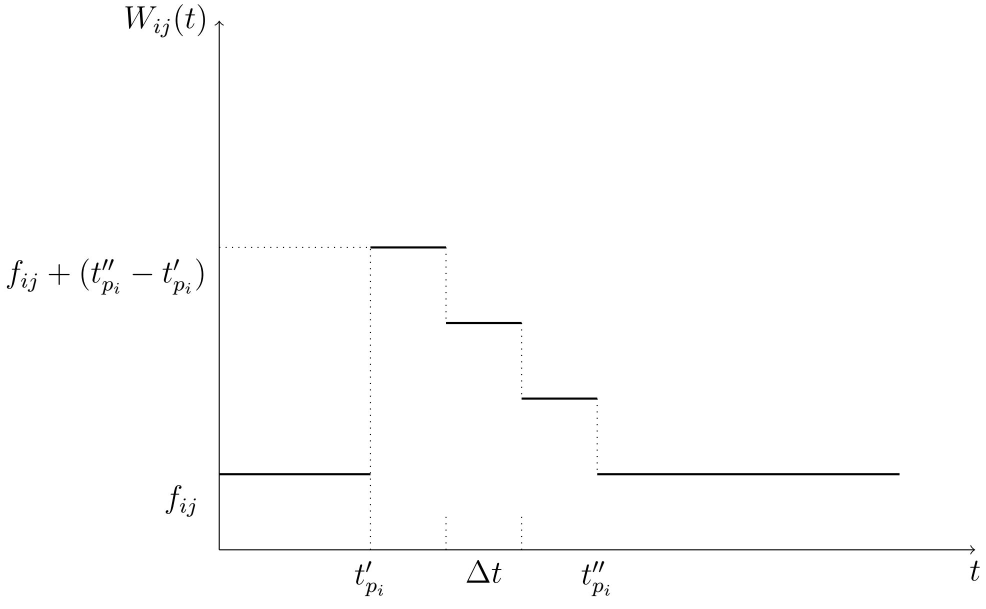


Figure 2: The time step function

Figure 3 describes the traversing time (called $W_{i,j}(t)$ in the figure) of the sub-route (i, j) as a function of the departing time step t. $W_{i,j}(t)$ can be calculated considering that at each time step t the UAS has two options to proceed from i to j: if no conflict occurs ($t \notin X_{i,j}$) the UAS can proceed directly from i to j and thus the traversing time corresponds to the flight time $f_{i,j}$, while in the event of conflict ($t \in X_{i,j}$) it consists of the sum of two components: the remaining holding time from t until the first no conflicting time step and the flight time $f_{i,j}$.

3.1. Problem definition

Using the previous notation, we define the Time Dependent Traveling Salesman Planning Problem in Controlled Airspace (TDTSPPCA) as follows: let $G = \{G_u, A, G_a, S, T\}$ be the mission conflict system, where at each mission way point $i \in M$, a holding pattern is permitted. Let β_i be the cost of a time step in the holding pattern associated with i, and γ the cost of a time step during the flight phase. Moreover, for a given set of flight parameters P_{uas}, let $f_{i,j}$ be the flight time from point i to point j, and k_{min} the minimum duration of a holding and k_{max} the maximum duration of a holding (with $k_{max} \geq k_{min}$). For each pair (i, j) of mission way points, let $X_{i,j}$ be the set of time steps in which a departure from point i to point j produces a conflict with an aircraft $a \in A$. The goal consists of finding the route starting and coming back to the same prefixed mission way point (airport) that minimizes the mission cost, defined by the weighted sum of the flying time and the holding time, maintaining the minimum separation from the piloted traffic and visiting all the mission way points.

The value of the optimal solution of the ATSP associated with the graph G_U represents a lower bound on the value of the optimal solution of the TDTSPPCA defined by the formulation (5)-(17) described in Section 5.1, while the value of the solution defined by the heuristic algorithm described in Section 4, called heuristic based on the ATSP route (HTSP), obtained by following the ATSP route and performing the appropriate holdings represents instead an upper bound on the value of the optimal solution of the TDTSPPCA. The TDTSPPCA is an asymmetric problem for two reasons: (i) the cost (flight time) of the sub-route (i, j) may be different from the cost of the sub-route (j, i) because the related mission way points may be at different altitudes; (ii) due to the presence of the air traffic, if the cost of the sub-routes is the same in both directions (symmetric TSP) two tours that present the same sub-route but in opposite directions could have different total times. Moreover, typical heuristic approaches for the TSP as the $k - opt$ exchange or insertion heuristics cannot easily be extended to the TDTSPPCA (see [11]). Another important characteristic of the problem follows from the properties of the considered objective function. If the holding costs $\beta_i, (i \in M)$ are all equal since it is not important where the UAS waits, once a sequence of mission way points is fixed, the best way to visit them consists of waiting as less as possible over each of them. In other words, the route cost depends only on the sequence of the mission way points. This obviously does not hold when the holding costs are different.

4. Heuristic Algorithms

The nature of the considered problem is such that the air traffic situation changes at least every minute; this is a significant difference with respect to the TDTSP developed for taking into account the traffic in big urban areas where the update of the traffic can be done every hour or couple of hours. This implies that in a reference period of one day, the time steps in a city traffic problem are in a small number with respect to those corresponding to a 4 hours endurance UAS mission with a time step of 1 minute. Moreover, since t_{max} is an upper bound defined using the UAS endurance, it may be quite long with respect to the real mission duration. An appropriate tailoring of the set T becomes critical to solve the TDTSPPCA and thus heuristic approaches can be used to properly reduce the size of the set T.

Two heuristic algorithms have been designed to this aim: a Nearest Neighbour algorithm (NN) and an ATSP route based algorithm (HTSP), plus a local search algorithm designed to further improve the solutions obtained.

The *ATSP route based algorithm* (HTSP) starts from the optimal TSP on the graph G_u and the resulting tour is copied in the route vector. The holding vector is obtained by following the route defined by the route vector and performing the appropriate holdings: if a departing time step from a mission way point belongs to a Conflict Interval the holding corresponds to the wait until the first available departing time step (otherwise the holding is set to 0). In the *Nearest Neighbour algorithm* (NN), the solution vector is initially void and the algorithm starts initializing the route vector with the airport. The solution is then iteratively built, searching for the nearest mission way point in terms of flight time and holding cost (total cost). Once a mission way point is inserted into the route vector, the related holding to reach it is inserted in the holding vector.

The neighbourhood used for the *Local Search algorithm* (LS), is the set of solutions obtained by performing all the exchanges between each pair of mission way points. The Local Search algorithm is initialized through the route provided by one of the two previously mentioned heuristic algorithms. If a cheaper route can be found in the neighbourhood the current solution is updated and the algorithm reiterates, otherwise the algorithm terminates with the current best solution. Since the described algorithms require a very short computing time, a good upper bound t_{ub} on the mission duration may be fastly computed as the smallest between their solutions. Using this upper bound the number of time steps n_{ts} is possibly reduced. Detailed results of the heuristic algorithms are reported in Section (7).

5. Time-Based Model

In this section we present a time extension of a classical ATSP formulation, a Variable Fixing Procedure to reduce the dimension of the model and a Branch and Cut Algorithm.

5.1. Mathematical Formulation and main features

Two types of binary variables are used to render the time dimension of the problem (variables x_{ij}^t) and to model the holding over the mission way points (variables y_i^t). Variable x_{ij}^t ($(i,j) \in V_U, t \in T$) has value 1 if the UAS starts to fly from mission way point i to mission way point j at time step t and 0 otherwise. Variable y_i^t ($i \in M, t \in T$) takes value 1 if the UAS arrives at the mission way point i at time step t and 0 otherwise. By using these variables, for each mission way point $i \in M$, the arrival and departure times are given, respectively, by $\sum_{t \in T} y_i^t$ and $\sum_{t \in T} \sum_{j \in M} x_{ij}^t$. The Time-Based model of the TDTSPPCA reads as follows:

$$\min \quad \gamma \sum_{t \in T} \sum_{(i,j) \in V_U} f_{ij} x_{ij}^t + \sum_{i \in M} \beta_i \Big(\sum_{t \in T} \sum_{j \in M} t x_{ij}^t - \sum_{t \in T} t y_i^t \Big) \tag{5}$$

$$\sum_{t \in T} \sum_{j \in M} x_{ji}^t = 1 \qquad i \in M \tag{6}$$

$$\sum_{t \in T} \sum_{j \in M} x_{ij}^t = 1 \qquad i \in M \tag{7}$$

$$\sum_{t \in T \setminus \{0\}} y_i^t = 1 \qquad i \in M \setminus \{0\} \tag{8}$$

$$\sum_{j \in M, t - f_{ji} \geq 0} x_{ji}^{t-f_{ji}} \leq y_i^t \qquad i \in M \setminus \{0\}, t \in T \tag{9}$$

$$\sum_{j \in M} x_{ij}^t \leq \sum_{t^*=0}^{t} y_i^{t^*} \qquad i \in M, t \in T \tag{10}$$

$$\sum_{t \in T} \sum_{j \in M} t x_{ij}^t - \sum_{t \in T} t y_i^t \geq k_{min} \qquad \forall i \in M \tag{11}$$

$$\sum_{t \in T} \sum_{j \in M} t x_{ij}^t - \sum_{t \in T} t y_i^t \leq k_{max} \qquad \forall i \in M \tag{12}$$

$$x_{ij}^t = 0 \qquad t : t + f_{ij} \geq t_{max}, (i,j) \in V_U \tag{13}$$

$$x_{ij}^t = 0 \qquad t \in X_{i,j}, (i,j) \in V_U \tag{14}$$

$$y_0^0 = 1 \tag{15}$$

$$x_{ij}^t \in \{0,1\} \qquad (i,j) \in V_U, t \in T \tag{16}$$

$$y_i^t \in \{0,1\} \qquad i \in M, t \in T, \tag{17}$$

where γ and β_i ($i \in M$) are given coefficients. The objective function is composed by two terms: the first term represents the total flight time between the mission way points; the second term describes the holding duration over each mission way point as the difference between the departure and the arrival time steps. The coefficients γ 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, the coefficients β_i can be increased appropriately. If γ = β_i =1 ($i \in M$) the objective function minimizes the total mission duration in terms of time steps and thus the impact of the ATC over the mission. To obtain the mission duration in time units it is necessary to consider the duration of a generic time step Δt. Constraints (6) and (7) are the assignment constraints, imposing that each mission way point i has one entering arc and one leaving arc, respectively. Constraints (8) ensure that each mission way point i has one arrival time. Constraints (9) and (10) joined with the initial condition (15), act as Subtour Elimination Constraints and ensure the temporal coherence of the tour. In addition they force the departure from the airport at time step $t = 0$. Constraints (9) ensure that if the UAS arrives at the mission way point i at time step t, it must have departed from the previous mission way point at an appropriate time step compatible with the flight time. Constraints (10) impose that if the UAS leaves the mission way point i at time step t, the arrival at i must occur in a previous or in the same time step. Both render the time coherence on the mission way points. These groups of constraints make it possible to build the Hamiltonian circuit over the mission way points as in the ATSP. Constraints (14) impose no departure, according to the minimum separation constraint, in the forbidden time steps while constraints (13) forbid departures after the last time step. Constraints (11) and (12) impose a lower bound (κ_{min}) and an upper bound (κ_{max}) on the duration of the holding over a mission way point i, ensuring that the holding is consistent with the UAS performances. As an alternative, it is possible to set κ_{min} equal to 0, solve the problem and replace the corresponding holdings by using different avoidance manoeuvres such as speed control. For example, a holding smaller than a lower bound value can be avoided by reducing the UAS speed in the related sub-path.

Even though they are not necessary for the elimination of the sub-tours, the following inequalities can be added to formulation (5)-(17) in order to strengthen its Linear Programming (LP) relaxation:

$$\sum_{t \in T} \sum_{i \in Sub, j \in M \setminus Sub} x_{ij}^t \geq 1 \qquad 2 \leq |Sub| \leq |M| - 2, Sub \subset M \tag{18}$$

Such inequalities are a time extension of the well known Subtour Elimination Constraints (SEC) of the Dantzig-Fulkerson-Johnson model ([5]) for the Asymmetric TSP (ATSP). The Padberg-Rinaldi ([14]) separation procedure can be efficiently used to separate in polynomial time such inequalities. Thus a Branch-and-Cut algorithm was developed based on formulation (5)-(18). Before starting the algorithm, a preprocessing phase is performed. This phase consists of two steps: computation of an upper bound on the mission duration and a Variable Fixing procedure.

5.2. Variable Fixing procedure

The Variable Fixing procedure is based on the concept of ”infeasible departure” time steps. An ”Infeasible Departure ” time step is a time step at which a departure of the UAS cannot take place; it is due to three reasons. The first one depends on the first time step at which it is possible to reach a given mission way point i. If t_i^{sp} is the first time step at which it is possible to arrive at i, no departure from i to any other mission way point can take place before t_i^{sp}. The shortest path time from the airport (mission way point 0) to a mission way point i makes it possible to define its related t_i^{sp}. The Dijkstra algorithm ([6]) is an effective algorithm that is able to find the shortest path between the airport and each other mission way point. Note that in this context a shortest path algorithm must take into consideration the presence of air traffic. The cost of an arc in the Dijkstra algorithm has to consider two components: the holding until the first available time step and the flight time. The Dijkstra algorithm has been properly adapted to satisfy this constraint and the shortest path from the airport to each mission way point has been found in a very short computing time. Moreover, the presence of air traffic is the second reason that makes a departure infeasible. The associated variables correspond to those considered in constraints (14) of the formulation (5)-(17). Finally no arrival at the airport can take place before the lower bound provided by the value of the optimal solution of the ATSP associated with the graph G_U. All these variables assume a value equal to zero in any solution of the problem. For this reason they are removed from the model in order to reduce its dimension significantly.

5.3. The Branch and Cut Algorithm

In this section we describe the main features of the proposed Branch and Cut algorithm (BC). Once the Variable Fixing procedure has been executed, the algorithm starts by solving the LP relaxation of model (5)-(17) and generating valid inequalities (18). If its solution is integer, an optimal solution has been found; otherwise an enumeration tree is constructed. At each node of such tree the algorithm tries to generate violated valid inequalities, by using the Padberg-Rinaldi Separation Procedure; once found, the inequalities are added to the current LP relaxation. The inequalities added at any node of the tree are valid for all the other nodes because inequalities (18) are valid for the formulation (5)-(17). Thus, at a given node, all the inequalities so far generated are incorporated in the lower bound computation. This Branch-and-Cut algorithm has been implemented using Ilog Cplex 12.1 with the default branching scheme. Even though this algorithm is able to solve to optimality many instances of the TDTSPPCA (see Section (7)), the time dependence of the decision variables leads to weak lower bounds.

6. Penalty-Based Model

To avoid the weakness of the time dependent formulation, the TDTSPPCA is reformulated as a TSP with Penalties associated with the routes. These penalties represent an alternative way to model the holdings necessary to avoid the conflicts.

6.1. Mathematical Formulation and main features

For a given mission way point $i \in M$, let $\tilde{H}_i$ be the set of all possible sequences of mission way points and holdings that present i as last reachable mission way point, i.e. the partial route that, starting from the airport, define a departure from i in a conflicting time step. Such conflict could be optimally avoided with a sequence of previous holdings: since the air traffic is known a priori, once a partial route is fixed, it is possible to find its optimal holdings distribution. Using this property, the new mathematical formulation extends the Dantzig-Fulkerson-Johnson TSP model ([5]) considering all possible conflicting partial routes and their related ”penalty” constraints. Finally, denoting with H_i the sequence of mission way points in $\tilde{H}_i$, we can define the coefficient k_h, i.e. the optimal holding of the element h of a conflicting partial route. Note that, in this context, a sequence of mission way points conflicting in i is a part of a Hamiltonian circuit and that it includes all mission way points until the successor of i since the holding duration is related to an arc and not only to a mission way point.

Two types of decision variables are used. Binary variable x_{ij} ($(i,j) \in V_U$) takes value 1 if the UAS uses the sub route (i,j) and 0 otherwise. Continuous variable p_i represents the holding over the mission way point i ($i \in M$). The MILP model of the TSPP reads as follows:

$$\min \quad \gamma \sum_{(i,j)\in V_U} f_{ij}x_{ij} + \sum_{i\in M} \beta_i p_i \tag{19}$$

$$\sum_{j\in M} x_{ji} = 1 \qquad i \in M \tag{20}$$

$$\sum_{j\in M} x_{ij} = 1 \qquad i \in M \tag{21}$$

$$\sum_{i\in Sub, j\in M\setminus Sub} x_{ij} \geq 1 \qquad 2 \leq |Sub| \leq |M| - 2, Sub \subseteq M \tag{22}$$

$$\sum_{(lj)\in H_i} (k_h - k_{min})(1 - x_{lj}) + p_h \geq k_h \qquad h \in H_i, H_i \in \tilde{H}_i, i \in M \tag{23}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in V_U \tag{24}$$

$$p_i \in [k_{min}, k_{max}] \qquad i \in M \tag{25}$$

As in (5), the first term of the objective function represents the total flight time through the mission way points; the second term describes the global weighted holding over all the visited mission way points as a ”penalty” that has to be added to the total cost of the route. To this end, it is necessary to add the ”penalty” constraints (23) to the standard constraints (20) -(22)) of the Dantzing-Fulkerson-Johnson ATSP formulation.(see [5] for further details).

Constraints (23) relate the route and the penalties (holdings). They enumerate all possible routes that could lead the UAS to a conflict. Clearly such constraints are exponential in number, but they can be separated using a specific procedure (see Section 6.2). To compute the coefficient k_h for each h of a given sequence H_i, it is necessary to solve the associated *Holding Assignment Problem* (HAP). In the specific case of not weighted holdings, it is sufficient to wait only at the last mission way point before the conflict until the first available departure time step.

6.2. Holding Assignment Problem

Given a conflicting sequence H_i and the amount of holding time steps over its last reachable mission way point i, the goal of the HAP is to find the best way to assign such holding time steps over the mission way points until i. We define K as a set of the holding time steps that have to be assigned. By reassigning the holdings, the departure and arrival time steps of the following mission way points change. Thus, these time steps could correspond to conflicting time steps. As a consequence, it is necessary to consider the presence of further conflicts that can occur by changing the holdings. Nevertheless this solution could be better than that associated with smaller holdings over more sensitive mission way points. To model this kind of conflicts, further penalties are added in a manner similar to that used for the Penalty-Based model. For a given mission way point $i \in M$ and the corresponding conflicting sequence H_i, we define Q as the set of all possible assignments of $|K|$ holding time steps over the mission way points until i. A generic assignment $q \in Q$ can be represented by the appropriate pairs (k, h), with $k \subseteq K$ and $h \in H_i$. Let k_h^q be the minimum holding duration over h for the assignment q, i.e. the time steps interval until the first available departure time step given the partial route H_i. We introduce a binary variable $z_{k,h}$ with value 1 if k holding time steps are assigned to the mission way point h and value 0 otherwise. Moreover we use the continuous variable p'_h to represent the further holding over h, and the binary variable u_h that takes value 1 if the holding h is used and value 0 otherwise. Given $i \in M$, H_i, K and Q, the corresponding MILP model reads as follow:

$$\min \quad \sum_{k \in K} \sum_{h \in H_i} \beta_h k z_{k,h} + \sum_{h \in H_i} \beta_h p'_h \tag{26}$$

$$\sum_{k\in K}\sum_{h\in H_i} kz_{k,h} + \sum_{h\in H_i} p'_h \geq |K| \tag{27}$$

$$u_h \geq z_{k,h} \qquad k \in K, h \in H_i \tag{28}$$

$$\sum_{k\in K} kz_{k,h} + p'_h \leq k_{max} \qquad h \in H_i \tag{29}$$

$$\sum_{k\in K} kz_{k,h} + p'_h \geq k_{min}u_h \qquad h \in H_i \tag{30}$$

$$\sum_{(k,h)\in q} B_1 k(1 - z_{k,h}) + p'_h \geq k^q_h \qquad h \in H_i, q \in Q \tag{31}$$

$$z_{k,h} \in \{0,1\} \qquad k \in K, h \in H_i \tag{32}$$

$$u_h \in \{0,1\} \qquad h \in H_i \tag{33}$$

$$p'_h \in [0, k_{max}] \qquad h \in H_i \tag{34}$$

For the considered mission way point $i \in M$, the objective function (26) minimizes the total cost of the associated holdings by considering two components: the holdings due to the reassignment and the holdings corresponding to the presence of further conflicts. Constraints (27) ensures that the global holding duration is consistent with the number of time steps that have to be assigned; Constraints (28) ensure that the appropriate variables u_h are activated. Constraints (29) and (30) represent the capacity constraints related, respectively, to the upper bound and the lower bound on the holding duration. Constraints (31) define the penalties by relating them with the reassignment; B_1 is a large number that can be set to k^q_h divided by k times the number of pairs (k, h) in q. To solve the HAP we separate the constraints (31) that are exponential in number. The algorithm starts by solving the model (26)-(34) without any constraint (31): if the solution does not imply further conflicts, the best solution is found. Otherwise, in each mission way point where a further conflict occurs the related holding until the first available time step is evaluated. Then the appropriate penalty k^q_h is added to the problem through the constraints (31) and the problem is re-solved. The algorithm iterates until a non-conflicting solution is found. The optimal values of the variables $z_{k,h}$ ($k \in K, h \in H_i$) allow then to compute the set of coefficients k_h for each $h \in H_i$ as follows:

$$k_h = \sum_{k\in K} k \; z_{k,h}$$

6.3. Cutting Plane Algorithm

In this section we describe the main features of the Cutting Plane algorithm (CP) proposed. The algorithm starts solving the model (19)-(25) without any constraints (23). This model corresponds to the $"TSP - relaxation"$ of the problem. If the route found presents no conflict the optimal route for the TDTSPPCA is the TSP route. Otherwise the route can be followed until the last reachable mission way point without conflicts, identifying in this manner a partial sequence of mission way points. Then the related HAP is solved using the procedure described in Section 6.2 in order to find the best holding distribution for the considered subsequence. The resulting holdings are imposed as penalties adding the appropriate constraints (23) and the problem is re-solved. This procedure is iteratively repeated until a non-conflicting route is found.

7. Computational Results

We tested the algorithms proposed in the previous sections 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 traffic movements per day. Computer Graphic tools have been used to model the air traffic scenario, to analyze the results and to rapidly evaluate the performances of the proposed algorithms. In this environment, 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.

The air traffic data represent the position and the altitude of departing, arriving and overflying aircraft during five different time periods. The days considered were: the 11th of November 2009 from 05:30 until 13:30 UTC, the 22nd 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.

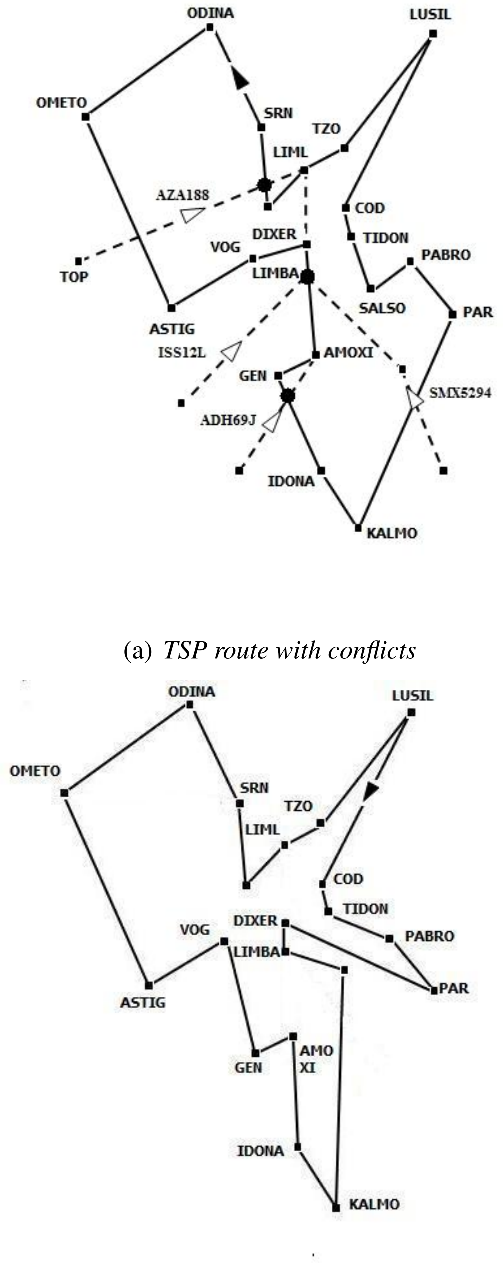
We considered two types of instances: Medium-Short and Medium-Long. The first type consists of visiting up to 20 mission way points, whose coordinates are randomly selected among the Navigation Points located within a circle having center in LIML and ray of 20NM (Nautical Miles). The second type consists of visiting up to 40 mission way points whose coordinates are randomly selected among the Navigation Points located within a circle having center in LIML and ray of 80NM. The costs of the holdings are assigned using the following considerations: if 1 is the unit time cost of the flight phase ($\gamma = 1$), a ground holding (in the airport) unit time cost is equal to 2. Instead, the airborne holdings have a cost decreasing according to their distance from the airport. The basic idea is that a UAS holding near the airport requires a more difficult management than a UAS holding far from the airport, because the former zone is more sensitive since the traffic is more concentrated.

To model a realistic update of the air traffic, a time step of one minute has to be set: this leads to a large number of time steps (more than 200) for a Medium-Long instances. The position and the altitude of each aircraft are updated at each time step. As reference, we use a UAS with performances similar to the

General Atomic MK9 Reaper, called Predator B. We set the cruise speed to 180 Knots and the maximum climb/descent rate to 1500 ft/min. Moreover we set k_{min} to 0 and k_{max} to 20 minutes.
As an example, the Medium-Long instance of the 11th of November 2009 is reported in Figures 3(a) and 3(b). In Figure 3(a) the TSP route is shown. The UAS is represented by a black triangle while the aircraft are represented by a white triangle. Dashed segments represent the aircraft routes and continuous segments represent the UAS route. In this case the time required for the UAS to overfly all 20 mission way points is 162 minutes (TSP optimal solution value) and 4 conflicts between the UAS and the aircraft are detected (represented by black bullets, one of them is covered by the flight SMX5294). In Figure 3(b) the route without conflicts is reported. The duration of the mission in the controlled air space is 171 minutes: a rerouting makes it possible to avoid the piloted air traffic.

An interesting example of UAS conflict resolution in such mission is reported in Figure 4. The points AMOXI, LIMBA and DIXER are lined up to Runway 36 of Linate; the arriving aircraft coming from South follow this route. Looking at the figures in sequence it is possible to recognize an arrival sequence concerning flights AZA2036, ACL324 and AZA2032. The UAS route includes the path between DIXER and LIMBA in the opposite direction. This path is performed by the UAS between ACL324 and AZA2032 without separation minima infringement.

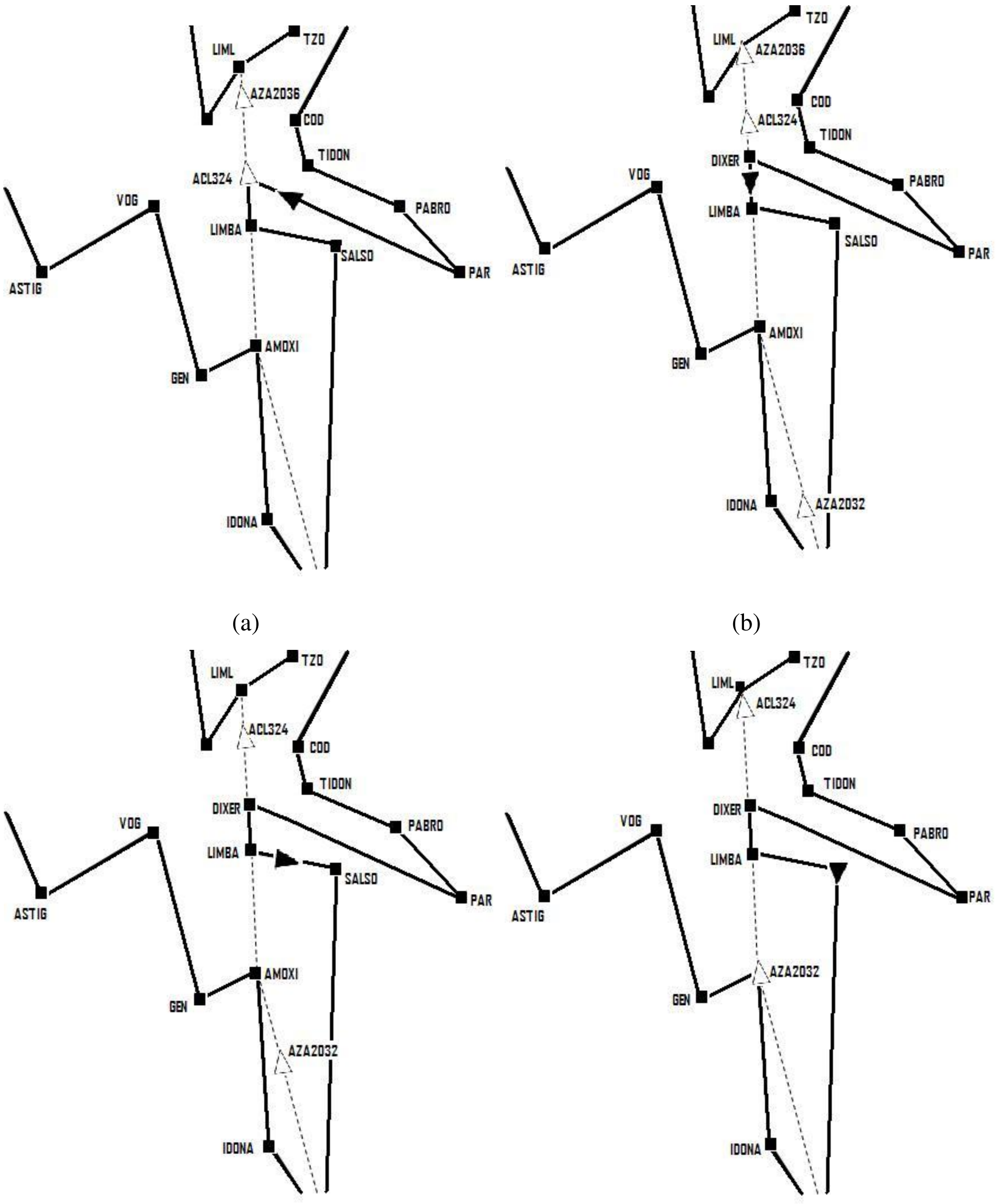
Tables 1 and 2 report the main characteristics and the results of the heuristic algorithms in the Medium-Short and Medium-Long instances, respectively. The name of an instance is composed by: the ICAO code of the Airport (LIML), the date, the simulation beginning time plus ”S” or ”L” for Medium-Short or Medium-Long instances respectively and the number of mission way points to visit (m). The other characteristics are: the number of simulated aircraft (n), the value in minutes of the TSP solution (Lower Bound on the value of the optimal solution in Controlled Air Space) and the number of conflicts on the TSP route. Then the tables report the heuristic solutions obtained by HTSP and by NN: for each of them two objective values and the related percentage gaps are reported. The first one (val) is the value of the solution found by the heuristic and the second one (LS) is the value of the Local Search applied to such solution. Finally the percentage gaps are computed between these upper bounds and the corresponding optimal solution value; for each group of instances the average gaps are also reported.
Table 3 reports the comparison between the Branch-and-Cut (BC) and the Cutting-Plane (CP) algorithms on the Medium-Short instances. The first two columns report the instance name and the optimal solution value found by both algorithms. Then we report the results corresponding to the BC algorithm. In particular, we report two lower bounds (LB_1 and LB_2) with the corresponding computing time. LB_1 is the



(a) *TSP route with conflicts*

(b) *Optimal route without conflicts*

Figure 3: Example of routes



(a)

(b)

(c)

(d)

Figure 4: Example of Avoidance

continuous relaxation value of model (5)-(17) while LB_2 is the continuous relaxation value of the same model with variable reduction and valid inequalities (18). Column T reports the computing time of the BC algorithm with a time limit of 1200 seconds (only in one case the algorithm was not able to solve the instance to proven optimality); then two columns report the number of nodes (Nodes) and the number of cuts (Cuts). Finally the table reports the solution time (T) and the number of iterations (It) of the CP algorithm. For each group of instances the average values are reported. Table 3 shows that on average it is better to use the CP algorithm for this group of instances but in two cases the computing time of the BC algorithm is lower. Table 4 reports the results on the Medium-Long Mission instances for only the CP algorithm, since the BC algorithm is not effective in these cases. In columns 2, 3 and 4 we report, respectively, the optimal solution value, the relative computing time and the number of iterations needed. In order to obtain good feasible solutions within short computing time, we executed the algorithm CP with a time limit equal to 1200 seconds. Since if the time limit is reached the corresponding solution is not feasible, we obtain a feasible solution by following the solution found until the first conflict is detected, and then we apply the heuristic algorithms HTSP and NN and the local search procedure (LS). Columns 5 and 6 report the lower bound found by the algorithm obtained within the time limit of 1200 seconds and the corresponding gap for the instances not solved to optimality within that time limit. For these 11 instances, the last four columns of the table (CP 1200sec + Heur) report the values and the gaps achieved applying the heuristic algorithms (without and with the local search procedure LS) to the solution found within the time limit (as described in Tables 1 and 2). In other words, in case of time limit, the infeasible solution found by the algorithm CP is completed using both HTSP and NN algorithms. The best value obtained among them is reported in column ”val” and the corresponding solution is used as input for the Local Search procedure (LS).

8. Future developments

Future works will focus on the extension to the dynamic case, on the use of other avoidance techniques and on the case of multi-UAS missions. In the dynamic case, the air traffic situation is not known in advance; however in a given step is known, for each aircraft, the route cleared by the air traffic controllers. In this way, a reliable and realistic forecast of the air traffic for the next few minutes is available and can be used for the planning of the next mission way point to be visited by the UAS. Moreover, the algorithms used for the static case, considered in the planning phase, can be easily extended to the dynamic case by defining two subsets of mission way points at each time step: the subset of points already visited and the subset of points that have to be visited. At a given time step, once the decision variables of the first

				HTSP				NN			
Instance	n	TSP	conf	val	gap	LS	gap	val	gap	LS	gap
LIML 11/11 05:30 S - 10	32	29	2	43	38.71	43	38.71	31	0.00	31.00	0.00
LIML 22/08 04:30 S - 10	28	29	0	29	0.00	29	0.00	51	75.86	47.00	62.07
LIML 25/08 04:30 S - 10	40	29	5	42	44.83	38	31.03	35	20.69	35.00	20.69
LIML 25/08 12:30 S - 10	34	29	1	34	9.68	34	9.68	31	0.00	31.00	0.00
LIML 26/08 04:30 S - 10	38	29	2	58	81.25	58	81.25	63	96.88	54.00	68.75
					34.9		32.1		38.7		30.3
LIML 11/11 05:30 S - 15	68	31	3	38	18.75	36	12.50	37	15.63	37.00	15.63
LIML 22/08 04:30 S - 15	61	31	1	37	8.82	37	8.82	37	8.82	37.00	8.82
LIML 25/08 04:30 S - 15	74	31	1	34	0.00	34	0.00	40	17.65	40.00	17.65
LIML 25/08 12:30 S - 15	71	31	2	38	15.15	33	0.00	36	9.09	35.00	6.06
LIML 26/08 04:30 S - 15	72	31	1	37	8.82	37	8.82	37	8.82	37.00	8.82
					10.3		6.0		12.0		11.4
LIML 11/11 05:30 S - 20	123	41	1	52	26.83	41	0.00	42	2.44	42.00	2.44
LIML 22/08 04:30 S - 20	117	41	3	49	13.95	46	6.98	43	0.00	43.00	0.00
LIML 25/08 04:30 S - 20	131	41	3	63	50.00	50	19.05	48	14.29	48.00	14.29
LIML 25/08 12:30 S - 20	127	41	0	41	0.00	41	0.00	42	2.44	42.00	2.44
LIML 26/08 04:30 S - 20	125	41	3	67	59.52	42	0.00	43	2.38	43.00	2.38
					30.1		5.2		4.3		4.3

Table 1: The table reports the main characteristics and the heuristic algorithm results for the Medium-Short Mission instances

subset are set to the values corresponding to the part of the route already flown, the problem is solved by determining the next point among the points of the second subset. This procedure is repeated until all the points have been visited. The main assumption of this approach is that the used algorithm must be able to solve the problem within a computing time smaller than the traffic updating interval. Tactical avoidance techniques will be modeled and used in case of changes in the aircraft routes or in the controller clearances that affect the route until the next mission way point (this corresponds to the route in progress that can not be changed using this methodology). The use of this approach based on the two subsets of points, will also allow changing tactically the point to be visited, or including new points or changing the position of the old ones. Rolling horizon techniques could be another approach to be investigated for the extension to the dynamic case. Thanks to temporal discretization supported by our approach, the algorithm can also be extended in order to consider moving targets (e.g., tracking animals or ships) or targets dependent upon weather conditions. Finally the uncertainty can be captured in the arc costs, e.g., giving higher costs to arcs which are likely to produce conflicts with the air traffic.

				HTSP				NN			
Instance	n	TSP	conf	val	gap	LS	gap	val	gap	val	gap
LIML 11/11 05:30 L - 10	32	55	2	68	6.25	67	4.69	86	34.38	86.00	34.38
LIML 22/08 04:30 L - 10	28	55	3	68	17.24	64	10.34	80	37.93	80.00	37.93
LIML 25/08 04:30 L - 10	40	55	3	63	14.55	63	14.55	90	63.64	85.00	54.55
LIML 25/08 12:30 L - 10	34	55	3	80	23.08	70	7.69	73	12.31	71.00	9.23
LIML 26/08 04:30 L - 10	38	55	3	68	19.30	67	17.54	85	49.12	85.00	49.12
					16.1		11.0		39.5		37.0
LIML 11/11 05:30 L - 15	68	98	1	117	15.84	105	3.96	128	26.73	128.00	26.73
LIML 22/08 04:30 L - 15	61	98	2	119	16.67	111	8.82	122	19.61	120.00	17.65
LIML 25/08 04:30 L - 15	74	98	2	100	1.01	100	1.01	126	27.27	126.00	27.27
LIML 25/08 12:30 L - 15	71	98	2	117	11.43	110	4.76	127	20.95	127.00	20.95
LIML 26/08 04:30 L - 15	72	98	1	106	7.07	104	5.05	122	23.23	121.00	22.22
					10.4		4.7		23.6		23.0
LIML 11/11 05:30 L - 20	123	162	4	189	10.53	189	10.53	197	15.20	197.00	15.20
LIML 22/08 04:30 L - 20	117	162	4	175	6.06	175	6.06	202	22.42	189.00	14.55
LIML 25/08 04:30 L - 20	131	162	4	192	15.66	176	6.02	204	22.89	204.00	22.89
LIML 25/08 12:30 L - 20	127	162	4	220	27.91	194	12.79	187	8.72	187.00	8.72
LIML 26/08 04:30 L - 20	125	162	3	204	24.39	181	10.37	201	22.56	190.00	15.85
					16.9		9.2		18.4		15.4
LIML 11/11 05:30 L - 25	211	176	5	190	2.15	190	2.15	220	18.28	220.00	18.28
LIML 22/08 04:30 L - 25	204	176	2	204	10.27	204	10.27	221	19.46	221.00	19.46
LIML 25/08 04:30 L - 25	216	176	4	192	4.35	192	4.35	220	19.57	196.00	6.52
LIML 25/08 12:30 L - 25	212	176	6	223	20.54	223	20.54	232	25.41	210.00	13.51
LIML 26/08 04:30 L - 25	214	176	3	204	11.48	204	11.48	211	15.30	211.00	15.30
					9.8		9.8		19.6		14.6
LIML 11/11 05:30 L - 30	213	179	5	214	12.63	214	12.63	229	20.53	229.00	20.53
LIML 22/08 04:30 L - 30	207	179	4	207	9.52	207	9.52	233	23.28	233.00	23.28
LIML 25/08 04:30 L - 30	220	179	3	203	9.73	203	9.73	215	16.22	211.00	14.05
LIML 25/08 12:30 L - 30	214	179	5	221	13.92	221	13.92	216	11.34	212.00	9.28
LIML 26/08 04:30 L - 30	216	179	4	205	8.47	205	8.47	211	11.64	211.00	11.64
					10.9		10.9		16.6		15.8
LIML 11/11 05:30 L - 35	229	191	6	225	14.80	218	11.22	232	18.37	231.00	17.86
LIML 22/08 04:30 L - 35	226	191	4	236	21.65	236	21.65	253	30.41	253.00	30.41
LIML 25/08 04:30 L - 35	271	191	4	216	9.09	216	9.09	244	23.23	231.00	16.67
LIML 25/08 12:30 L - 35	245	191	7	253	30.41	220	13.40	253	30.41	253.00	30.41
LIML 26/08 04:30 L - 35	250	191	2	228	14.57	225	13.07	244	22.61	244.00	22.61
					18.1		13.7		25.0		23.6
LIML 11/11 05:30 L - 40	247	199	9	228	9.09	228	9.09	259	23.92	258.00	23.44
LIML 22/08 04:30 L - 40	232	199	6	262	27.18	251	21.84	280	35.92	280.00	35.92
LIML 25/08 04:30 L - 40	282	199	6	220	6.80	220	6.80	263	27.67	263.00	27.67
LIML 25/08 12:30 L - 40	257	199	10	226	10.78	224	9.80	263	28.92	253.00	24.02
LIML 26/08 04:30 L - 40	261	199	4	232	12.62	230	11.65	250	21.36	247.00	19.90
					13.3		11.8		27.6		26.2

Table 2: The table reports the main characteristics and the heuristic algorithm results for the Medium-Long Mission instances

		BC							CP	
Instance	opt	LB_1	T_1	LB_2	T_2	T	Nodes	Cuts	T	It.
LIML 11/11 05:30 S - 10	31	26	0.0	29	0.0	3.8	38	6	6.7	34
LIML 22/08 04:30 S - 10	29	26	0.1	29	0.0	1.5	0	6	0.1	1
LIML 25/08 04:30 S - 10	29	26	0.0	29	0.0	0.9	0	7	0.9	8
LIML 25/08 12:30 S - 10	31	26	0.0	29	0.0	5.2	30	6	19.5	53
LIML 26/08 04:30 S - 10	32	26	0.1	29	0.0	104.8	1290	37	71.2	136
						23.2	271.6	12.4	19.7	46.4
LIML 11/11 05:30 S - 15	32	28	0.2	32	0.0	49.3	0	6	0.2	2
LIML 22/08 04:30 S - 15	34	28	0.3	32	0.0	516.4	273	23	33.4	58
LIML 25/08 04:30 S - 15	34	28	0.2	32	0.0	1200.0	1233	75	16.7	47
LIML 25/08 12:30 S - 15	33	28	0.1	32	0.0	103.1	41	6	2.7	12
LIML 26/08 04:30 S - 15	34	28	0.2	32	0.0	356.0	266	33	31.4	58
						445.0	362.6	28.6	16.9	35.4
LIML 11/11 05:30 S - 20	41	37	0.4	41	0.1	65.5	578	32	0.6	2
LIML 22/08 04:30 S - 20	43	37	0.4	42	0.1	962.5	2864	44	69.7	118
LIML 25/08 04:30 S - 20	42	36	0.3	42	0.1	32.7	259	23	6.3	15
LIML 25/08 12:30 S - 20	41	37	0.3	41	0.1	368.3	5089	53	0.1	1
LIML 26/08 04:30 S - 20	42	37	0.3	42	0.1	89.3	316	29	1.2	4
						303.6	1821.2	36.2	15.6	28.0

Table 3: The table reports the comparison between BC algorithm and CP algorithm for the Medium-Short Mission instances.

9. Conclusions

In this paper we study the Time Dependent Traveling Salesman Planning Problem in Controlled Airspace. We have proposed and compared two different MILP models and designed two exacts algorithms to solve them. The first one is a time dependent formulation of the problem able to model a real update of the air traffic. Then, in order to avoid the weakness of the time dependent formulation, we propose a second formulation based on penalties. The methodology we propose is innovative since the conflict resolution uses an ATM approach, i.e. we model the avoidances through manoeuvres planned by Air Traffic Controllers in order to optimize a conflict free route. Some heuristic algorithms have also been proposed and tested. Computational results show how the proposed algorithms are able to solve real-world instances in short computing time, which makes the proposed algorithms suitable for extensions to real time re-optimization.

		CP		CP 1200sec		CP 1200sec + Heur			
Instance	opt	T	It.	lb_2	gap	val	gap	LS	gap
LIML 11/11 05:30 L - 10	64	59.2	118	*	*	*	*	*	*
LIML 22/08 04:30 L - 10	58	1.3	14	*	*	*	*	*	*
LIML 25/08 04:30 L - 10	55	0.1	2	*	*	*	*	*	*
LIML 25/08 12:30 L - 10	65	98.3	203	*	*	*	*	*	*
LIML 26/08 04:30 L - 10	57	0.8	18	*	*	*	*	*	*
		32.0	71.0						
LIML 11/11 05:30 L - 15	101	1.9	19	*	*	*	*	*	*
LIML 22/08 04:30 L - 15	102	1.8	17	*	*	*	*	*	*
LIML 25/08 04:30 L - 15	99	0.1	2	*	*	*	*	*	*
LIML 25/08 12:30 L - 15	105	81.1	145	*	*	*	*	*	*
LIML 26/08 04:30 L - 15	99	0.4	5	*	*	*	*	*	*
		17.1	37.6						
LIML 11/11 05:30 L - 20	171	4486.0	346	170	0.58	171	0.00	171	0.00
LIML 22/08 04:30 L - 20	165	4.1	21	*	*	*	*	*	*
LIML 25/08 04:30 L - 20	166	11.8	35	*	*	*	*	*	*
LIML 25/08 12:30 L - 20	172	3545.0	352	171	0.58	172	0.00	172	0.00
LIML 26/08 04:30 L - 20	164	1.2	12	*	*	*	*	*	*
		1609.6	153.2						
LIML 11/11 05:30 L - 25	186	2456.9	456	186	0.00	186	0.00	186	0.00
LIML 22/08 04:30 L - 25	185	1111.9	235	*	*	*	*	*	*
LIML 25/08 04:30 L - 25	184	130.8	64	*	*	*	*	*	*
LIML 25/08 12:30 L - 25	185	894.6	543	*	*	*	*	*	*
LIML 26/08 04:30 L - 25	183	122.9	121	*	*	*	*	*	*
		943.4	283.8						
LIML 11/11 05:30 L - 30	190	6518.1	1231	187	1.58	214	12.63	213	12.11
LIML 22/08 04:30 L - 30	189	1176.2	632	*	*	*	*	*	*
LIML 25/08 04:30 L - 30	185	45.1	81	*	*	*	*	*	*
LIML 25/08 12:30 L - 30	194	2843.4	1987	188	3.09	221	13.92	221	13.92
LIML 26/08 04:30 L - 30	189	1124.2	427	*	*	*	*	*	*
		2341.4	871.6						
LIML 11/11 05:30 L - 35	196	326.4	181	*	*	*	*	*	*
LIML 22/08 04:30 L - 35	194	29.2	48	*	*	*	*	*	*
LIML 25/08 04:30 L - 35	198	11546.3	771	194	2.02	218	10.10	215	8.59
LIML 25/08 12:30 L - 35	194	11.1	22	*	*	*	*	*	*
LIML 26/08 04:30 L - 35	199	4455.1	507	193	3.02	220	10.55	204	2.51
		3273.6	305.8						
LIML 11/11 05:30 L - 40	209	123327.4	2931	204	2.39	228	9.09	228	9.09
LIML 22/08 04:30 L - 40	206	18408.3	949	203	1.46	262	27.18	242	17.48
LIML 25/08 04:30 L - 40	206	17140.3	937	203	1.46	220	6.80	206	0.00
LIML 25/08 12:30 L - 40	204	114.0	97	*	*	*	*	*	*
LIML 26/08 04:30 L - 40	206	49235.2	867	201	2.43	232	12.62	232	12.62
		41645.0	1156.2						

Table 4: Table reports the results of the CP algorithm for the Medium-Long Mission instances.

References

[1] Abeledo, H., R. Fukasawa, A. Pessoa, E. Uchoa. 2013. The time dependent traveling salesman problem: polyhedra and algorithm. *Mathematical Programming Computation* **5**(1) 27–55.

[2] Albiach, J., J. Sanchis, D. Soler. 2008. An asymmetric TSP with time windows and with time-dependent travel times and costs: An exact solution through a graph transformation. *European Journal of Operational Research* **189**(3) 789–802.

[3] Beasley, J.E. 1981. Adapting the savings algorithm for varying inter-customer travel times. *Omega* **9**(6) 658 – 659.

[4] Chen, H., C. Hsueh, M. Chang. 2006. The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review* **42**(5) 383–408.

[5] Dantzig, G., R. Fulkerson, S. Johnson. 1954. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America* **2**(4) pp. 393–410.

[6] Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK* **1**(1) 269–271.

[7] Fleischmann, B., M. Gietz, S. Gnutzmann. 2004. Time-Varying Travel Times in Vehicle Routing. *Transportation Science* **38**(2) 160–173.

[8] Gouveia, L., S. Voss. 1995. A classification of formulations for the (time-dependent) traveling salesman problem. *European Journal of Operational Research* **83**(1) 69 – 82.

[9] Gutin, G., A. Punnen, A. Barvinok, E. Gimadi, A. Serdyukov. 2002. The Traveling Salesman Problem and Its Variations.

[10] Kuo, Y., C. Wang, P. Chuang. 2009. Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds. *Computers & Industrial Engineering* **57**(4) 1385–1392.

[11] Malandraki, C., M. S. Daskin. 1992. Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms. *Transportation Science* **26**(3) 185–200.

[12] Malandraki, C., R.B. Dial. 1996. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European Journal of Operational Research* **90**(1) 45 – 55.

[13] Mueller, E., C. Santiago, A. Cone, T. Lauderdale. 2013. Effects of UAS Performance Characteristics, Altitude, and Mitigation Concepts on Aircraft Encounters and Delays. *Air Traffic Control Quarterly* **21**(1).

[14] Padberg, M., G. Rinaldi. 1991. A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems. *SIAM Review* **33**(1) 60.

[15] Park, Y. 2000. A solution of the bicriteria vehicle scheduling problems with time and area-dependent travel speeds. *Computers & Industrial Engineering* **38**(1) 173–187.

[16] Persiani, C. A., S. Bagassi. 2013. Route planner for unmanned aerial system insertion in civil non-segregated airspace. *Journal of Aerospace Engineering (to appear)* .

[17] Picard, J. C., M. Queyranne. 1978. The Time-Dependent Traveling Salesman Problem and Its Application to the Tardiness Problem in One-Machine Scheduling. *Operation Research* **26** 86–110.

[18] Soler, D., J. Albiach, E. Martinez. 2009. A way to optimally solve a time-dependent Vehicle Routing Problem with Time Windows. *Operations Research Letters* **37**(1) 37–42.

[19] Stecco, G., J. Cordeau, E. Moretti. 2007. A branch-and-cut algorithm for a production scheduling problem with sequence-dependent and time-dependent setup times. *Computers & Operations Research* **35**(8) 2635–2655.