TNO 2013 S10139

The UAV-Mission Planning Problem
with Time Windows and Stochastic Fuel Consumption

Master Thesis

January 31, 2013

Corrinne Luteyn

321671

Supervisors TNO:

L. Evers MSc

Dr. A.I. Martins Botto de Barros

Supervisors EUR:

Dr. D. Huisman

L. Evers MSc

ERASMUS UNIVERSITEIT ROTTERDAM

Abstract

In military operations Unmanned Aerial Vehicles (UAVs) are used for reconnaissance of target locations in the area of operations. These target locations each have their own priority. In this thesis the UAV-Mission Planning Problem (UAV-MPP) is addressed where the fuel usage of the flight between each pair of targets is known a priori only probabilistically and the information about the target locations can only be obtained within their assigned time window. The goal is to maximize the total gathered information value during a flight which is restricted by the fuel capacity of the UAV. This problem can be modeled by the Stochastic Orienteering Problem with Time Windows (SOPTW), which is both practically and theoretically relevant.

To solve this problem, two different approaches are presented. The first approach constructs an initial tour before the flight which is adjusted to the realized fuel usages during the flight. To construct an initial tour, different stochastic programming models are used including three variants of a chance constrained programming model and two variants of a recourse model. The second approach is a One-Step-Ahead Routing (OSAR) approach in which the next location is determined after recording the previous. A case study is performed to evaluate and compare the presented methods.

Contents

Part I

Background

In this first part we will introduce the problem addressed in this thesis and we will discuss both the military and the scientific background of the problem. In Chapter 1 the practical and theoretical relevance of the UAV mission planning problem with time windows and stochastic fuel consumption is discussed. After that, we will give a formal description of the problem in Chapter 2. In Chapter 3 we will review the found literature about problems related to our research.

1. Introduction

During military operations, Unmanned Aerial Vehicles (UAVs) are used for reconnaissance, especially when it is too dangerous for a manned aircraft. The flight of a UAV can be partly controlled by computers on board or remotely by an operator on the ground or in another vehicle. The technology in the scout vehicles is able to capture both full motion videos and still imagery of valuable targets in the area of operations. Each target in the area has its own given information value. It is therefore necessary to create a route before the flight, taking into account all restrictions such as fuel capacity and time windows of the targets. Since it is unlikely that a UAV can visit all targets in one flight, a tour which visits only a subset of targets, should be planned. The main objective of this tour is to maximize the total gathered reward from the planned targets. Therefore, UAV-mission planning can be seen as an optimization problem. The value of the parameters can be deterministic and therefore known beforehand or parameters can be random variables that follow a certain probability distribution. In that case, their values are known a priori only probabilistically.

The basic UAV-mission planning problem (UAV-MPP), where all parameters are assumed to be known, can be modeled like an Orienteering Problem (OP), according to Evers et al. [1]. The name of this problem comes from the family of sports, orienteering, as mentioned by Chao et al. in [2]. Players of such a game have to visit as much as possible control points and go back to the start location within a certain time frame. To navigate in the rough terrain, they have a compass and a map. Each of the control posts has an associated score and the player who gathered the most points, is the winner of the game.
The OP can be seen as a combination between two widely known combinatorial optimization problems, the Knapsack Problem (KP) and the Travelling Salesman Problem (TSP). It can also be seen as a generalization of the TSP, where it is not necessary to visit all vertices during the tour.

In the following part of this section we will discuss four practical elements which results in four variants of the basic UAV-MPP.

First of all, due to several circumstances during the flight, the realized fuel consumption of a UAV is not fixed in reality. For instance, the wind direction during a UAV-mission can affect the fuel consumption. Flying headwind will be more costly, while flying downwind will be less consuming, which means that the fuel usage on an arc can fluctuate. Since the wind direction is quite predictable in advance, this can be used in the planning of the tour. Sudden changes in the wind direction are unfortunately not predictable. It is therefore necessary to model the fuel consumption with a probability distribution, in order to consider both positive and negative unforeseen events. Since the fuel consumption is not deterministic anymore, but a random variable, the problem is stochastic. Therefore, we will refer to this variant as the Stochastic Orienteering Problem (SOP).

Through telephone tabs, social media and other intelligence, sometimes more information about the targets is available. When this is additional information about the time of the activity at a target, this can be modeled like a time window. A time window is given by an earliest and a latest time a target can be visited. Outside this time window, there is less or no information at that particular target and thus the reward of the target is less or nothing. It is referred to as 'time-sensitive-targeting'. An example of a realistic situation in a UAV-mission is that the operators have picked up some information about a possible meeting of a suspicious group of people. This meeting will probably take place between half past three and half past four. To get more information about this meeting, the UAV has to record this target within the given time window. Before and after this time window, there is less or possibly no useful information at this target and therefore recording in those moments can be not valuable. Because time windows are introduced, the optimization of the UAV-MPP becomes more difficult. Therefore, (meta-)heuristics are often used to solve this extension of the problem, which can be modeled by the Orienteering Problem with Time Windows (OPTW).

In the case of large missions, there is a possibility to do a UAV-mission with multiple vehicles. All of these vehicles have their own route and visit their own group of targets. Since it is not possible to earn a reward of a target twice, it is advisable to visit each target only once. The Team Orienteering Problem (TOP) can be used to model this extension of the UAV-MPP.

The last extension of the basic problem is the online version of the UAV-MPP. This means that there is a changing set of targets. Targets in the area may appear or disappear during the flight. When new target information comes available, the tour of the UAV has to be reoptimized. At the moment, there is less literature available about the Online Orienteering Problem (OOP), but the research on this problem is in progress.

In this thesis, a combination of these four extensions is investigated. We will focus on the UAV-mission planning problem with time windows and stochastic fuel consumption. Note that fuel consumption in our research is a measure for time. All travel costs and time windows are given in fuel units. This variant can be modeled by the Stochastic Orienteering Problem with Time Windows (SOPTW). To the best of our knowledge, no research is available on this problem. In UAV mission planning, this variant is certainly practically relevant as we discussed before.
Theoretically this problem is relevant as well, since proposed methods for the SOP cannot be used for the SOP with time windows. In these methods, time windows are not considered and therefore applying these algorithms can violate the time window constraint. However, solution approaches for a variant of the OP with time windows can be used for the same variant without time windows. Furthermore, due to the stochasticity of the fuel consumption, deterministic solution approaches of the OPTW cannot be used for the SOPTW. By planning an initial tour, the uncertainty in the parameters is not taken into account. This can result in violated time windows and an exceeded fuel limit during the flight. Since an approximation for the distribution of the fuel consumption can be made, it is useful to consider this information in an a priori planning.

To solve the UAV-MPP with time windows and stochastic fuel consumption, there are different approaches. We will discuss the two most widely used.

In the first approach the problem is divided into two parts. The offline problem is the part before the flight. During this part, an initial tour is constructed. This can be done either exactly or heuristically. All a priori known information, such as time windows and probability distribution functions of the fuel consumption, is included in these calculations. Since the targets are known in advance, a large amount of computation time is available. It is therefore not a problem if creating such a tour takes several hours.
During the flight, information about the realized values of the fuel consumption comes available. It could be necessary to adjust the initial tour, due to disappointing realizations caused by poor flight conditions. Sometimes, especially when the circumstances are comfortable to flight, it is possible to visit more targets than planned in the initial tour. Also in that case, a new tour has to be created.

This problem is called the online part of the UAV-MPP. The calculations in this part have to be done very quickly, since the computation time in real-time situations is limited. It is therefore necessary to implement a fast, but effective heuristic.
This approach is commonly used in research for routing problems where new targets may appear during the flight. Every time a new target appears the initial tour is reoptimized, like is done by Lorini et al. [3].

Another approach is to determine only the next target to visit in the tour. This can be done by constructing a new complete tour at the moment of departing a target. Just only the first target of this tour is considered. The advantage of this method is that the whole area is taken into account. Another method is to consider only the neighborhood of the current target and select one of the unvisited targets in this area.
Since the calculations in this approach have to be done during the flight and therefore very quickly, like during the online part of the first approach, it is possible to use the same methods. The main difference is that in this second approach reoptimizing takes place after recording just one target, while for the online part of the first approach a strategy is selected when the tour should be reoptimized. This strategy could be reoptimized after recording just one or two targets, but also construct a new tour only three times during the flight.
In the literature this approach is mostly used for problems with time-dependent or stochastic travel times, e.g. costs, such as done by Garcia et al. in [4] and by Gao and Huang in [5].

The goal of this thesis is to construct and implement several methods, both exactly and heuristically, for the discussed approaches of the UAV-mission planning problem with time windows and stochastic fuel consumption. To test the different methods, a case study is done. A Monte-Carlo simulation is used for each of the several datasets. The empirical results of this case study are summarized using some quality measures.

This thesis is organized as follows. In the rest of this part, the background of the problem is further investigated. In the second part of this thesis several methods for solving the offline problem are proposed, while in the third part some strategies for the real-time adaptive routing are presented. The case study, which is executed to test and compare the different proposed approaches, and its results are discussed in Part IV. We will end up in Part V with a conclusion and some outlines for further research.

2. Problem description

In this chapter we will give a formal description of the problem.

In the SOPTW, N represents a given set of targets. The starting point is denoted by vertex s and the ending point by vertex t. For notational convenience, set V is defined as $N \cup \{s\} \cup \{t\}$ and $|V|$ as its cardinality. In the case of UAV-missions the start and end point coincide and this is the depot, most of the time. Each target $i \in V$ has a given information (profit) value p_i. The value of the start and end point is equal to zero. A time window $[E_i, L_i]$ is assigned to each target. Recording of a target is only possible within its time window. We assume that the time windows of all targets are hard, which means that before the earliest time E_i and after the latest time L_i the information value is equal to zero. When the vehicle arrives too early at a target, there is the possibility to wait until the target is 'open', which means that recording is possible.

The given set A contains the flight paths (i, j) between each pair of locations in V, where $i \neq j$. There are no arcs terminating in vertex s and no arcs originating from vertex t. The fuel consumption on arc $(i, j) \in A$, denoted by f_{ij}, is the fuel usage needed both to travel from location i to location j and to gather information at location j. Note that recording is unnecessary and therefore excluded if location j is the end point. These parameters are assumed to be random variables that follow a certain probability distribution. Their values are known a priori only probabilistically. Since the fuel capacity of a UAV is limited by a given amount of fuel F, it is possible that not all targets can be recorded. In the SOPTW a tour is determined which maximizes the total expected collected information, from the starting point, passing along a subset of targets, to the end point, taking into account the fuel uncertainty. The methods to solve the SOPTW, discussed in this thesis, handle the fuel capacity and the time windows for the targets differently.

The SOPTW can also be seen as a graph, $G = (V, A)$, where V is the vertex set and A is the arc set. In this definition an information value p_i and a time window $[E_i, L_i]$ are associated with each vertex $i \in V$ and a probabilistic fuel consumption f_{ij} with each arc $(i, j) \in A$. To solve the SOPTW, a Hamiltonian path $G' \subset G$ over a subset of V, with a fixed start s and end point t, has to be determined, in order to maximize the total expected gathered information value. For this optimization process the fuel uncertainty and limitations on the total fuel usage on this path and the departure time of the vehicle from each target should be taken into account.

3. Literature review

In this chapter we will review the available literature about the problems which are related to our research. For all of the discussed problems we will describe the differences to our research and the difficulties in implementing the proposed methods.

3.1 The Orienteering Problem

Since the introduction of the Orienteering Problem (OP) by Tsiligrides in [6], a lot of research is done on this problem. Several exact, meta-heuristic and heuristic approaches have been proposed. During the first years, research on the OP was focused on finding new solution methods as found by Tsiligrides [6] and Golden et al. [7]. Later on, extensions of the basic problem were introduced, such as the Orienteering Problem with Time Windows (OPTW) by Kantor and Rosenwein [8], the Team Orienteering Problem (TOP) by Chao et al. [2] and the combination of these two, the Team Orienteering Problem with Time Windows (TOPTW) by Vansteenwegen et al. [9]. These are all deterministic variants of the OP. In such variants the travel time between two targets is assumed to be fixed and therefore fluctuations are not taken into account. As we already mentioned in the introduction, the application of solution approaches for these extensions to our problem, carries the risk that time windows will be violated or the UAV is running out of fuel under disappointing circumstances. When using these methods in our research, at least a worst scenario control has to be provided. A detailed overview of these variants of the OP can be found in the survey of Vansteenwegen et al. [10].

The last few years another extension of the OP has been investigated, the Stochastic Orienteering Problem (SOP). In this variant, the deterministic travel and service times between two targets are replaced by random variables, which follow a certain probability distribution. These times, and therefore the travelling costs, are known a priori only probabilistically. For the OP related problems, this is first introduced and solved by Teng et al. [11], who present the Time-Constrained Traveling Salesman Problem with Stochastic Travel and Service Times (TCTSP). Their model is limited to discrete travel and service time distributions. Another related problem is the Stochastic Selective Travelling Salesperson Problem (SSTSP), introduced by Tang and Miller-Hooks in [12]. In the SSTSP, the travel and service times are stochastic and the authors propose both exact and heuristic methods for solving the SSTSP. Campbell et al. [13] present the Orienteering Problem with Stochastic Travel and Service Times (OPSTS). In this paper, for some special cases, a dynamic programming model is used for solving the OPSTS exactly. Since the running time for this model grows exponentially with the number of nodes, they suggest a variable neighborhood search heuristic (VNS) to solve the larger and more realistic cases. Other methods to handle the uncertainty are Robust Planning, applied to the OP by Evers et al. [1], and Stochastic Programming. For that, the Two-Stage Orienteering Problem (TSOP) is presented by Evers et al. [14]. In their paper, they compare the Robust Orienteering Problem (ROP) with the TSOP.

Since time windows are not considered in the SOP, the discussed methods cannot be used directly in our research. For the most of these methods, adjusting is difficult or impossible. However, a VNS heuristic can be adapted to our problem relatively easily, since there are route improving heuristics for routing problems with time windows available in literature, such as presented by Potvin and Rousseau [15].

To the best of our knowledge, there is no research available about the Stochastic Orienteering Problem with Time Windows (SOPTW). This extension of the OP contains both stochastic travel times and a time window for each target. As there are many practical situations, e.g. UAV mission planning, where this modeling is more realistic than the deterministic variant, it is interesting to investigate this problem. It is therefore necessary to analyze the available literature on related problems.

Closely related to the SOP is the Time-dependent Orienteering Problem (TDOP), which is introduced by Fomin and Lingas in [16]. In this problem, the travel and service times are dependent of the time of departure. Therefore, the travel times can be described with an explicit function which is known in advance. The authors of this article give a detailed proof of the accuracy of their greedy heuristic. An application of the TDOP in the field of discrete manufacturing is given by Wang and Tang [17]. This paper considers a hybrid meta-heuristic for the prize-collecting single machine scheduling problem with sequence-dependent setup times. Del Bimbo and Pernici [18] describe another application of the TDOP. They use it to optimize the saccades planning for distant target identification. Recently, Li et al. [19] proposed a mathematical model and an exact algorithm to solve the TDOP.

Also for this basic problem extensions are introduced. For multiple parallel tours, the Time-dependent Team Orienteering Problem (TDTOP) is presented by Li in [20]. The Time-dependent Orienteering Problem with Time Windows (TDOPTW), which is introduced by Garcia et al. [4], is useful for tourists to plan a tour for one day in a city with a lot of Points Of Interest (POIs). Tourists can choose to travel between POIs by public transportation or on foot. In this paper, a hybrid approach, which combines an Iterated Local Search (ILS) with a precalculated average travel time matrix, is used for real-time route planning. When a tourist stays for several days in the city it is necessary, according to Garcia et al. [21], to solve a Time-dependent Team Orienteering Problem with Time Windows (TDTOPTW).

In contrast to our problem, the TDOP and its variants are deterministic problems, because all parameters are known in advance. These problems can be solved exactly or heuristically. Some proposed heuristic methods, such as the Hybrid Approach, presented by Garcia et al. [4], can also be applied to the SOPTW, since this method uses only the average travel time between two targets.

3.2 The Vehicle Routing Problem

Also Vehicle Routing Problems (VRPs) are related to the (Team) Orienteering Problem. A VRP can be seen as a TOP with two extra constraints. First of all, all targets must be visited by a vehicle. Secondly, each vehicle has a fixed capacity. Besides that, the objectives of the two problems are also different. The main goal of the OP is to maximize the total reward, while that of the VRP is to minimize the total number of vehicles or the total number of kilometers. A lot of research on the VRP and its extensions is done. In this literature review only the extensions which are closely related to the SOPTW are discussed.

The Vehicle Routing Problem with Time Windows (VRPTW) is introduced by Baker in [22]. Thereafter, a large number of exact and (meta-)heuristic solution approaches are proposed, e.g. by Desrosiers et al. [23] and Solomon [24]. The issue with this deterministic problem is the same as with the (T)OPTW, since in both the travel times are fixed. Therefore, most of the solution approaches for this problem cannot be applied to the SOPTW. An interested reader can refer to the overview of El-Sherbeny [25].

Another related extension of the VRP is the Stochastic Vehicle Routing Problem (SVRP), first studied by Laporte et al. in [26]. This problem consists of the planning of optimal vehicle routes with probabilistic travel and service times. In the found literature, several methods are used to solve this problem. Chance-constrained programming (CCP) is an often used method. Laporte et al. [26] are the first who implement an L-Shaped algorithm to successfully solve the CCP-formulation of the SVRP. Creating a modeling scheme based on queuing theory, as has been done by Woensel et al. in [27], is another way to solve the SVRP. Travellers' equilibrium is used by Connors and Sumalee [28] to study the stochasticity of the travel times. Since in this stochastic VRP time windows are not considered, like in the SOP, application of the proposed methods for the SVRP is very hard. Therefore, it could be more useful to analyze the presented solution approaches for the SVRP with time windows.

The Stochastic Vehicle Routing Problem with Time Windows (SVRPTW) has been introduced more recently. Wong et al. [29] propose a two-stage stochastic integer program with recourse for this problem. In their paper, only discrete random distributed travel times are taken into account. To handle all probability distributions, Ando and Taniguchi [30] use in their case study a Genetic Algorithm (GA) to solve the problem. Also Gao [31] uses a GA to solve the CCP-formulation of the SVRPTW. In order to accelerate the solution process, Li et al. [32] present a Tabu Search-based heuristic. In these last four referred articles, the time windows are assumed to be hard, which means that a location can only be visited within its time window. In the found literature, there are also researchers who assume that in time windows in the SVRPTW are soft. By incurring a penalty, a location can be visited outside its time window. These penalties are developed using a fixed cost, a linear cost and sometimes a quadratic loss penalty. This Stochastic Vehicle Routing Problem with Soft Time Windows (SVRPSTW) can be found by Hsu et al. in [33], by Russell and Urban in [34] and by Taş et al. in [35].
Presented solution approaches for the SVRPTW, such as Chance-constrained Programming by Li et al. [32], can be adapted to the SOPTW relatively easily. In the following chapter we will present a CCP-formulation for the SOPTW, based on these adjustments.

Part II

The offline problem

In the second part of this thesis we will discuss the offline part of the UAV-MPP with time windows and stochastic fuel consumption, where an initial tour is constructed taking into account the stochasticity of the fuel consumption. In the first two chapters of this part we will introduce some different approaches to create an initial tour. In Chapter 4 a chance-constrained programming model for the SOPTW is presented, while in Chapter 5 two stochastic programming models with recourse are introduced. In Chapter 6 we will discuss some solution methods for the introduced models.

4. Chance-Constrained Programming

In this chapter we will discuss how Chance-Constrained Programming (CCP) can be used to create an initial tour. First, we will give a mathematical formulation for the deterministic OPTW. After that, a CCP-formulation is presented. Finally, we will introduce a quadratically constrained programming reformulation of the model.

4.1 Deterministic Orienteering Problem with Time Windows

As discussed in Chapter 2, in the SOPTW the fuel consumption on a flight path between two locations is not fixed, but a random variable that follows a certain probability distribution. Since the value of these parameters is known a priori only probabilistically, this is a stochastic problem. When the fuel consumption is assumed to be fixed and known in advance, the problem is deterministic. To start our research on the SOPTW, it will be useful to consider first a mathematical formulation for the deterministic OPTW. Therefore we present a formulation of the OPTW making use of the notation introduced in Chapter 2. As previously mentioned, the only difference is that the fuel consumption f_{ij} is no longer stochastic, but deterministic in this case.

To formulate a Mixed Integer Programming (MIP) for the OPTW, the following decision variables are defined. First of all, $x_i = 1$, if target i is visited on the tour and 0 otherwise. Decision variables $y_{ij} = 1$, if target j is visited right after target i, this means that the flight path from target i to target j is selected in the tour and 0 otherwise. Finally, d_i is the departure time of the vehicle, e.g. the UAV, after recording target i. The MIP formulation of the OPTW (P4.1) is the following:

$$\text{(P4.1)} \qquad max \sum_{i \in N} p_i x_i \qquad (4.1)$$

$$subject\ to \sum_{\{j:(i,j)\in A\}} y_{ij} = x_i \qquad \forall i \in V \backslash \{t\} \qquad (4.2)$$

$$\sum_{\{i:(i,j)\in A\}} y_{ij} = x_j \qquad \forall j \in V \backslash \{s\} \qquad (4.3)$$

$$\sum_{(i,j)\in A} f_{ij} y_{ij} \le F \qquad (4.4)$$

$$d_i - d_j + f_{ij} \le (1 - y_{ij})F \qquad \forall (i,j) \in A \qquad (4.5)$$

$$E_i x_i \le d_i \le L_i x_i \qquad \forall i \in V \qquad (4.6)$$

$$x_i \in \{0,1\} \qquad \forall i \in V \qquad (4.7)$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad (4.8)$$

$$d_i \ge 0 \qquad \forall i \in V \qquad (4.9)$$

The Objective function (4.1) maximizes the total gathered information value. Constraint sets (4.2) and (4.3) guarantee that the tour is connected and each target is visited at most once. Constraint (4.4) limits the total fuel consumption. Constraint set (4.5) excludes sub tours and defines the departure times. Constraint set (4.6) restricts the end of the recording to the time window. Note that the departure time can be equal to the earliest time, since the fuel consumption f_{ij} includes the fuel usage during the recording of target j. It is therefore necessary to define E_i as the earliest time a target can be left after recording.

The possibility to wait until a target is 'open' is not explicitly modeled in this formulation, but it is implicitly possible by Constraint set (4.5). This constraint set requires that when the flight path from target i to target j is selected in the tour, the difference between the departure times of these targets has to be larger or equal to the fuel consumption on this path. Since this difference could be larger than the fuel used both during the flight from target i to target j and during the recording of target j, it is possible to wait until the reached target is 'open'. The fuel consumption during this waiting time is not included in the total fuel usage calculated in Constraint (4.4). However, this extra fuel usage while waiting at a target, is included in the definition of the departure times of the next targets in the tour and therefore also in the arrival time at the end point t by Constraint set (4.5). By setting the latest time of the end point t equal to the fuel capacity of the vehicle, the total amount of fuel used during flying, waiting and recording should be less than or equal to the fuel capacity F by Constraint set (4.6). Note that it is also possible to wait longer than strictly necessary. Since spending fuel for additional waiting yields nothing, the amount of fuel that is used for waiting will be minimized in the optimal solution.

4.2 Chance-Constrained programming formulation

Due to the stochastic fuel consumption, it is not possible to use the MIP-formulation of the OPTW, presented in Section 4.1, for the SOPTW. To formulate a deterministic mathematical model for stochastic problems such as the SOPTW, stochastic programming is introduced in literature. There are several different approaches in this spectrum, but for routing problems chance-constrained programming is one of the most commonly used methods. As we have seen in our literature review in Chapter 3, Laporte et al. [26] have used CCP to solve the SVRP and Gao [31] and Li et al. [32] for the SVRPTW. Chance-constrained programming is introduced by Charnes and Cooper in [36]. The main idea of this approach is to maximize the objective subject to both deterministic constraints and stochastic constraints which must be satisfied with prescribed levels of probability.

Taking the stochastic fuel consumption into account, Constraint (4.4) and Constraint set (4.6) of (P4.1) will be stochastic constraints and we will therefore model them by a chance constraint. These chance constraints are based on Li et al. [32]. Since Li et al. study the SVRPTW, some adjustments are required to fit the SOPTW that we are addressing.

Considering Constraint (4.4), the fuel usage during the flight will be a variable sum of random numbers. In real-world situations, a UAV has a fixed fuel capacity, which cannot be exceeded. In the corresponding chance constraint, the total fuel consumption may exceed the fuel capacity F with a certain probability. Note that when this threshold is less than 1, adjusting the tour during the flight could be necessary. The chance constraint according to Constraint (4.4) is

$$P\left\{\sum_{(i,j)\in A} f_{ij}y_{ij} \leq F\right\} \geq \alpha \tag{C1^1}$$

where P is the probability measure and α is the threshold. This constraint implies that the total fuel usage during the flight may exceed the capacity of the UAV with a probability less than $1-\alpha$. A lower value of α means that adjusting during the flight is required with a higher probability. Since a chance constraint is based on a linear constraint from the deterministic model, it is reasonable that this threshold is about 0.8 or higher and consequently the probability to meet the corresponding deterministic constraint will be large.

Secondly, we consider the probability of departing from a target outside its time windows. One way to formulate a chance constraint with respect to the time windows is by defining a confidence level β by which all targets are to be recorded within their time windows. A higher value of β indicates that the route has a lower probability of causing additional waiting time or missing time windows. As Chance constraint (C1) this chance constraint is based on a deterministic constraint set and therefore it is reasonable that threshold β is also about 0.8 or higher.
The following chance constraint set is introduced for the time windows of each target:

$$P\{E_i x_i \leq d_i \leq L_i x_i\} \geq \beta \qquad \forall i \in N \qquad \text{(C2)}$$

Due to Constraint set (4.5) of (P4.1), the value of d_i depends on the stochastic fuel consumption both during the flight and during the recording of the targets. Therefore, d_i is a random variable.

Another possibility to formulate a chance constraint set for the time windows is to consider only a hard deadline, i.e. the UAV only has to depart from a target before the latest time with a probability δ. The earliest time of the time window is used to restrict the waiting time. The probability that the UAV arrives before the earliest time and has to wait until the target is 'open' and recording is possible, has to be smaller than $1-\gamma$. The lower value of γ means that the UAV has a smaller probability of waiting. The corresponding Chance constraint set (C3) can be written as

$$\begin{cases} P(d_i \geq E_i x_i) \geq \gamma & \text{(C3.1)} \\ P(d_i \leq L_i x_i) \geq \delta & \text{(C3.2)} \end{cases} \qquad \forall i \in N$$

Note that when both Set (C3.1) and (C3.2) are considered, the probability that the UAV arrives and departs within the time windows, which is modeled by β in (C2), is equal to $\gamma+\delta-1$. The main difference between (C2) and (C3) is that the probabilities of waiting on the opening of a target γ and missing a target δ are fixed in Set (C3) and can vary in Set (C2), since only the sum minus one is fixed by β. To ensure that the probability a target is departed within its time window is sufficiently larger, both γ and δ should be about 0.8 or higher.
When only the second set of chance constraints is taken into account, the problem should called Stochastic Orienteering Problem with Deadlines (SOPD), based on literature [37]. In this problem to each target only a latest time is assigned for which the recording should take place. There is still no literature available about this problem, but the models discussed in this thesis can be relatively easily adapted to the SOPD by removing the earliest time constraint or waiting time constraints.

[1] The numbering of the constraints and equations in this chapter is organized as follows:
- The number of linear constraints starts with the number of the chapter, followed by a serial number
- The number of chance constraints starts with a C, followed by a serial number.
- The number of quadratic constraints starts with a Q, followed by the same serial number as the corresponding chance constraint.
- Equations starting with an R are auxiliary equations for the rewriting of the chance constraints to quadratic constraints. The serial number of these equations begins with the same number as the corresponding chance constraint, followed by its own serial number.

4.3 Quadratically constrained programming reformulation

To rewrite the CCP-formulation of the SOPTW to a quadratically constrained program, an assumption about the probability distribution of the fuel consumption on each arc has to be made. We assume that the fuel usage on the arc between target i and target j is normally distributed, thus $f_{ij} \sim N(\mu_{ij}, \sigma_{ij}^2)$. Furthermore, the fuel consumption on different arcs is assumed to be independently. Since predicted deviations such as weather and wind forecasts are already included in the expected value of f_{ij}, only unforeseen circumstances which occur randomly will affect the fuel consumption. Based on this assumption, the chance constraint sets of previous section can be rewritten to their deterministic equivalent.

4.3.1 Fuel Constraint

Starting at Chance constraint (C1),

$$P\left\{\sum_{(i,j)\in A} f_{ij}y_{ij} \leq F\right\} \geq \alpha$$

with $f_{ij} \sim N(\mu_{ij}, \sigma_{ij}^2)$ and α is the threshold. This can be rewritten as

$$P\{T \leq F\} \geq \alpha \tag{R1.1}$$

where T is the total fuel consumption during the flight, thus $T = \sum_{(i,j)\in A} f_{ij}y_{ij}$. Since the sum of independent normally distributed random numbers is also normally distributed with its mean equal to the sum of the means and its variance equal to the sum of the variances, $T \sim N\left(\sum_{(i,j)\in A}\mu_{ij}y_{ij}, \sum_{(i,j)\in A}\sigma_{ij}^2 y_{ij}\right)$. Based on the characteristics of the normal distribution, (R1.1) is equivalent to

$$P\left\{Z \leq \frac{F-\mu_T}{\sigma_T}\right\} \geq \alpha \tag{R1.2}$$

where Z is a standard normally distributed variable, $\mu_T = \sum_{(i,j)\in A}\mu_{ij}y_{ij}$ and $\sigma_T = \sqrt{\sum_{(i,j)\in A}\sigma_{ij}^2 y_{ij}}$. Given the properties of a standard normally distributed variable, (R1.2) can be rewritten as

$$\frac{F-\mu_T}{\sigma_T} \geq Z(\alpha) \tag{R1.3}$$

where $Z(\alpha)$ is the z-score corresponding to the α-percentile. This value is given when α is known. Note that Equation (R1.3) is therefore deterministic and can also be written as

$$F - \mu_T \geq Z(\alpha)\sigma_T \tag{R1.4}$$

which is equivalent to

$$F - \sum_{(i,j)\in A}\mu_{ij}y_{ij} \geq Z(\alpha)\sqrt{\sum_{(i,j)\in A}\sigma_{ij}^2 y_{ij}} \tag{R1.5}$$

This is a deterministic, but non-linear constraint, since the square root of a variable is a non-linear operation. Equation (R1.5) can be reformulated as a quadratic constraint by squaring both sides. Consequently, Chance constraint (C1) can be rewritten as

$$\left[F - \sum_{(i,j)\in A} \mu_{ij} y_{ij}\right]^2 \geq [Z(\alpha)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 y_{ij} \qquad \text{(Q1)}$$

Note that this last rewriting is only correct when $F - \sum_{(i,j)\in A} \mu_{ij} y_{ij}$ is nonnegative and $\alpha \geq 0.5$. These requirements are similar, since the z-score of the 0.5-percentile is equal to zero. As mentioned before, the requirement $\alpha \geq 0.5$ is a reasonable assumption, because a chance constraint is based on a linear constraint from the deterministic model and therefore the threshold to meet this deterministic constraint will be large.

4.3.2 Deadline and Waiting Constraint

For Chance constraint set (C2) and (C3), approximately the same reasoning can be used to reformulate the deterministic quadratic equivalent. In the following part, we will discuss the quadratic deterministic equivalent of Constraint set (C3). We will start with the second constraint set of Chance constraint set (C3), because reformulating this set is most related to the reformulation of Chance constraint (C1).

Recall Constraint set (C3.2),

$$P(d_i \leq L_i x_i) \geq \delta \qquad \forall i \in N \qquad \text{(R3.2.1)}$$

In Equation (R3.2.1), d_i is the random variable. Due to the assumption that the fuel consumption is normally distributed, d_i is also normally distributed. Note that in this assumption the waiting time of the vehicle is ignored to simplify the problem. This means that the departure time of a target could be before the earliest time of that target and thus recording could take place before the target is 'open'. Consequently, this results in an overestimating of the total gathered profit. In the third variant of the CCP, discussed in the last subsection of this section, the expected waiting times are taken into account by determining the departure times of the targets.

Based on the assumption that d_i is normally distributed, the mean of d_i is equal to the sum of the means of the arcs before target i, while the variance of d_i is equal to the sum of the variances of the arcs before target i. To determine the mean and variance of d_i, some auxiliary variables are introduced. First of all, the integer variables u_i denote the position of target i in the tour. Therefore, the following constraint set is added to the MIQCP, which is based on the IP formulation of the TSP presented by Miller et al. in [38]:

$$u_i - u_j + 1 \leq (1 - y_{ij})|V| \qquad \forall (i,j) \in A \qquad (4.10)$$

where $u_i \in \{1, \dots, |V|\}$. Constraint set (4.10) replaces Constraint set (4.5) in the MIQCP, since both constraints avoid sub tours. Furthermore, Constraint set (4.5) defines the departure times using the fuel usage. In the SOPTW these parameters are stochastic and therefore only probabilistically known.

Secondly, the binary variables $Q_{ki} = 1$, if target i comes before target k in the tour, which means that $u_i < u_k$ and 0 otherwise. This can be done by adding the following constraint set to the MIQCP:

$$\begin{cases} u_k - u_i \leq Q_{ki}|V| \\ u_i - u_k + \varepsilon \leq (1 - Q_{ki})|V| \end{cases} \qquad \forall k, \forall i \in V \qquad (4.11)$$

where $Q_{ki} \in \{0,1\}$.

The final set of introduced auxiliary variables are also binary variables, $H_{kij} = 1$, if arc (i,j) is in the tour before arriving at target k, which means that both $Q_{ki} = 1$ and $y_{ij} = 1$ and 0 otherwise. To implement these variables, the following constraint set is added to the program:

$$\begin{cases} Q_{ki} + y_{ij} - 1 \leq H_{kij} \\ H_{kij} \leq \dfrac{Q_{ki} + y_{ij}}{2} \end{cases} \qquad \forall k \in V, \forall (i,j) \in A \qquad (4.12)$$

where $H_{kij} \in \{0,1\}$. The first constraint set ensures that $H_{kij} = 1$ if both $Q_{ki} = 1$ and $y_{ij} = 1$, while the second restricts the value of H_{kij} to zero otherwise. The second one is also necessary, because in the third variant of the CCP the expected waiting time is minimized. When the vehicle arrives too early at a target, it will be better for the objective to increase one of the H_{kij}-variables to one, instead of collecting some waiting time.

Based on this notation, $d_k \sim N\left(\sum_{(i,j)\in A} \mu_{ij} H_{kij}, \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij}\right)$. Consequently, Equation (R3.2.1) can be rewritten as

$$P\left\{Z \leq \frac{L_k x_k - \mu_{d_k}}{\sigma_{d_k}}\right\} \geq \delta \qquad \forall k \in N \qquad (R3.2.2)$$

where Z is a standard normally distributed variable, $\mu_{d_k} = \sum_{(i,j)\in A} \mu_{ij} H_{kij}$ and $\sigma_{d_k} = \sqrt{\sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij}}$. For the same reasoning as in the reformulation of Chance constraint (C1), the quadratic deterministic equivalent of (R3.2.2) is

$$\left[L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \geq [Z(\delta)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad (Q3.2)$$

Note that this is only correct when $L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij}$ is nonnegative and $\delta \geq 0.5$. As we already mentioned, these assumptions are reasonable.

Using the introduced auxiliary variables, we can also formulate the quadratic deterministic equivalent of the first set of Chance constraint set (C3). The Chance constraint set (C3.1),

$$P(E_i x_i \leq d_i) \geq \gamma \qquad \forall i \in N \qquad (R3.1.1)$$

can be rewritten as

$$P(-d_i \leq -E_i x_i) \geq \gamma \qquad \forall i \in N \qquad (R3.1.2)$$

where $-d_i \sim N\left(-\mu_{d_i}, \sigma_{d_i}^2\right)$. Based on the characteristics of the normal distribution, Equation (R3.1.2) is equivalent to

$$P\left\{Z \le \frac{-E_k x_k + \mu_{d_k}}{\sigma_{d_k}}\right\} \ge \gamma \qquad \forall k \in N \qquad \text{(R3.1.3)}$$

which is similar to Equation (R3.2.2). Thus, the quadratic deterministic equivalent of Chance constraint set (C3.1) is

$$\left[-E_k x_k + \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \ge [Z(\gamma)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad \text{(Q3.1)}$$

Note that also $-E_k x_k + \sum_{(i,j)\in A} \mu_{ij} H_{kij}$ has to be nonnegative and $\gamma \ge 0.5$.

Based on the discussed reformulations of (C1) and (C3) and the introduced auxiliary variables, the MIQCP of the CCP-formulation for the SOPTW is as follows:

$$\text{(P4.3)} \qquad max \sum_{i\in N} p_i x_i \qquad \text{(4.1)}$$

$subject\ to$ Constraints (4.2), (4.3), (4.7), (4.8), (4.10), (4.11) and (4.12)

$$\left[F - \sum_{(i,j)\in A} \mu_{ij} y_{ij}\right]^2 \ge [Z(\alpha)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 y_{ij} \qquad \text{(Q1)}$$

$$F - \sum_{(i,j)\in A} \mu_{ij} y_{ij} \ge 0 \qquad \text{(Q1')}$$

$$\left[-E_k x_k + \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \ge [Z(\gamma)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad \text{(Q3.1)}$$

$$-E_k x_k + \sum_{(i,j)\in A} \mu_{ij} H_{kij} \ge 0 \qquad \forall k \in N \qquad \text{(Q3.1')}$$

$$\left[L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \ge [Z(\delta)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad \text{(Q3.2)}$$

$$L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} \ge 0 \qquad \forall k \in N \qquad \text{(Q3.2')}$$

$$u_i \in \{0, \dots, |V|\} \qquad \forall i \in V \qquad \text{(4.13)}$$

$$q_{ki} \in \{0,1\} \qquad \forall k, \forall i \in V \qquad \text{(4.14)}$$

$$H_{kij} \in \{0,1\} \qquad \forall k \in V, \forall (i,j) \in A \qquad \text{(4.15)}$$

4.3.3 Time Windows Constraint

In this part we will discuss the quadratic deterministic equivalent of Chance constraint set (C2).

Chance constraint set (C2) can be split into two parts. The first part ensures that waiting is not necessary with probability γ_i, while the second part ensures that the target is recorded before the deadline with probability δ_i To combine these parts, an extra constraint set is added. The sum of these probabilities minus 1 has to be larger or equal to the given threshold β.

$$\begin{cases} P(d_i \geq E_i x_i) \geq \gamma_i \\ P(d_i \leq L_i x_i) \geq \delta_i \\ \gamma_i + \delta_i - 1 \geq \beta \end{cases} \qquad \forall i \in N \qquad (C2')$$

Note that the first two constraints are the same as (C3), except that γ and δ are fixed in Set (C3) and in (C2') γ_i and δ_i are decision variables and may vary for each target and only β is fixed. Therefore, we can rewrite

$$P(d_i \geq E_i x_i) \geq \gamma_i \qquad \forall i \in N \qquad (R2.1.1)$$

as

$$\left[-E_k x_k + \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \geq [Z(\gamma_i)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad (R2.1.2)$$

Since γ_i is a decision variable in this case and not a given probability like in the previous part, Equation (R2.1.2) is not linear, due to the non-linear operation of determining a z-score given a variable percentile. To give a linear approximation of this equation, we introduce two new sets, a set of possibilities P and a set of times W. Each possibility p has a given threshold t_p and a given squared z-score z_p. Each target i has two times, an earliest time and a latest time. We also introduce an extra binary decision variable c_{iwp} which is assigned the value 1 in case possibility p is chosen for time w of target i. Only one threshold can be assigned to each time of each target. Consequently, the following constraint has to be added to the MIQCP:

$$\sum_{p\in P} c_{iwp} = 1 \qquad \forall i \in N \qquad (4.16)$$

where $c_{iwp} \in \{0,1\}$.

Constraint (4.17) ensures that the sum of the thresholds of a target minus 1 has to be larger or equal to the probability that the departure time falls within the time window.

$$\sum_{w\in W} \sum_{p\in P} t_p c_{iwp} - 1 \geq \beta \qquad \forall i \in N \qquad (4.17)$$

Based on this notation, the quadratic approximation of Equation (R2.1.2) will be

$$\left[-E_k x_k + \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \geq \sum_{p\in P} z_p c_{k1p} \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad (Q2.1)$$

Furthermore, the second Constraint set of (C2'),

$$P(d_i \leq L_i x_i) \geq \delta_i \qquad \forall i \in N \qquad \text{(R2.2.1)}$$

can be quadratically approximated by

$$\left[L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \geq [Z(\delta_i)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad \text{(R2.2.2)}$$

which is equivalent to

$$\left[L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \geq \sum_{p\in P} z_p c_{k2p} \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad \text{(Q2.2)}$$

Based on the reformulation of (C2) the MIQCP for SOPTW can also be formulated as:

$$\text{(P4.2)} \qquad max \sum_{i\in N} p_i x_i \qquad (4.1)$$

$subject\ to$ Constraints $(4.2), (4.3), (4.7), (4.8), (4.10) - (4.17)$, (Q1) and (Q1′)

$$\left[-E_k x_k + \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \geq \sum_{p\in P} z_p c_{k1p} \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad \text{(Q2.1)}$$

$$-E_k x_k + \sum_{(i,j)\in A} \mu_{ij} H_{kij} \geq 0 \qquad \forall k \in N \qquad \text{(Q2.1′)}$$

$$\left[L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij}\right]^2 \geq \sum_{p\in P} z_p c_{k2p} \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad \text{(Q2.2)}$$

$$L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} \geq 0 \qquad \forall k \in N \qquad \text{(Q2.2′)}$$

$$c_{iwp} \in \{0,1\} \qquad \forall i \in N, \forall w \in W, \forall p \in P \qquad (4.18)$$

4.3.4 Waiting time

As we mentioned before, the expected value of the waiting time at each target is not taken into account in both Formulation (P4.2) and (P4.3) to simplify the problem. This simplification results in an overestimating of the expected total gathered profit. In this third formulation we will discuss a CCP-formulation of the SOPTW where we also consider the expected waiting time of the vehicle during the flight. Note that this results in a decrease in the expected total gathered profit, compared to (P4.2) or (P4.3), but also in a more adequate estimation of the realized total gathered profit.

The nonnegative variable waiting time w_i is the expected waiting time at target i, which is defined in the MIQCP by the following constraint set:

$$w_k \geq E_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} - ww_k \qquad \forall k \in N \qquad (4.19)$$

where $w_k \geq 0$ and ww_k is the sum of the expected waiting times before target k. The relation between these two variables is

$$ww_j = ww_i + w_i \qquad if\ y_{ij} = 1 \qquad \forall (i,j) \in A \qquad (R4.1.1)$$

We can rewrite Equation (R4.1.1) to Constraint set (4.20), which is added to the MIQCP.

$$\begin{cases} ww_j - ww_i - w_i \geq M(1 - y_{ij}) \\ ww_j - ww_i - w_i \leq -M(1 - y_{ij}) \end{cases} \qquad \forall (i,j) \in A \qquad (4.20)$$

where M is a big number.

Using the introduced variables, the expected departure time of target k is equal to $\mu_{d_k} = \sum_{(i,j)\in A} \mu_{ij} H_{kij} + w_k + ww_k$. Consequently, the realized departure time of target k is the sum of the realized fuel consumption before target k and the realized waiting time at and before target k. Note that the realized value of the waiting time is inversely proportional to the realized fuel consumption before target k. However, the waiting time is not normally distributed, since all 'negative values', which is the case as the UAV departs after the earliest departure time, are set equal to zero. Therefore, the distribution of the departure time of target k is the sum of a normal distribution and a one-sided truncated normal distribution which are dependent. Since the shape of the distribution is not very important and only the value of the δ-percentile is relevant, we assume that the departure time of target k is normally distributed with its mean equal to μ_{d_k} as stated above and its variance $\sigma_{d_k}^2$ equal to the variance of the fuel consumption before target k multiplied by a scale parameter λ. This is a reasonable assumption, because of two reasons. First, since the sum of two normally distributed variables is normally distributed and by a one-sided truncated normal distribution one of the tails follows approximately a normal distribution, the sum of a normally distributed variable and a one-sided truncated normally distributed variable is also approximately normally distributed. Secondly, because of the dependency between the two distributions, the covariance is nonzero. The scale parameter λ is added to the variance of the departure times in order to compensate this covariance.

To minimize the error originated to this assumption, the total expected waiting time should be minimized. This can be done by changing the Objective function of the MIQCP (4..1) to

$$max \sum_{i\in V} p_i x_i - \kappa w_i \qquad (4.1')$$

Deadline Constraint

The introduction of the waiting time will also change the deadline Constraint sets (Q3.2) and (Q3.2'). Recall Equation (R3.2.2),

$$P\left\{Z \leq \frac{L_k x_k - \mu_{d_k}}{\sigma_{d_k}}\right\} \geq \delta \qquad \forall k \in N \qquad (R3.2.2)$$

which can be rewritten as

$$L_k x_k - \mu_{d_k} \geq Z(\delta)\sigma_{d_k} \qquad \forall k \in N \qquad \text{(R4.2.1)}$$

where $Z(\delta)$ is the z-score corresponding to the δ-percentile. We have assumed that $d_k \sim N(\sum_{(i,j)\in A} \mu_{ij} H_{kij} + w_k + ww_k, \lambda \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij})$. Consequently, Equation (R4.2.1) is equivalent to

$$\left[L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} - w_k - ww_k\right]^2 \geq \lambda [Z(\delta)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij} \qquad \forall k \in N \qquad \text{(Q4.1)}$$

which is only correct when $L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} - w_k - ww_k \geq 0$ and $\delta \geq 0.5$.

A condition of Mixed Integer Quadratically Constrained Programming is the convexity of the program, according to Galli and Letchford [39]. This means that the P_q matrix of Quadratic constraint q, rewritten in the standard form

$$\boldsymbol{a}_q^T \boldsymbol{x} + \boldsymbol{x}^{\boldsymbol{T}} P_q \boldsymbol{x} \leq \boldsymbol{r}_q,$$

should be positive semi-definite in minimization problems and negative semi-definite in maximization problems. A quadratic constraint is also convex when it can be transformed into a second order cone. Unfortunately, Quadratic constraint set (Q4.1) satisfies neither requirements. However, due to the non-negative condition of the waiting time, the combination between Constraint set (Q4.1) and Constraint set (Q4'),

$$L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} - w_k - ww_k \geq 0 \qquad \forall k \in N \qquad \text{(Q4')}$$

is convex.

To solve the MIQCP, it is necessary to rewrite Constraint set (Q4.1) to a convex equivalent. Therefore we introduce two auxiliary binary variables, wt_{kfg} and wwt_{kfg}, such that

$$\sum_{(f,g)\in F} c_{fg} wt_{kfg} = w_k \qquad \forall k \in N \qquad (4.21)$$

and

$$\sum_{(f,g)\in F} c_{fg} wwt_{kfg} = ww_k \qquad \forall k \in N \qquad (4.22)$$

where $wt_{kfg}, wwt_{kfg} \in \{0,1\}$ and parameter c_{fg} is a trivial matrix with the values in each column is equal to $1 * 10^{-(g-1)}$.

Since a sum of binary variables is restricted, the convex equivalent of Constraint set (Q4.1) is

$$\left[L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} - \sum_{(f,g)\in F} c_{fg}(wt_{kfg} + wwt_{kfg})\right]^2 \geq \lambda [Z(\delta)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij}$$

$$\forall k \in N \qquad \text{(Q4.2)}$$

Fuel Constraint

Also the fuel consumption during the waiting time should be taken into account. The expected total fuel consumption during the flight is increased by the total expected waiting time to $\sum_{(i,j)\in A} \mu_{ij} y_{ij} + \sum_{i\in V} w_i$. The distribution of the total fuel consumption T is the sum of normal distributions and truncated normal distributions, which are dependent. Like for the distribution of the departure time, we assume a normal distribution with a mean equal to the expected total fuel consumption and a variance equal to the variance of the fuel consumption during the flight time and the recording time multiplied by a scale parameter θ. Combining Equation (R1.4) and the assumption $T \sim N\left(\sum_{(i,j)\in A} \mu_{ij} y_{ij} + \sum_{i\in V} w_i , \theta \sum_{(i,j)\in A} \sigma_{ij}^2 y_{ij}\right)$, we can rewrite (Q1) to

$$\left[F - \sum_{(i,j)\in A} \mu_{ij} y_{ij} - \sum_{i\in V} w_i\right]^2 \geq \theta[Z(\alpha)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 y_{ij} \tag{Q1.1}$$

under the condition that $F - \sum_{(i,j)\in A} \mu_{ij} y_{ij} - \sum_{i\in V} w_i \geq 0$ and $\alpha \geq 0.5$.

Like the Constraint set (Q4.1), Constraint (Q1.1) is not convex in itself. Due to the non-negativity of the waiting times, the combination of Constraint (Q1.1) and Constraint (Q1.1'),

$$F - \sum_{(i,j)\in A} \mu_{ij} y_{ij} - \sum_{i\in V} w_i \geq 0 \tag{Q1.1'}$$

is convex.

We introduce another auxiliary binary variable twt_{fg} to rewrite Constraint (Q1.1) to its convex equivalent, such that

$$\sum_{(f,g)\in F} c_{fg} twt_{fg} = \sum_{i\in V} w_i \tag{4.23}$$

where $twt_{fg} \in \{0,1\}$.

Consequently, the convex equivalent of Quadratic constraint (Q1.1) is equal to

$$\left[F - \sum_{(i,j)\in A} \mu_{ij} y_{ij} - \sum_{(f,g)\in F} c_{fg} twt_{fg}\right]^2 \geq \theta[Z(\alpha)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 y_{ij} \tag{Q1.2}$$

The total MIQCP including the waiting time is as follows:

$$(\text{P4.4}) \qquad max \sum_{i\in V} p_i x_i - \kappa w_i \tag{4.1'}$$

$subject\ to$ Constraints $(4.2), (4.3), (4.7), (4.8), (4.10) - (4.15), (4.19) - (4.23)$

$$\left[F - \sum_{(i,j)\in A} \mu_{ij} y_{ij} - \sum_{(f,g)\in F} c_{fg} twt_{fg}\right]^2 \geq \theta[Z(\alpha)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 y_{ij} \tag{Q1.2}$$

$$F - \sum_{(i,j)\in A} \mu_{ij} y_{ij} - \sum_{i\in V} w_i \geq 0 \qquad \text{(Q1.1')}$$

$$\left[L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} - \sum_{(f,g)\in F} c_{fg}(wt_{kfg} + wwt_{kfg})\right]^2 \geq \lambda[Z(\delta)]^2 \sum_{(i,j)\in A} \sigma_{ij}^2 H_{kij}$$

$$\forall k \in N \qquad \text{(Q4.2)}$$

$$L_k x_k - \sum_{(i,j)\in A} \mu_{ij} H_{kij} - w_k - ww_k \geq 0 \qquad \forall k \in N \qquad \text{(Q4')}$$

$$wt_{kfg} \in \{0,1\} \qquad \forall k \in V, \forall (f,g) \in F \qquad (4.24)$$

$$wwt_{kfg} \in \{0,1\} \qquad \forall k \in V, \forall (f,g) \in F \qquad (4.25)$$

$$twt_{fg} \in \{0,1\} \qquad \forall k \in V, \forall (f,g) \in F \qquad (4.26)$$

5. Two-Stage Stochastic Programming with Recourse

In this chapter we will discuss the application of another widely used approach in stochastic programming. Two variants of a two-stage stochastic programming problem for the SOPTW will be presented.

5.1 Stochastic Programming with Recourse

Stochastic Programming with Recourse (SPR) is an approach of stochastic programming which is introduced by Dantzig in [40]. For routing problems it is for example applied by Laporte et al. [26] to the SVRP, by Li et al. [32] to the SVRPTW and by Evers et al. [14] to the SOP. In SPR the problem is modeled in multiple stages. At each stage more realizations of the stochastic parameters are observed. Furthermore, at each stage decisions should be made, such that the expected objective of the current and subsequent stages is maximized or minimized. Most of the SPR-formulations for routing problems make use of two-stage stochastic programming. In the first stage an a priori route is constructed. After the values of the stochastic travel times are realized between the first and the second stage, the costs of final route is determined by incurring a penalty for time overruns or loss of missed targets.

In this chapter we will introduce two different stochastic programming models with recourse for the SOPTW. The first model is based on the TSOP, introduced by Evers et al. [14] for the SOP without time windows. We present the Two-stage Orienteering Problem with Time Windows (TSOPTW) to solve the SOPTW. In this model before the flight an initial tour is constructed based on the probability distributions of the fuel usages on the arcs. We assume that the fuel realization on arc (i, j) is observed after both flying from target i to target j and the recording of target j. For the arcs that have not yet been traversed only the distribution function of the fuel usage is known. In the second stage of this model the next target of the final route is determined by applying a so-called recourse action. We assume that the time windows are hard in this model, which means that recording outside a time window yields nothing. Note that the initially constructed tour in combination with a recourse action is a solution for both the offline and the online part of the UAV-MPP.

In the second model instead of applying a recourse action, we use a penalty function for a late departure from a target after its deadline, like is done by Li et al. [32] and Russell and Urban [34]. We introduce different penalty functions for lateness. This Penalized Orienteering Problem with Time Windows (POPTW) is also modeled in two stages. The expected execution costs of the tour, constructed in the first stage, are considered in the second stage. In this model we assume that all fuel usages are realized at the same time between the first and the second stage, since the first stage tour cannot be adjusted in the second stage. Notice that in this model the information value of a target without its time window is not by definition equal to zero, which means that the time windows are assumed to be soft in this case.

5.2 Two-stage Orienteering Problem with Time Windows

In this section we will formulate the Two-stage Orienteering Problem with Time Windows (TSOPTW), which consists of two stages. During the first stage an initial tour is constructed and when the fuel realizations are observed one by one in the second stage, the final route is determined. The order of the targets in this final route should be the same as in the initial tour, but it is allowed to skip targets because of their time window or to return to the depot earlier than planned due to the fuel level of the UAV.

5.2.1 First stage

In this first stage an initial tour is constructed, which maximizes the expected profit gathered by the executing of this tour. By the construction of the initial tour, time windows and the total fuel capacity are not imposed explicitly, but considered in the objective of this stage.

To formulate the first stage of the TSOPTW, all sets and parameters introduced in Chapter 2 and the binary decision variables, x_i and y_{ij}, introduced in Section 4.1 are used. Also the integer variables u_i which denote the position of target i in the initial tour are used. The stochastic variable $\tilde{f}$ represents the stochastic fuel usages. The MIP formulation of the first stage of the TSOPTW (P5.1) is the following:

$$\text{(P5.1)} \qquad max_y \, \mathbb{E}_{\tilde{f}}\left[h\left(y,\tilde{f}\right)\right] \tag{5.2.1}$$

$$subject\ to \sum_{\{j:(i,j)\in A\}} y_{ij} = x_i \qquad \forall i \in V\backslash\{t\} \tag{5.2.2}$$

$$\sum_{\{i:(i,j)\in A\}} y_{ij} = x_j \qquad \forall j \in V\backslash\{s\} \tag{5.2.3}$$

$$u_i - u_j + 1 \leq (1 - y_{ij})|V| \qquad \forall (i,j) \in A \tag{5.2.4}$$

$$u_i - u_j + 1 \geq -(1 - y_{ij})|V| \qquad \forall (i,j) \in A \tag{5.2.5}$$

$$x_i \in \{0,1\} \qquad \forall i \in N \tag{5.2.6}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{5.2.7}$$

$$u_i \in \{1, \dots, |V|\} \qquad \forall i \in N \tag{5.2.8}$$

The Objective function (5.2.1) maximizes the expected profit given the probability distribution of the fuel realizations gathered in the second stage. Constraint sets (5.2.2) and (5.2.3) guarantee that the initial tour is connected and each target is visited at most once. Constraint sets (5.2.4) and (5.2.5) define the order of the tour and avoid sub tours.

Since the possibility exists to skip targets from the initial tour in the final route, the initial tour can include all targets. The targets which will not be included in the first stage when Program (P5.1) is used, can be added at the end of the tour. These targets will be visited in none of the final routes, otherwise it would be optimal for Program (P5.1) to include them in the initial tour. Note that in this case the first stage of the TSOPTW corresponds to a travelling salesman problem with a different objective function. Program (P5.1) has the objective to maximize the expected gathered profit in the second stage instead of the normal objective of the TSP to minimize the total distance. Program (P5.2) is the new Mixed Integer Program of the first stage of the TSOPTW.

(P5.2) $$max_y \, \mathbb{E}_{\tilde{f}} \left[h\left(y, \tilde{f}\right)\right] \tag{5.2.1}$$

$subject\ to$ Constraints (5.2.4), (5.2.7) and (5.2.8)

$$\sum_{\{j:(i,j)\in A\}} y_{ij} = 1 \qquad \forall i \in V \backslash \{t\} \tag{5.2.9}$$

$$\sum_{\{i:(i,j)\in A\}} y_{ij} = 1 \qquad \forall j \in V \backslash \{s\} \tag{5.2.10}$$

The Objective function (5.2.1) and the Constraint sets (5.2.4), (5.2.7) and (5.2.8) are the same as in Program (P5.1). Constraint sets (5.2.9) and (5.2.10) guarantee that the initial tour is connected and each target is visited. Notice that in Program (P5.1) both Constraint set (5.2.4) as Constraint set (5.2.5) are necessary to define the order of the tour correctly and therefore also the positions of the targets in the tour indicated by the u_i-variables. This is due to the characteristic of the OP that not all targets have to be visited. Since the order of the tour should be used in the second stage, it is necessary that the values of the u_i-variables are correct, which means that the positions of two consecutive targets differ only one from each other. Otherwise, deviations in the order of the final route with respect to the order of the initial tour are possible. In Program (P5.2) all targets have to be visited and therefore only Constraint set (5.2.4) is necessary to define the u_i-variables correctly.

The main advantage of Program (P5.2) over Program (P5.1) is that the calculation time of (P5.2) is much shorter than that of (P5.1). The reduction in the calculating time is the consequence of the smaller number of decisions that should be made in the program. In Program (P5.1) both the included targets and the order of the included targets have to be determined. On the other hand in program (P5.2) only the order of the targets should be decided.

5.2.2 Second stage

In the second stage of the TSOPTW the final route is constructed step by step. Based on the initial tour, the already observed fuel realizations and the recourse action at each location the next target of the flight is determined. In the TSOPTW the following recourse action is applied: the next target in the final route is equal to the next target in the initial tour, except when the probability that this target can be reached before its deadline is below a predefined level α or when the remaining fuel quantity is insufficient to fly to this target and back to the depot in the worst case. In that case the next target of the initial tour is skipped in the final route. Note that the next target in the final route is the first target in the initial tour which satisfies the recourse requirements, starting at the current location. This means that this target can be reached before its deadline with a probability of at least α and when the remaining fuel quantity is sufficient to fly to this target and back to the depot in the worst case.

In order to determine whether the recourse requirements are satisfied for the next target in the initial tour, two limit values could be calculated. The first limit value is equal to sum of the fuel consumption up to the current location and the α-percentile of the fuel usage on the flight path between the current and the next location. If this value is smaller than or equal to the deadline of the next location the first recourse requirement is satisfied. The second limit value is equal to the sum of the fuel consumption up to the current location and the worst case realizations from the current to the next location and from the next location to the depot. To meet the second recourse requirement this value should be smaller or equal to the fuel capacity of the UAV.

Deviations in these limit values could have both negative and positive effects on the total gathered profit. It is reasonable that increasing a limit value could have a negative effect. This is the case when the next target should be skipped, because the increased limit value is larger than its deadline or the fuel capacity. It is also reasonable that decreasing a limit value could have a positive effect. When the decreased limit value is smaller than the deadline of next target or than the fuel capacity, a target which should be skipped, is recorded in the final route. If this recording takes place before the deadline, the total gathered profit is increased.

However, also an increase of a limit value could have a positive effect on the total gathered profit, while a decrease could also have a negative effect. The total gathered profit is decreased by a decrease of the limit value, if this results in the recording of the next location and causing that another, more valuable, target of the initial tour should be skipped. Moreover, an increase of a limit value results in an increase of the total gathered profit if this makes that the next target should be skipped, while a more valuable target later in the initial tour can be included in the final route.

Note that this reasoning is also valid for deviations in other decision variables such as the sum of the realized fuel usages and the waiting time.

For each scenario $\omega \in \Omega$ of the fuel consumption the final route is determined using a Mixed Integer Program (MIP). Given the initial tour, the recourse action and the fuel consumption in the given scenario, the final route is fixed, but to optimize the initial tour of the first stage the determination of the final route is also modeled as a MIP. In this second stage the initial tour and the realized fuel consumptions $f_{\omega ij}$ are given. The decision variables of the first stage are used to describe the initial tour. Since the initial tour cannot be changed during the second stage, both the binary variables x_i and y_{ij} and the integer variables u_i are parameters in the MIP of this stage (P5.3).

In the remainder of this section we will introduce a mixed integer programming model for the second stage of the TSOPTW. Because of the size of the model, we will discuss its constraints in parts.

First of all, in the final route not all targets which are included in the initial tour have to be visited. Therefore, we introduce the binary decision variables $b_{\omega i}$, for which holds that $b_{\omega i} = 1$ if target i is visited in the final route of scenario ω and 0 otherwise. Not only the selected targets are important, but also the selected flight paths. Hence, we define the binary variables $t_{\omega ij}$, if the flight path from target i to target j is in the final route of scenario ω, $t_{\omega ij} = 1$ and 0 otherwise. Like the initial tour, the final route has to be connected and each selected target should be visited once. Consequently, Constraint sets (5.2.11) and (5.2.12) are the first two sets of constraints in Program (P5.3).

$$\sum_{\{j:(i,j)\in A\}} t_{\omega ij} = b_{\omega i} \qquad \forall \omega \in \Omega, \forall i \in V\backslash\{t\} \qquad (5.2.11)$$

$$\sum_{\{i:(i,j)\in A\}} t_{\omega ij} = b_{\omega j} \qquad \forall \omega \in \Omega, \forall j \in V\backslash\{s\} \qquad (5.2.12)$$

where $t_{\omega ij}, b_{\omega i} \in \{0,1\}$.

If target i is included in the initial tour but skipped in the final route of scenario ω, we set binary variable $c_{\omega i} = 1$ and 0 otherwise. Based on this definition, in each scenario for each target which is included in the initial tour, either $b_{\omega i}$ or $c_{\omega i}$ could be equal to one, which is guaranteed by

$$b_{\omega i} + c_{\omega i} = x_i \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.2.13)$$

where $c_{\omega i} \in \{0,1\}$. Note that this constraint set can only be used when in the first stage the decision variables x_i are defined as described in Program (P5.1). When Program (P5.2) is used, Constraint set (5.2.13) should be

$$b_{\omega i} + c_{\omega i} = 1 \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.2.13')$$

where $c_{\omega i} \in \{0,1\}$.

Since the order of visited targets in the final route should the same as in the initial tour, the following constraint is added to Program (P5.3):

$$u_i - u_j + 1 \le \left(1 - t_{\omega ij}\right)|V| \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.14)$$

To determine the departure time of the vehicle from target i in scenario ω, we define the auxiliary nonnegative decision variable $ff_{\omega i}$. The value of this variable is the realized fuel consumption up to and including the recording of target i in scenario ω. Constraint set (5.2.15) and (5.2.16) together ensure that the value of $ff_{\omega i}$ is equal to the realizations of fuel usages in scenario ω on the arcs before target i.

$$ff_{\omega i} - ff_{\omega j} + f_{\omega ij} \le \left(1 - t_{\omega ij}\right)F \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.15)$$

$$ff_{\omega i} - ff_{\omega j} + f_{\omega ij} \ge -\left(1 - t_{\omega ij}\right)F \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.16)$$

where $ff_{\omega i} \ge 0$. As we discussed earlier in this subsection, both constraints are necessary since deviations could increase the total gathered profit. Remark that in the combination of these two constraint sets, the fuel consumption during the time waiting before a target can be recorded is not included. Therefore, we introduce the nonnegative decision variable $w_{\omega i}$ which represents the realized waiting time at target i in scenario ω. To define the value of these variables, the following constraint set is included in Program (P5.3):

$$w_{\omega i} \ge E_i - ff_{\omega i} - ww_{\omega i} \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.2.17)$$

where $w_{\omega i} \ge 0$ and $ww_{\omega i}$ is the sum of the realized waiting times before target i in scenario ω. The relation between these two variables is defined by

$$\begin{cases} ww_{\omega j} - ww_{\omega i} - w_{\omega i} \le \left(1 - t_{\omega ij}\right)F \\ ww_{\omega j} - ww_{\omega i} - w_{\omega i} \ge -\left(1 - t_{\omega ij}\right)F \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.18)$$

where $ww_{\omega i} \ge 0$. Also in this case both constraints are necessary for the reasoning given earlier in this subsection.

The realized departure time from target i in scenario ω is equal to the sum of the realized fuel usage during the flight up to target i, including the realized waiting times before and at target i. The definition of the realized departure time $d_{\omega i}$ is

$$d_{\omega i} = ff_{\omega i} + w_{\omega i} + ww_{\omega i} \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.2.19)$$

where $d_{\omega i} \geq 0$.

When the departure time of a target falls outside of the given time window of that particular target, the gathered profit at that target is zero, because we have assumed that the time windows are hard in our case. The binary decision variable $h_{\omega i} = 1$, if the departure time from target i in scenario ω is before the latest time and the target is visited and 0 otherwise. The following constraint set is included in the second stage program to define $h_{\omega i}$:

$$\begin{cases} d_{\omega i} - L_i \leq (1 - h_{\omega i})F \\ \qquad h_{\omega i} \leq b_{\omega i} \end{cases} \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.2.20)$$

where $h_{\omega i} \in \{0,1\}$.

To implement the recourse action, we introduce several auxiliary decision variables. First of all, the nonnegative variable $g_{\omega ij}$ represents the α-percentile of the departure time from target j in scenario ω when arc (i,j) is used. This means that the departure time of the vehicle from target j coming from target i in scenario ω is smaller than $g_{\omega ij}$ with a probability of α. Consequently,

$$g_{\omega ij} = d_{\omega i} + a_{ij}$$

where $g_{\omega ij} \geq 0$ and a_{ij} is the α-percentile of the arc (i,j). Note that this definition is only true when target i is not skipped, which means that $c_{\omega i} = 0$. Therefore, Constraint set (5.2.21) is added to Program (P5.3).

$$\begin{cases} g_{\omega ij} - d_{\omega i} - a_{ij} \leq c_{\omega i}F \\ g_{\omega ij} - d_{\omega i} - a_{ij} \geq -c_{\omega i}F \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.21)$$

To define this limit value correctly, both constraints are essential.

When a target j is skipped in scenario ω, the values of the $g_{\omega ij}$-variables remain the same, but the next target is different. For instance, suppose that the vehicle is at target i and $y_{ij} = 1$, which means that the next target in the initial tour is target j and according to the recourse action, target j should be skipped. The new next target will be target k for which holds that $y_{jk} = 1$. In order to determine whether target k should also be skipped according to the recourse action, it is necessary to consider target k from target i. For this reason, the values of the $g_{\omega jk}$-variables are the same as the values of the $g_{\omega ik}$-variables for scenario ω if $c_{\omega j} = 1$ and $y_{ij} = 1$. Consequently,

$$\begin{cases} g_{\omega jk} - g_{\omega ik} \leq F\left[\left(1 - c_{\omega j}\right) + \left(1 - y_{ij}\right)\right] \\ g_{\omega jk} - g_{\omega ik} \geq -F\left[\left(1 - c_{\omega j}\right) + \left(1 - y_{ij}\right)\right] \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A, \forall k \in N \quad (5.2.22)$$

Also both these constraints are necessary to prevent deviations in the limit values.

The second auxiliary variable we introduce, is the binary variable $m_{\omega ij}$. If it is possible to use the flight path from target i to target j in scenario ω regarding the latest time of target j, $m_{\omega ij} = 1$ and 0 otherwise. Constraint set (5.2.23) guarantees that $m_{\omega ij} = 1$, if $g_{\omega ij} \leq L_j$ and 0 otherwise.

$$\begin{cases} g_{\omega ij} - L_j \leq (1 - m_{\omega ij})F \\ g_{\omega ij} - L_j \geq -m_{\omega ij}F \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.23)$$

where $m_{\omega ij} \in \{0,1\}$. Since the definition of this variable is only based on satisfying the requirement that the departure time of target j should be earlier than the latest time with a probability of α, another two auxiliary variables are introduced for meeting the second requirement of the recourse action. The value of the nonnegative variable $gd_{\omega ij}$ is the arrival time at the depot in scenario ω when the vehicle returns at the depot immediately after visiting target j coming from target i and the fuel usage on both arcs is equal to the worst case realizations. Given that W_{ij} is the sum of the worst case realizations of arc (i,j) and arc (j,t),

$$gd_{\omega ij} = d_{\omega i} + W_{ij}$$

if target i is not skipped in the final route, which means that $c_{\omega i} = 0$. Consequently,

$$\begin{cases} gd_{\omega ij} - d_{\omega i} - W_{ij} \leq c_{\omega i}F \\ gd_{\omega ij} - d_{\omega i} - W_{ij} \geq -c_{\omega i}F \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.24)$$

where $g_{d\omega ij} \in \{0,1\}$. As mentioned before, both increasing and decreasing the limit values could have a positive effect on the total gathered profit. Therefore, also these both constraints are essential to define the limit values correctly.

For the same reasoning as previously mentioned,

$$gd_{\omega jk} = gd_{\omega ik}$$

if $c_{\omega j} = 1$ and $y_{ij} = 1$. Hence,

$$\begin{cases} gd_{\omega jk} - gd_{\omega ik} \leq F\left[\left(1 - c_{\omega j}\right) + \left(1 - y_{ij}\right)\right] \\ gd_{\omega jk} - gd_{\omega ik} \geq -F\left[\left(1 - c_{\omega j}\right) + \left(1 - y_{ij}\right)\right] \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A, \forall k \in N \quad (5.2.25)$$

To indicate whether it is possible to use the flight path from i to j while satisfying the requirement that the total fuel consumption in scenario ω returned at the depot after visiting target j coming from target i under worst case circumstances should be smaller than the fuel capacity of the vehicle, we introduce the binary variable $md_{\omega ij}$. If the mentioned requirement is met, $md_{\omega ij} = 1$ and 0 otherwise, which is ensured by Constraint set (5.2.26).

$$\begin{cases} gd_{\omega ij} - F \leq (1 - md_{\omega ij})F \\ gd_{\omega ij} - F \geq -md_{\omega ij}F \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.26)$$

where $md_{\omega ij} \in \{0,1\}$. In order to prevent deviations in the limit values, both constraints should be added to the MIP.

The last auxiliary variable we introduce for Program (P5.3), is the binary variable $mt_{\omega ij}$. If both requirements of the recourse action are satisfied for the combination of target i and j in scenario ω, which means that both $m_{\omega ij}$ as $md_{\omega ij}$ are equal to 1, $mt_{\omega ij} = 1$ and 0 otherwise. Therefore, Constraint set (5.2.27) is added to Program (P5.3).

$$\begin{cases} m_{\omega ij} + md_{\omega ij} - 1 \leq mt_{\omega ij} \\ \dfrac{m_{\omega ij} + md_{\omega ij}}{2} \geq mt_{\omega ij} \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.27)$$

where $mt_{\omega ij} \in \{0,1\}$.

According to the recourse action, a target from the initial tour could only be skipped in the case that at least one of the requirements is not satisfied. Based on the definitions of the introduced auxiliary variables in this section, a target should be skipped in scenario ω when $mt_{\omega ij} = 0$ and $y_{ij} = 1$. This means that in the initial tour the next target is target j, but it is not allowed to use the flight path from target i to target j in scenario ω and therefore target j should be skipped in the final route. This last constraint set of Program (P5.3) guarantees that a target is only skipped, when the requirements are not met.

$$\begin{cases} y_{ij} + \left(1 - mt_{\omega ij}\right) - 1 \leq c_{\omega j} + \left(1 - y_{ij}\right) \\ \dfrac{y_{ij} + \left(1 - mt_{\omega ij}\right)}{2} \geq c_{\omega j} - \left(1 - y_{ij}\right) \end{cases} \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.28)$$

Note that the addition of $\pm\left(1 - y_{ij}\right)$ is necessary, since the decision about skipping target j should not be affected by a flight path from target i to target j which is not included in the initial tour.

5.2.3 Objective function

In the previous section we have discussed the constraints of Program (P5.3). In this section we will discuss the objective function of the second stage program, Program (5.3).

First, we will investigate the profit of the final route. When a target is visited within its time window and the vehicle has departed before the latest time, the profit of the recording of that target is equal to the parameter p_i, which we have introduced in Chapter 2. The binary variable $h_{\omega i} = 1$, if the departure time of the vehicle from target i in scenario ω is before the latest time, which means that the target is visited within its time window, and 0 otherwise. Based on this definition, the information is gathered from target i in scenario ω if $h_{\omega i} = 1$. Consequently, the total profit gathered in the second stage in scenario ω is equal to

$$\sum_{i \in N} p_i h_{\omega i}$$

The objective of the second stage is not only to maximize the total gathered profit in scenario ω, but also to minimize the waiting time. This is necessary, because additional waiting time could have a positive effect on the profit value in the execution of the recourse action. If the vehicle arrives before the earliest time of target i in scenario ω, waiting time occurs. This means that the nonnegative variable $w_{\omega i}$ will be strictly positive. Since the all $w_{\omega i}$-variables should be larger or equal to 0, the total waiting time, incurred during the final route of scenario ω, is

$$\sum_{i \in N} \kappa w_{\omega i}$$

where κ is a scale parameter for the waiting time.

For this reason, the objective function of Program (P5.3) is

$$h(y, f_\omega) = max \sum_{i \in N} p_i h_{\omega i} - \sum_{i \in N} \kappa w_{\omega i}$$

5.2.4 Summary

The TSOPTW is a large mixed integer program with the objective of finding an initial tour resulting in the maximum total estimated expected profit, based on the predefined recourse action that will be applied during the actual flight. For each scenario $\omega \in \Omega$, the gathered profit minus the waiting time is determined based on the first stage initial tour and the recourse action. The complete formulation of the TSOPTW (P5.3) is as follows:

$$\text{(P5.3)} \qquad max_y \, \mathbb{E}_{\tilde{f}} \left[h(y, \tilde{f}) \right] \qquad (5.2.1)$$

$subject\ to$ Constraints $(5.2.4), (5.2.7) - (5.2.12), (5.2.13'), (5.2.14) - (5.2.28)$

$$b_{\omega i}, c_{\omega i}, h_{\omega i} \in \{0,1\} \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.2.30)$$

$$t_{\omega ij} \in \{0,1\} \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.31)$$

$$ff_{\omega i}, w_{\omega i}, ww_{\omega i}, d_{\omega i} \geq 0 \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.2.32)$$

$$g_{\omega ij}, gd_{\omega ij}, m_{\omega ij}, md_{\omega ij}, mt_{\omega ij} \geq 0 \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.2.33)$$

5.3 Penalty Method for Stochastic Programming with Recourse

In this section we will introduce another two-stage stochastic programming model with recourse for the SOPTW, the Penalized Orienteering Problem with Time Windows (POPTW). In this model a penalty is incurred when a target is visited after its latest time. During the first stage a tour is constructed, such that the profit is maximized and the expected incurred penalty in the second stage is minimized. We will introduce four different types of penalty functions. The first two types of penalty functions are developed using a fixed cost which means that the penalty is independent of the length of the delay, while the other two types are smooth penalty functions, where the penalty increases as the length of the delay becomes longer.

5.3.1 First stage

During the first stage a tour is constructed, of which cannot be derogated in the second stage. This means that all included targets have to be visited in the second stage. In the first stage the time windows of the targets and the fuel capacity of the vehicle are not taken into account. However, lateness and running out of fuel during the execution of the tour does not affect the feasibility, since the time windows are assumed to be soft, but a penalized cost will be subtracted from the objective function value. Also sub tours are not excluded during the first stage of the POPTW. However, since solutions which contain sub tours are not feasible in the second stage, sub tours are implicitly excluded in the first stage.

In the Mixed Integer Programming formulation of the first stage we make use of the already mentioned binary variables, x_i and y_{ij}. The variable $x_i = 1$, if target i is included in the tour and 0 otherwise and the variable $y_{ij} = 1$, if the flight path from target i to target j is selected in the tour. The MIP formulation of the first stage (P5.4) is the following:

$$\text{(P5.4)} \qquad max_y \sum_{i \in N} p_i x_i - \mathbb{E}_{\tilde{f}}\left[c\left(y, \tilde{f}\right)\right] \qquad (5.3.1)$$

$$subject\ to \sum_{\{j:(i,j)\in A\}} y_{ij} = x_i \qquad \forall i \in V \backslash \{t\} \qquad (5.3.2)$$

$$\sum_{\{i:(i,j)\in A\}} y_{ij} = x_j \qquad \forall j \in V \backslash \{s\} \qquad (5.3.3)$$

$$x_i \in \{0,1\} \qquad \forall i \in N \qquad (5.3.4)$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad (5.3.5)$$

The objective of Program (P5.4) is to maximize the total gathered profit, of which the expected incurred penalty is subtracted. Constraint sets (5.3.2) and (5.3.3) guarantee that the tour is connected and each target is visited at most once.

5.3.2 Second stage

At the beginning of the second stage we assume that all fuel realizations are available. Based on these realizations the incurred penalty costs of the first stage tour are determined for each scenario $\omega \in \Omega$.

Since deviations from the constructed tour are not allowed, the first stage decision variables x_i and

y_{ij} are parameters in this stage. Given these parameters and the fuel realizations of scenario $\omega, f_{\omega ij}$, the departure times from the targets in scenario ω are defined using by the following constraint:

$$d_{\omega i} - d_{\omega j} + f_{\omega ij} \leq F(1 - y_{ij}) \qquad \forall \omega \in \Omega, \forall (i,j) \in A \qquad (5.3.4)$$

where the decision variable $d_{\omega i}$ is the departure time from target i in scenario ω and $d_{\omega i} \geq 0$. This constraint ensures that the departure time of two consecutive targets in the tour differs at least the fuel usage on the arc (i,j) from each other and that sub tours are excluded from the tour. In this program waiting time is not explicitly modeled, but Constraint (5.3.5),

$$d_{\omega i} \geq E_i x_i \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.3.5)$$

ensures that the departure time from target i in each scenario is after the earliest time. Note that the waiting time resulting from Constraint (5.3.5) is both authorized and included in the departure times of the rest of the tour by Constraint (5.3.4).

To determine whether a penalty should be incurred at target i in scenario ω, which means that the departure of the vehicle from target i is later than its deadline, we introduce the binary variable $q_{\omega i}$. If the departure time of target i in scenario ω, $d_{\omega i}$ is larger than the latest time of target i, L_i, $q_{\omega i} = 1$ and 0 otherwise, which is guaranteed by

$$d_{\omega i} - L_i \leq q_{\omega i} F \qquad \forall \omega \in \Omega, \forall i \in V \backslash \{s\} \qquad (5.3.6)$$

where $q_{\omega i} \in \{0,1\}$.

5.3.3 Penalty Function

In this subsection we will introduce four different types of penalty functions and we will discuss their implementation.

First of all, we should notice that the different penalty functions affect the penalty for lateness at a target. The penalty for running out of fuel in scenario ω is always the same and is equal to

$$P = \sum_{i \in N} p_i$$

which means that all information value gathered in that scenario is inherently offset and the yield of that scenario is less or equal to zero. This penalty is incurred for scenario ω when the arrival time at the end point is later than its latest time L_t, which is equal to the fuel capacity F. In that case, $d_{\omega t} \geq L_t$ and consequently, $q_{\omega t} = 1$.

The first two types of penalty functions are developed using a fixed cost for lateness at the targets. In the first type, the costs of lateness at a target are the same for each target. This means that when the departure time of a target is later than its deadline, a penalty pf is incurred. When the value of this penalty is low compared to the profits of the targets, it is likely to have more late targets. Note that the yield of a late target which is recorded after the deadline, can be either positive or negative, like is displayed in Figure 5.1.

The penalty cost function for the first stage tour $c(y, f_\omega)$ is in this first case

$$c(y, f_\omega) = pf \sum_{i \in N} q_{\omega i} + q_{\omega t} P \qquad \forall \omega \in \Omega \qquad (5.3.7)$$

The penalty function of the second type is target dependent. In this case the penalty which is incurred by a late target, can be different for each target. We assume that this penalty pf_i is a function of the profit of target i. The information value gathered by recording a target after its deadline, is only a part of the profit gathered by recording within the time window. This part, a, is fixed for each target. Consequently, $pf_i = a \cdot p_i$, where $0 \leq a \leq 1$. Note that when $a = 1$, the yield of recording a late target is equal to zero and the time windows are hard in this case. However, when $a = 0$, there is no penalty incurred for recording a target after its deadline. In Figure 5.2 the second type of penalty function and the consequences of this type for the yield of target i are displayed.

With this second type of penalty function the penalty cost function for the first stage tour is

$$c(y, f_\omega) = \sum_{i \in N} pf_i q_{\omega i} + q_{\omega t} P \qquad \forall \omega \in \Omega \qquad (5.3.8)$$

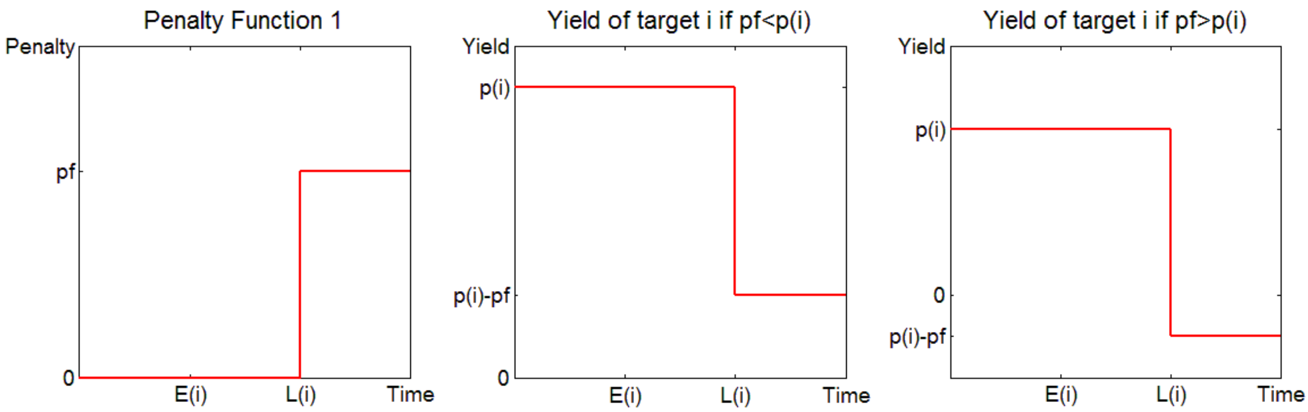

Figure 5.1: The first type of penalty function and its consequences for the yield of target i

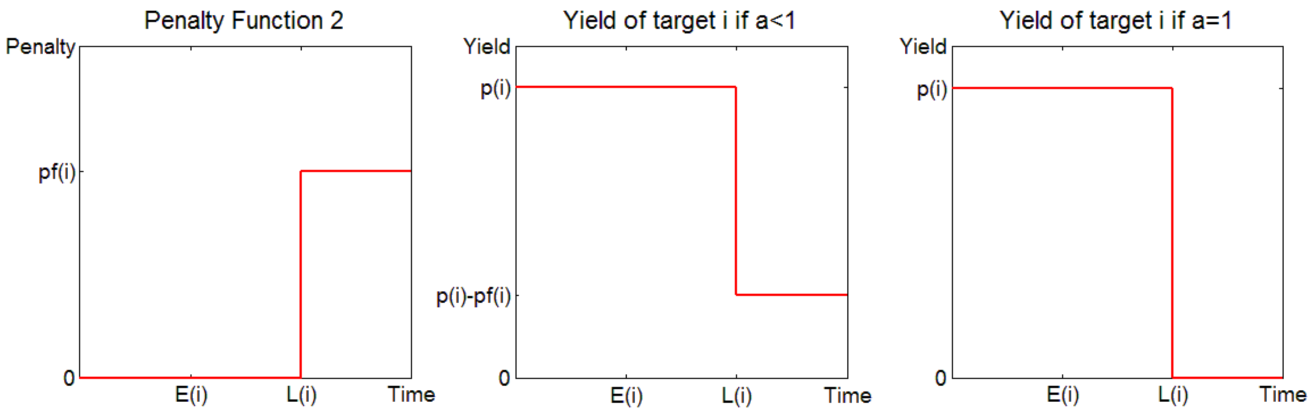

Figure 5.2: The second type of penalty function and its consequences for the yield of target i

The two other types of penalty functions are smooth penalty functions, which means that the incurred penalty increases when the late period becomes longer. In the first smooth penalty function the costs for lateness increases linearly to the late time. In this case the penalty incurred for a late departure from target i is a decision variable instead of a parameter, as it is the case in the cost functions of the first two types. The nonnegative variable $pf_{\omega i}$ represents the incurred penalty for lateness at target i in scenario ω and is defined by Constraint (5.3.9).

$$pf_{\omega i} \geq b(d_{\omega i} - L_i) \qquad \forall\omega \in \Omega, \forall i \in N \qquad (5.3.9)$$

where $pf_{\omega i} \geq 0$ and parameter b is the scale parameter of the penalty function.

The main advantage of a smooth penalty function is that not only the number of late targets is minimized, but also the length of the delay at the late targets. When the value of the parameter b is large, the total delay will be small. In Figure 5.3 the linear penalty function and the yield of target i are displayed.

The penalty cost function in the case of a linear penalty function is equal to

$$c(y, f_\omega) = \sum_{i \in N} pf_{\omega i} + q_{\omega t}P \qquad \forall\omega \in \Omega \qquad (5.3.10)$$

The last type of penalty function is the quadratic penalty function. This means that the cost for lateness increases quadratically with the length of the delay. The penalty incurred at target i in scenario ω, $pf_{\omega i}$ is defined with a quadratic constraint:

$$pf_{\omega i} \geq c(d_{\omega i} - L_i)^2 \qquad \forall\omega \in \Omega, \forall i \in N \qquad (5.3.11)$$

where parameter c is the scale parameter of the penalty function. Based on the definition mentioned earlier, $q_{\omega i} = 1$ if the vehicle departs too late from target i in scenario ω. Consequently, a penalty is only incurred if $q_{\omega i} = 1$, which means that $d_{\omega i} - L_i \geq 0$. Note that when $d_{\omega i} - L_i < 0$, $(d_{\omega i} - L_i)^2 \geq 0$ and a penalty is incorrectly incurred according to Constraint (5.3.11). Therefore, instead of Constraint (5.3.11) Constraint (5.3.11’) is used to guarantee that a penalty is only incurred when $q_{\omega i} = 1$.

$$pf_{\omega i} - c(d_{\omega i} - L_i)^2 \geq -cF^2(1 - q_{\omega i}) \qquad \forall\omega \in \Omega, \forall i \in N \qquad (5.3.11')$$

In Constraint set (5.3.11’) F^2 is used instead of F, because of the quadratic term at the left side of the constraint. This quadratic term $(d_{\omega i} - L_i)^2$ can be larger than F when target i is in the beginning of the tour and the latest time of target i is large, therefore the square of F used.
This quadratic smooth penalty function and its consequences for the yield of target i are displayed in Figure 5.4.

The cost function for the first stage tour using a quadratic smooth penalty function is the same as the penalty cost function in the case that a linear penalty function is used.

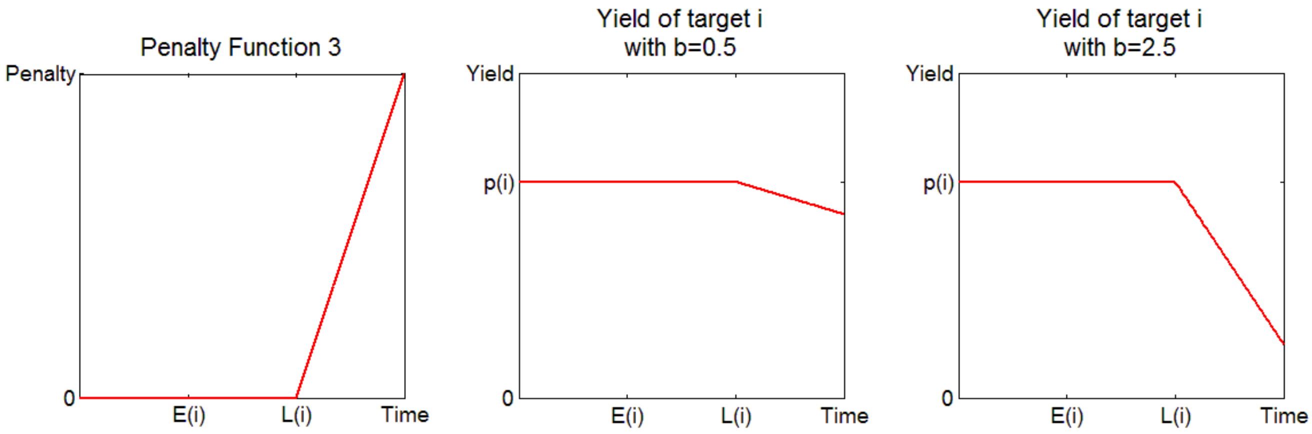

Figure 5.3: The linear penalty function and its consequences for the yield of target i

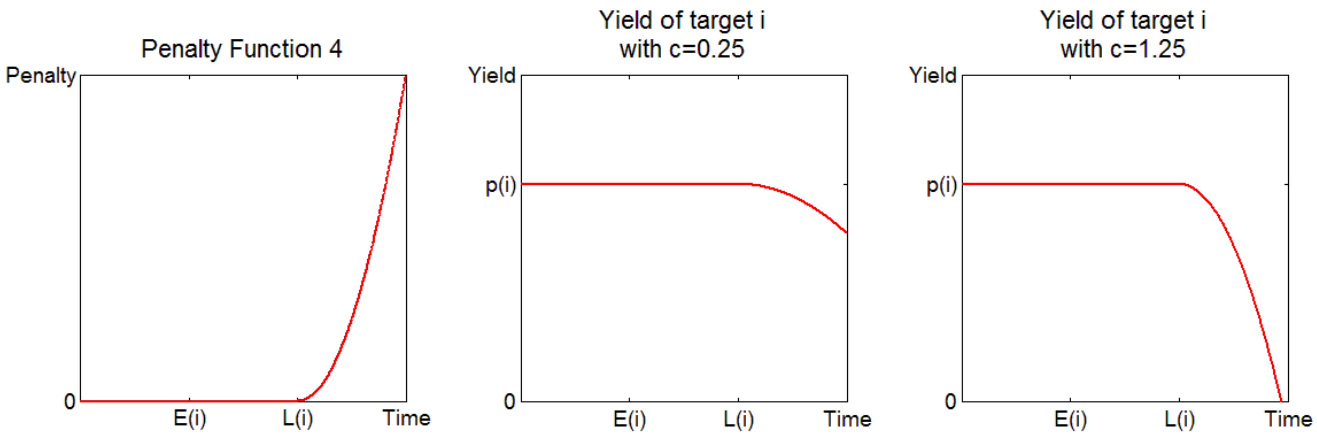

Figure 5.4: The quadratic penalty function and its consequences for the yield of target i

5.3.4 Summary

In the POPTW the objective is to determine a first stage tour which maximizes the total gathered information value and minimizes the total expected incurred penalty. The complete formulation of the POPTW is as follows:

$$\text{(P5.5)} \qquad max_y \sum_{i \in N} p_i x_i - \mathbb{E}_{\tilde{f}}\left[c\left(y, \tilde{f}\right)\right] \qquad (5.3.1)$$

$subject\ to$ Constraints $(5.3.2) - (5.3.6)$

$$q_{\omega i} \in \{0,1\} \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.3.12)$$

$$d_{\omega i} \geq 0 \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.3.13)$$

Note that when a smooth penalty function is considered,

$$pf_{\omega i} \geq 0 \qquad \forall \omega \in \Omega, \forall i \in N \qquad (5.3.14)$$

should be added to Program (P5.5). In case of a linear penalty function also Constraint (5.3.9) should be included and in case of a quadratic penalty function Constraint (5.3.11’).

6. Solution methods

In this chapter some solution methods for the CCP-model and the both SPR-models, the TSOPTW and the POPTW, are presented. First, we will discuss some approaches which are specific for a model and subsequently we will introduce a Tabu-Search Based Heuristic, based on Li et al. [32], which can efficiently solve all presented models.

6.1 Chance-Constrained Programming model

In this section we will discuss two solution methods for the Chance-Constrained Programming model, presented in Chapter 4.

The Mixed Integer Quadratically Constrained Problems (MIQCPs), presented in Section 4.3, are quadratic reformulations of the Chance-Constrained Program, presented in Section 4.2. These quadratically constrained problems, (P4.2)-(P4.4), can be solved by the optimization software package IBM ILOG CPLEX Optimization Studio (CPLEX). This solver can solve linear programming problems and mixed integer linear problems, but also convex and non-convex quadratic problems and convex quadratically constrained problems. As mentioned in Subsection 4.3.4, our reformulations of the CCP-formulation are convex quadratically constrained problems, therefore CPLEX can solve them to optimality. However, because the OPTW is proved to be NP-hard [8], it is unlikely that these problems can be solved in polynomial time.

The software package CPLEX has to be controlled by an external program, such as Java, C++ or Matlab. It is also accessible through independent modeling systems such as AIMMS. This last modeling system also provides a robust optimization add-on which can solve linear and mixed integer programming models with uncertain parameters to optimality. In this add-on also chance constraints are included in the functionalities. This means that the chance constraints discussed in Section 4.2 can be directly imported into AIMMS combined with the deterministic OPTW (P4.1). Subsequently, AIMMS makes a second order cone reformulation of the chance constrained problem, which is solved by CPLEX. This second order cone reformulation of the problem will be very similar to the first presented quadratically constrained reformulation (P4.2).
The main advantage of the quadratically constrained reformulations discussed in this thesis is that when CPLEX is used by an interface as for example Java or Matlab, the chance constrained programming formulation of the SOPTW can still be solved. Without these reformulations CPLEX should be used with an independent modeling system which can handle chance constraints. Note that most independent modeling systems can handle both formulations, which means that the quadratically constrained reformulation can also be solved using for example AIMMS.

6.2 Stochastic Programming models with Recourse

In this section we will discuss a solution method for both stochastic programming models with recourse, presented in Chapter 5.

Since we have assumed that the fuel consumption on an arc follows a continuous probability distribution, the objective functions of both SPR-models are nonlinear. Furthermore, for computational reasons it is impossible to take all scenarios of a continuous distribution into account. In order to solve these problems, Sample Average Approximation (SAA) can be used. SAA is a well-known method in literature, used to solve stochastic models with a large or infinite number of scenarios. It is introduced for two-stage stochastic programs with recourse by Shapiro and Homem-de-Mello in [41]. Their work is extended by Mak et al. [42] and Kleywegt et al. [43]. In this technique the expected value function in the objective function of a stochastic problem is approximated by the standard sample mean estimator.

Given a random sample Ω of size $|\Omega|$of the fuel usages f the standard sample mean estimator of the objective function of P(5.3),

$$max_y\, \mathbb{E}_{\tilde{f}}\left[h\left(y,\tilde{f}\right)\right] \tag{5.2.1}$$

is equivalent to

$$max_y \frac{1}{|\Omega|}\sum_{\omega\in\Omega} h(y, f_\omega)$$

Recall that decision variable y denotes that chosen arcs in the tour and parameter f represents the fuel consumption.

Therefore, the SAA objective function of the TSOPTW becomes

$$max \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left[\sum_{i\in N} p_i h_{\omega i} - \sum_{i\in N}\kappa w_{\omega i}\right] \tag{6.2.1}$$

where p_i is the profit of target i, decision variable $h_{\omega i} = 1$, if the recording of target i falls within its time window in scenario ω and $w_{\omega i}$denotes the waiting time before target i in scenario ω.

Note that by increasing the sample size $|\Omega|$, the solution based on the SAA objective function will exponentially converge to the optimal solution of the TSOPTW with a probability of one, according to Kleywegt et al. [43].

As in the TSOPTW, we can also use SAA to solve the POPTW, because of the continuity of the probability distributions of the fuel usages. Recall the Objective function of the first stage (5.3.1),

$$max_y \sum_{i\in N} p_i x_i - \mathbb{E}_{\tilde{f}}\left[c\left(y,\tilde{f}\right)\right] \tag{5.3.1}$$

where $\mathbb{E}_{\tilde{f}}\left[c\left(y,\tilde{f}\right)\right]$ is the expected penalty cost in the second stage. In the previous chapter we have introduced different penalty cost functions for the first stage tour. For fixed penalty costs, Cost function (5.3.7) should be used. Cost function (5.3.8) should be used for target dependent penalty costs. For smooth penalty costs, Cost function (5.3.10) should be selected.

The SAA-objective function of the POPTW is

$$max \sum_{i\in N} p_i x_i - \frac{1}{|\Omega|}\sum_{\omega\in\Omega} c(y, f_\omega) \tag{6.2.2}$$

where $c(y, f_\omega)$ is the penalty cost function corresponding to the selected type of penalty function.

When the original objective function of the TSOPTW or the POPTW is replaced by the SAA-objective function, both problems are linear and deterministic. Therefore, it is possible to solve these problems with CPLEX, but only for small sets of targets. This is caused by the large number of constraints and variables in these models. Furthermore, the SAA of the objective function is better when a lot of scenarios are taken into account, which would entail many second-stage variables. Therefore, we will present a Tabu-Search Based Heuristic in next section.

6.3 Tabu-Search Based Heuristic

In this section we will introduce a Tabu-Search Based Heuristic for the SOPTW (TSB), which is based on the heuristic for the SVRPTW presented by Li et al. in [32]. As mentioned earlier, the OPTW is proved to be NP-hard [8]. It is therefore unlikely that the CCP-model, the TSOPTW or the POPTW can be solved in polynomial time. To construct an offline tour for the Case Study later in this thesis, a TS-based heuristic is developed. This heuristic can efficiently approximate the CCP-model and both the TSOPTW and the POPTW. The heuristic starts with a tour which is feasible for the given model and contains a tabu list to prevent circling. After a general overview about tabu-search, the main components of the heuristic are outlined and a detailed description of the algorithm is provided.

6.3.1 Tabu-Search

Tabu-Search (TS) is one of the oldest meta-heuristics, introduced by Glover in [44]. In each iteration of the original heuristic all solutions in the neighborhood of the current solution are investigated and the best of them is selected as the new current solution, even if this solution is worse than the current solution. This enables the algorithm to escape from a local optimum. Visiting recently selected solutions is forbidden by a tabu list to prevent circling. This tabu list often does not contain forbidden solutions, but only forbidden moves. After a fixed number of iterations or after a constant number of iterations without an improvement of the best found solution, the algorithm is finished and the best found solution is returned.

In literature tabu-search is frequently used to solve both deterministic and stochastic routing problems. For example, TS is applied by Gendreau et al. [45] to solve the VRP and by Bräysy and Gendreau [46] to the VRPTW. In the case of the SVRPTW, Li et al. [32] and Taş et al. [35] have applied this technique. Also for the deterministic TOP, there are several researchers who use TS to solve the problem. The first were Tang and Miller-Hooks in [47] and after them also Archetti et al. in [48]. In this thesis we will apply TS to the SOPTW.

6.3.2 Main Components of the Tabu-Search Based Heuristic

In this subsection the main components of the TSB are discussed. We will describe the construction of the initial solution, the solution evaluation, the neighborhood structure and the tabu structure.

Initial Solution

Similar to other local search algorithms, the TSB needs an initial solution to start its exploration in the solution space. In contrast to Li et al. [32], this solution should be feasible and therefore the deterministic solution cannot be used as initial solution like is done in the cited article. Note that the deterministic solution could be feasible, but this cannot be guaranteed for the CCP-model. In the TBS a feasible initial solution is created using a best neighbor strategy. This solution is constructed by adding repeatedly from the allowed targets, the target with the highest information value-distance ratio. This ratio is equal to p_j/d_{ij} , where d_{ij} is the distance to target j from the previous added target i. This procedure is repeated until none of the non-included targets is allowed.

In case of the CCP-model, a target is allowed when, depending on the chosen modeling, the time window constraint or the deadline and eventually the waiting time constraint are satisfied with a probability of at least the given threshold β. Furthermore, the overall fuel constraint should also be fulfilled for the route including that target with a probability of at least the given threshold α.

In the TSOPTW and the POPTW all solutions are feasible, since either targets can be skipped or a penalty is incurred for a late visit at the targets. In case of the TSOPTW, a target is allowed if it can visited before its deadline with a probability of α and the depot can be reached after recording this target both in the worst case. In case of the POPTW, a target is allowed if recording this target has a positive expected yield. In both cases, the fuel realizations are assumed to be equal to their mean in this constructing phase.

Solution Evaluation

The solution evaluation is different for each of the three models. The score of a solution is equal to the objective function value of that particular model. Consequently, for the CCP-model the score of a solution is equal to the sum of the information values of the targets included in that route, eventually minus a fraction κ the total expected waiting time. For the TSOPTW and the POPTW, sample scenario evaluation is used to determine the score of a route, which means that the expectation of the objective function is estimated by the mean of the scores of a sample of scenarios. When the TSOPTW should be approximate by the heuristic, for each scenario the recourse action is applied to the given solution and the score of the route is set equal to the mean of the realized profits over all scenarios. The score of a solution for the POPTW is determined by calculating the penalty incurred by execution of the route for all scenarios. The total gathered profit minus the average incurred penalty will be the score of the route. Note that for every solution evaluation of the TSOPTW or the POPTW the same scenarios should be used for an equitable comparison between the score of different solutions.

Neighborhood Structure

The neighborhood of a solution is defined by six different neighborhood operators. A neighbor, which is a candidate solution, can be generated by applying one of the operators to the current solution. The six operators we have used this TS-based heuristic are

- Reversal: the inversion of a continuous segment of targets in the tour.
- Exchange: the interchange of two targets in the tour.
- Relocation: the displacement of a target in the tour to another place in the tour.
- Replacement: the substitution of a target in the tour by a non-included target.
- Insertion: the addition of a target to the tour.
- Removal: the deletion of a target from the tour.

Note that in Li et al. [32] the SVRPTW is studied, while the SOPTW is addressed in this thesis. Therefore, most operators used in our research differ from the operators used in the previously mentioned article. The Reversal operator is also used in the cited article, which is called the 2-Opt operator. The Exchange and Relocation operators are based on the operators with the same name in the research by Li et al. [32], but adjusted to the SOPTW. The other three operators are specific for the Orienteering Problem and therefore not based on the used article. Since in the VRP all targets should be visited, these operators are not allowed or not possible.

For each candidate solution an operator is randomly selected and applied to the current solution. To decrease the number of solution evaluations required in each iteration, only a fixed number of all candidate solutions is investigated. The probability to select a particular operator to generate a candidate solution is the same for each operator. In contrast to Li et al. [32], all candidate solutions should be feasible. All possible solutions are feasible for the TSOPTW and for the POPTW, but in the CCP-models a solution is feasible if the chance constraints are satisfied.

In order to consider the problem structure with time windows and to improve the quality of the candidate solution, the operators are not randomly applied to the current solution, like is done by Li et al. [32]. We have divided the current solution into n equal parts. The Reversal and Exchange operator are applied within a randomly chosen part of the current solution. The displacement of a target in the Relocation operator is also only allowed within a randomly chosen part. However, these operators are applied to randomly chosen targets or segment of targets within the given part. For the application of the Replacement and Insertion operator a non-included target is randomly selected. According to the deadline of this target, the part of substitution or insertion is determined. Part p is selected if the deadline of the target which should be inserted, falls in the $\left\lfloor \frac{p}{n} \right\rfloor$ -th part of the fuel capacity F. Within this determined part the place of the inserted or substituted target is randomly chosen. The last operator, Removal, is not affected by the dividing into parts of the current solution and by applying this operator a randomly chosen target of the current solution is deleted.

The new current solution is the solution in the investigated neighbors with the highest score. Note that this score can be lower than the score of the current solution. However, the solution with the highest overall score will be stored.

Tabu Structure

To avoid circling between a subset of solutions and to explore a larger part of the solution space, a tabu list with prohibited moves is implemented. These prohibited moves are the reverse moves of the previous modifications in order to prevent a return to a previous visited solution. In each iteration after a new current solution is selected, the inverse of the modification leading to this solution is declared tabu and stored in the tabu list. We consider a random tabu structure, as is done by Gendreau et al. in [45] and by Li et al. in [32]. This means that the number of iterations a prohibited modification is tabu, is a random integer which is uniformly generated from $[\underline{\theta}, \overline{\theta}]$. Note that this number of iterations could be different for each prohibited modification.

We define six different tabu lists $TABU_i$ where $i = 1,2, \dots ,6$, to achieve the prohibited modifications for the six operators. The first five tabu lists are matrices, where element $TABU_i(j,k)$ specifies the tabu status of the modification (j,k) for operator i. If $TABU_i(j,k) > 0$, modification (j,k) is tabu for operator i. The sixth tabu list is a column where element $TABU_6(j)$ indicates the tabu status of the modification (j) for the sixth operator. If $TABU_6(j) > 0$, modification (j) is tabu for operator Removal.

The modification (j,k) with $j \neq k$ in Reversal is defined as the inversion of the part between the j^{th} and k^{th} target in the tour. Therefore, repeating modification (j,k) leads to the original solution. Also applying modification (k,j) after modification (j,k) restores the original solution. For this reason it is forbidden to re-inverse the segment between the j^{th} and k^{th} target in the tour and consequently, the modifications (j,k) and (k,j) for operator 1 become tabu.

In Exchange the modification (j,k) with $j \neq k$ represents the interchange of the j^{th} and the k^{th} target in the tour. As in Reversal, both repeating modification (j,k) and modification (k,j) re-establishes the original solution. Therefore, these both modifications become tabu for operator 2. Note that Exchange modification (j,k) can also be undone by Reversal modification (j,k) or (k,j) if the absolute difference between j and k is less than three. The opposite is true as well, applying Exchange modification (j,k) or (k,j) after Reversal modification (j,k) restores the original solution if $|j-k| \leq 2$. To prevent circling also these modifications has to become tabu after the respective modification.

The displacement of the j^{th} target of the tour to the k^{th} position in the tour is denoted by Relocation modification (j,k) with $j \neq k$. Notice that the new position of the j^{th} target is determined in the tour without this target. This displacement can be undone by the inverse modification (k,j), which has to become tabu to avoid a return to previous solutions. When a target is moved only one position forward or backward, which means that $|j-k| = 1$, the Relocation modification (j,k) can also be undone by repeating Relocation modification (j,k) or by executing Exchange modifications (j,k) or (k,j) or Reversal modifications (j,k) or (k,j), since exchanging of two consecutive targets in the tour is the same as moving one of the two targets one position forward or backward. This means that also the Exchange modification (j,k) and Reversal modification (j,k) can be reversed by Relocation modification (j,k) or (k,j) when $|j-k| = 1$.

The Replacement modification (j,k) is defined as the substitution of the j^{th} target in the tour by target k, where target k has not yet been included in the tour. This modification can be reversed by the Replacement modification (j,l), where target l is equal to the j^{th} target of the original tour. Note that this operator can only be applied when not all targets are included in the current solution. Furthermore, this is also the case for the fifth operator, Insertion.
In Insertion the modification (j,k) describes the addition of the not yet included target k at position j in the tour. Operator Removal effectuates the opposite, because the deletion of the j^{th} target in the tour is denoted by Removal modification (j). Therefore, Removal modification (j) after Insertion modification (j,k) leads to the original solution and also Insertion modification (j,l) after Removal modification (j) where target l is equal to the j^{th} target of the original tour restores the original tour. Consequently, it is forbidden to reinsert target l at position j or to remove the just inserted target at position j.

We have implemented a tabu list with prohibited modifications instead of prohibited solutions in order to reduce the computation time to verify a solution is tabu. However, this often results in more than one solution being tabu. Some of these prohibited solutions could have a high score and might not have been visited. To mitigate this problem, aspiration criteria are introduced. These criteria allow overriding of the tabu status of a solution if they are satisfied. In this thesis we use a commonly used aspiration criterion that a tabu solution can be overridden if it has a higher objective value than the currently best known solution.

6.3.1 Tabu-Search Based Heuristic for the SOPTW

In this subsection we will provide a detailed description of the Tabu-Search Based Heuristic for the SOPTW (TSB).

The heuristic starts with constructing an initial solution in the way described in the previous subsection, corresponding to the model which should be solved by the TSB. For this initial solution, which is the current solution, the objective function of the investigated model is calculated. Since this is the only solution found up to this far, it is also the currently best known solution.
Subsequently, the heuristic follows a loop for a fixed number of iterations, T_{max}.
First, a constant number N_{max} of candidate solutions, which are neighbors of the current solution, are generated using the six operators, presented in the previous subsection. The tabu status of the modifications made by the different operators is not taken into account in this stadium of the loop but the candidate solutions should be feasible. Each of these N_{max} candidate solutions is evaluated and the score of the model objective function is calculated.
Secondly, for all tabu solutions in the investigated part of the neighborhood is verified whether the aspiration criterion is satisfied. If that is not the case, the tabu solution is removed from the list of candidate solutions, otherwise the solution is stored by the non-tabu solutions at the list of admissible solutions.
After that the best solution, which is the solution with the highest score in the list of admissible solutions, is determined. If the score of this best solution is higher than the score of the best solution found so far, the currently best known solution is updated. The best solution found in this iteration is the new current solution.
Consequently, the tabu list should be updated. The inversion of the modification made to move from the previous current solution to the new current solution has become tabu for a random number of iterations. The tabu status of all other modifications is reduced by one.
After the completion of the loop, the algorithm is finished and the best solution found is returned.

The stopping criterion of the TBS is also different of the TS-based heuristic of Li et al. in [32]. In our heuristic we use only one stopping criterion which is that the heuristic is finished after a fixed number of iterations, while in Li et al. [32] the heuristic is also finished when there is no improvement of the best found score for a constant number of consecutive iterations.

Part III

Adaptive Routing

In this part we will discuss some strategies to adjust the planned tour during the flight in real-time in order to anticipate to the realized fuel consumptions. In Chapter 7 we will introduce adaptive routing strategies for the online part of the UAV-MPP with time windows and stochastic fuel consumption where the initial tour constructed in the offline part is adapted to respond to the actual circumstances. In the other chapter of this part we will present one-step-ahead routing strategies, where only the next location is determined. Both deterministic strategies where the stochasticity of the fuel consumption is not taken into account and stochastic strategies are introduced in Chapter 8.

7. The online problem

In this chapter we will discuss some strategies to adjust the initially constructed tour to the realized fuel usages. The initial tour which is constructed with the methods presented in Part II, could not be achievable under all circumstances. For example, during the execution of an initial tour constructed with the CCP-model, the UAV is running out of fuel with a probability of $1-\alpha$. The TSOPTW approach is likely to schedule a relatively high number of targets, since the TSOPTW policy allows planned targets to be skipped. Since for running out of fuel in the POPTW only a penalty is incurred, the initial tour constructed in the POPTW could require more fuel than the given fuel capacity of the UAV. Therefore, two adaption strategies are proposed in this chapter. The first adaption strategy is the recourse action we already used in the determination of the initial tour of the TSOPTW while an alternative recourse action is used in the second adaption strategy.

7.1 Recourse Action

First, we will recall the recourse action we have defined in Section 5.2. In the second stage of the TSOPTW we have applied the recourse action that the next target in the final route is equal to the next target in the initial tour, except when the probability that this target can be reached before its deadline is below a predefined level α or when the remaining fuel quantity is insufficient to fly to this target and back to the depot in the worst case. In that case the next target of the initial tour is skipped in the final route. This means that the next target in the final route is the first target in the initial tour which satisfies the recourse requirements, starting from the current location. This recourse action is applied during the second stage of the TSOPTW, which means that the initial tour constructed with this method is optimized for this adaptive routing strategy. During the flight this strategy can be applied to all initial tours. However, it is expected that the final tour based on the initial tour of the TSOPTW will gather more information value than the final tours based on an initial tour of another method. This is due to the fact that the recourse action is already taken into account in the construction of the initial tour in the TSOPTW.

7.2 Alternative Recourse Action

In order to overcome the problem just mentioned, we will define also another recourse action which is more based on the structure of the CCP-model and the POPTW. In contrast to the TSOPTW, the possibility to skip targets during the execution of the initial tour does not exist in the CCP-model and the POPTW. Therefore, in the alternative recourse action the initial tour is followed until the remaining fuel quantity is insufficient to fly to the next target and back to the depot in the worst case. In that case the UAV has to go back to the depot and the flight is finished. Since it is not possible to skip a target because of the small probability to reach it before its deadline, it is reasonable that the departure time of some targets in the flight will fall outside their time window. The gathered profit for these targets is equal to zero. It is expected that this is often the case by the execution of the by TSOPTW constructed initial tour, because in the construction the possibility to skip targets is considered.

8. One-Step-Ahead Routing

In this chapter we will discuss another approach to solve the SOPTW. The previous approach constructs an initial tour before the flight which is adjusted to the realized fuel usages during the flight, while in this approach the next location is determined after recording the previous. We will introduce this approach as the One-Step-Ahead Routing (OSAR) approach. In this chapter we will present both Deterministic and Stochastic One-Step-Ahead Routing.

In OSAR the next location is determined during the flight when the recording of a target is finished. At that moment the realized fuel usage on the flight path just flown, as well as the realized fuel during the recording of the current target becomes available and consequently, also the total fuel consumption during the flight up to that point is known. Before the UAV, can continue its flight, the next location should be determined. This means that the determination of the next target has to be done very quickly, since it should be done in a real-time situation.

We will present two approaches for Deterministic OSAR in this chapter. Both approaches ignore the stochasticity of the fuel consumption, which means that the probability distribution functions of the fuel usages is not taken into account in the determination of the following location of the flight. In order to prevent the UAV of running out of fuel, the worst case realizations of the fuel usages are however taken into consideration. We will also present the stochastic variant of both approaches, which do take into account the probability distribution of the fuel consumption.

8.1 Best Neighbor Approach

The first approach we will introduce is the Best Neighbor Approach (BNA). In this approach the next location of the UAV is determined by selecting the feasible target with the highest profit-fuel ratio. Two variants of the BNA are presented in this section, the Deterministic Best Neighbor Approach (DBNA) and the Stochastic Best Neighbor Approach (SBNA).

8.1.1 Deterministic Best Neighbor Approach

In this deterministic approach the stochasticity of the fuel consumption is ignored, which means that in the determination of the next location of the flight of the UAV is assumed that the fuel usage on each arc is fixed and equal to the mean of the given probability distribution function. Therefore, a target is feasible if it can be recorded before its deadline, taking into account the already used fuel and the expected fuel usage on the arc from the current position to that target. Given the current position i, the total fuel consumption up to the current location, ff_i and the expected fuel usage on the arc (i,j), μ_{ij}, target j is feasible if

$$ff_i + \mu_{ij} \leq L_j \tag{8.1.1}$$

Note that the total fuel consumption at the departure of the depot at the beginning of the flight is equal to zero, accordingly $ff_1 = 0$. Note that for the depot $i = 1$.

In order to prevent the UAV for running out of fuel, a worst case control is also provided for the targets. A target is feasible if both the deadline condition and the worst case condition are satisfied. The worst case control for a target is fulfilled if the remaining fuel is sufficient to fly from the current position to that target, to record that target and to fly back to the depot, all under the worst case circumstances. Given the worst case fuel usage for the flight from the current position i through target j back to the depot, W_{ij}, the worst case condition is met for target j if

$$ff_i + W_{ij} \leq F \tag{8.1.2}$$

where ff_i is the total fuel consumption up to the current location and F is the fuel capacity of the UAV.

The next location in the tour of the UAV is the feasible target with the highest profit-fuel ratio, which has not yet been visited, i.e. the maximum information value per fuel unit is selected. The profit-fuel ratio for a target is calculated as the profit of that target divided by the fuel usage to collect this information value. The fuel required to collect the information value of target j is the sum of the fuel used to fly from the current location i to target j, the fuel spent on the recording of target j and if the UAV arrives before the earliest time of target j, the fuel consumed during the waiting time before target j. The profit-fuel ratio of target j is equal to

$$\frac{p_j}{\mu_{ij} + w_j}$$

where p_j is the profit of target j and w_j is the waiting time before target j.

This process is repeated until there are no feasible and unvisited targets anymore. In that case the following location of the UAV is the depot and subsequently the flight is finished.

8.1.2 Stochastic Best Neighbor Approach

Since the stochasticity of the fuel consumption, which is ignored in the DBNA, is taken into account in the SBNA, the feasibility conditions and the profit-fuel ratio are different than in the DBNA. The main processes of the approach, such as the selection of the following destination of the UAV, are still the same. Due to the stochastic fuel usages on the flight paths in the area of operations, a target is feasible if there is a strictly positive probability that the target can be recorded before its deadline given the total fuel consumption up to the current position of the UAV. Independent of the assumed probability distribution function of the fuel usages on the arcs, the probability to reach a target with a deadline smaller than the already used fuel is equal to zero, since a negative fuel usage on an arc is meaningless. Given the total fuel consumption up to the current position i, ff_i, the deadline condition for target j is satisfied if

$$ff_i \leq L_j \tag{8.1.3}$$

Besides the deadline condition, the worst case control is applied in the SBNA as well, which means that feasible targets should also meet the worst case condition, Condition 8.1.2.

Moreover, also in the profit-fuel ratio used in the SBNA, the stochasticity of the fuel consumption is considered. The profit-fuel ratio of a target is the expected profit of that target given the current fuel consumption divided by the expected fuel required to collect this information value, thus the ratio is equal to

$$\frac{\mathbb{E}(Profit)}{\mathbb{E}(Fuel)}$$

The expected profit of a target is calculated as the probability to record that target before its deadline multiplied by the information value of that target. For the definition of this probability for target j, the remaining time before its deadline t_j should be determined, which is equal to

$$t_j = L_j - ff_i$$

To record target j before its deadline, the realized fuel consumption of arc (i,j) should be smaller or equal to time t_j. The probability that target j can be reached before its deadline, given the total fuel consumption up to the current position i, is equal to

$$P(f_{ij} \leq t_j)$$

where f_{ij} is the stochastic fuel usage on the arc (i,j), which follows a given probability distribution. Consequently, the expected profit of target j is

$$\mathbb{E}(Profit) = P\left(f_{ij} \leq t_j\right) * p_j$$

The expected fuel to collect this information value contains two parts. The first part is the expected fuel usage during the flight from current location i to target j and during the recording of target j. The expected fuel usage $\mathbb{E}(f_{ij})$on arc (i,j) is equal to μ_{ij}. The second part is the expected waiting time before target j. The waiting time of target j is defined as the maximum between zero and the earliest time of target j of which the realized fuel usage on arc (i,j) and the fuel consumption before the current position i are subtracted, accordingly

$$w_j = \max(0, E_j - f_{ij} - ff_i)$$

Consequently, the expected waiting time before target j is

$$\mathbb{E}(w_j) = \max(0, E_j - \mu_{ij} - ff_i)$$

Therefore, the profit-fuel ratio for target j in the SBNA is equal to

$$\frac{P\left(f_{ij} \leq t_j\right) * p_j}{\mu_{ij} + \max\left(0, E_j - \mu_{ij} - ff_i\right)}$$

where i is the current position and ff_i is the total fuel consumption up to target i.

8.2 Repeated Tabu-Search Based Heuristic Approach

We will now present another method to determine the following destination in OSAR. While in the BNA only the best neighbor of the current position is selected, also the neighborhood of the following destination is taken into account in the Repeated Tabu-Search Based Heuristic Approach (RTSBA). In this approach a deterministic or stochastic variant of the Tabu-Search Based Heuristic (TSB), presented in Section 6.3, is executed to determine the next target of the flight. The output of this heuristic is a route for the remaining part of the flight, but only the first target of this route from the current position is considered. In the remainder of this section we will discuss two variants of the RTSBA, the Repeated Deterministic Tabu-Search Based Heuristic Approach (RDTSBA) and the Repeated Stochastic Tabu-Search Based Heuristic Approach (RSTSBA). In the first variant the stochasticity of the fuel consumption is ignored, while the second variant will take this information into account.

8.2.1 Repeated Deterministic Tabu-Search Based Heuristic Approach

In this approach a deterministic variant of the TSB, presented in Section 6.3, is executed to determine the next location in the flight of the UAV. Every time the UAV has arrived at a target and the recording of that target is finished, the feasible and unvisited targets are selected. These targets are the input of the TSB. A target is feasible if both Condition (8.1.1) and Condition (8.1.2) are satisfied.

The main structure of the deterministic variant of the TSB is the same as the main structure of the TSB, only a part of the main components of the heuristic are adjusted. Both the Neighborhood Structure and the Tabu Structure remain the same. The initial solution is constructed using the DBNA of Section 8.1.1 where is assumed that the realized fuel consumption is equal to its expectation. The score of a route which is determined in the solution evaluation is equal to sum of the profits of the targets in the route. Note that these solutions are routes starting at the current position.

As in the original TSB, all candidate solutions should be feasible. In this deterministic variant a route is feasible if each target in the route can be reached before its deadline when is assumed that all fuel realizations are equal to their expectation. Each target in the tour should also satisfied the worst case control when is assumed that all fuel realizations before that target are equal to their expectation. Given the current position i and the realized fuel consumption before the current position ff_i, the first condition is satisfied for solution R if

$$ff_i + \mathrm{M}_j \leq L_j \qquad \forall j \in R \qquad (8.2.1)$$

where M_j is the sum of the expectations of the fuel usages on the arcs in R before target j and target j is after target i in solution R.

The second condition is a worst case condition, which is applied to prevent the UAV for running out of fuel. Given the current position, the realized fuel consumption before the current position and the targets j and k, which are consecutive targets in solution R, the worst case condition is satisfied if

$$ff_i + \mathrm{M}_j + W_{jk} \leq F \qquad \forall j, k \in R \qquad (8.2.2)$$

where W_{jk} is the worst case fuel usage for the flight from target j through target k back to the depot and F is the fuel capacity of the UAV.

The output of this deterministic variant of the TSB is the best found route with the highest score from the current position to the depot. Only the first target in this route is considered, since RDTSBA is a One-Step-Ahead Routing Approach.

8.2.2 Repeated Stochastic Tabu-Search Based Heuristic Approach

In this section we will discuss an approach where repeatedly a stochastic variant of the TSB is executed. Like in the RDTSBA, every time the recording of a target is finished, the set of feasible and unvisited targets is determined. In RSTSBA targets are feasible if both Condition 8.1.3 and Condition 8.1.2 are satisfied. This set of targets is the input of the stochastic variant of the TSB.

The initial solution of the stochastic variant of the TSB is constructed using the SBNA, where we assume that the realized fuel usages are not available and only their distribution function is known. This results in a larger uncertainty in the departure times of the following targets. In order to prevent a long initial solution with a lot of targets which have a probability of zero to be reached before their deadline, for each target in the initial solution should hold that the minimum fuel usage during the route before that target is smaller than its deadline. Consequently, all targets in the initial solution have a positive probability to be reached before their deadline.

Furthermore, the solution evaluation in this stochastic variant of the TSB is also different as the evaluation of the deterministic variant, presented in Subsection 8.2.1. The score of a solution in the stochastic TSB is equal to the expected gathered information value during the executing of the route. The expected profit of target j in solution R, which is a route from the current location i to the depot, is equal to

$$\mathbb{E}(Profit) = P\left(f_{kj} \leq t_j\right) * p_j$$

where target k is the immediate predecessor of j and $t_j = L_j - ff_i - \mathrm{M}_k$. The score of solution R is equal to the sum of the expected profits of all targets in the tour.

In this stochastic variant we assume that all solutions are feasible. By using the just presented solution evaluation, solutions with at the beginning of the route a target which has a small probability to be reached before its deadline, will have a low score. This is caused by the fuel consumption during the flight to and the recording of this 'closed' target. Through this useless fuel usage the following targets have a smaller probability to be recorded in time and therefore their expected profit will be smaller. Note that the worst case control for the entire route is not necessary, since the worst case condition, Condition 8.1.3, is satisfied for all input targets and only the first target of the route is taken into account.

Like the deterministic variant, the output of this stochastic variant of the TSB is the best route found with the highest score from the current position to the depot. As mentioned before, since the RSTSBA is an OSAR approach, only the first target in the output route is considered.

Part IV

Case Study

In this part we will evaluate a case study executed to test the previously discussed approaches to solve the SOPTW. In Chapter 9 we will describe the data used in our case study. The model settings and the used performance measures are discussed in Chapter 10, while in the following chapters the results of the different approaches are presented. In Chapter 11 the results of the hybrid approach are presented. The One-Step-Ahead-Routing approach is evaluated in Chapter 12.

9. Data description

To test the in Part II and III introduced approaches, models and strategies, we extend some of the deterministic datasets used by Vansteenwegen in [9] to stochastic datasets by introducing stochastic fuel consumption on the flight paths. From these datasets we only use the first nine datasets (c101-c109). These sets are based on the Solomon's datasets of vehicle routing problems with time windows. For all of these considered datasets the deterministic optimal routes are known. In this chapter we will give a description of the structure of these datasets and the probability distribution of the fuel usages on the flight paths and during the recording.

9.1 Test instances

All test instances contain the same 100 targets in an area of 100x100. As starting and ending point of the flight a depot is situated in the middle of the area. The targets are situated in clusters of approximately ten targets around the depot. Each of these targets has a fixed information value which is a multiple of ten in the range of [10,50]. In figure 9.1 the area of operations of our case study is displayed.

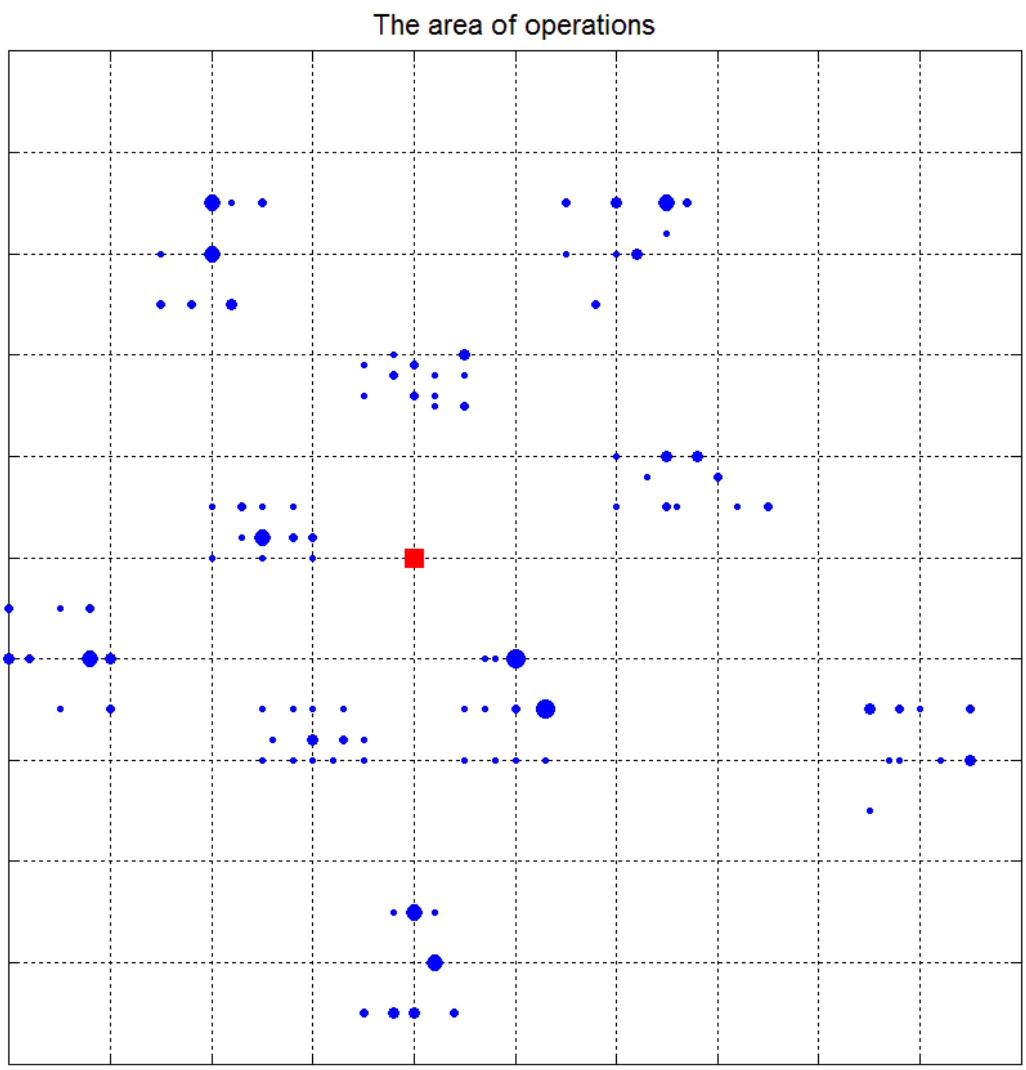

Figure 9.1: The area of operations where the size of a target represents its information value and the depot is displayed by a red square

Since both the locations of the targets and depot and the information values of the targets are the same for each of the nine test instances, the differences between the considered test instances are in the time windows of the targets. However, the time window of the depot is equal for each instance, which means that the fuel capacity of the vehicle, e.g. UAV, is also the same in each test instance. We assume that the UAV has a fuel capacity F of 1236 units.

In the remaining part of this section we will discuss for each test instance some characteristics of the assigned time windows.

The time windows of first four test instances follow a specific structure. All assigned time windows in Test instance c101 have a length of less than 100 fuel units, which means that for each target i in the instance $L_i - E_i \leq 100$. In the second instance 75 targets have a small time window, which means that it has a length of less than 100 units, and 25 targets have a very large time window, which means that it has a length of more than 1000 fuel units. The small time windows are the same as the time windows in the first test instance. The percentage of very large time windows increases in the third test instance to 50%, which means that 50 targets have a small time window and 50 targets have a very large one, and to 75% in the fourth instance. The targets with a very large time window in an instance are also in the part of targets with a very large time window in the consecutive test instance. Therefore, comparing two consecutive test instances, the time windows of only 25 targets will be different.

The other five test instances are individual cases where the time windows are different for each instance. Test instances c105, c106 and c108 include time windows of different length. In Test instance c105 the assigned time windows have a length of at least 75 up to 177 fuel units. The minimum length of the time windows in Test instance c106 is 29 fuel units and the maximum length in this test instance is 387 units, while in Test instance c108 the length of time windows ranges between 149 and 353 fuel units.
In the Test instance c107 and c109 the length of all time windows is the same. This length is equal to 180 units for Test instance c107 and for Test instance c109 this length is equal to 360 fuel units.

In Table 9.1 for each test instance the average length of the time windows as well the minimum and maximum length are displayed. Based on the characteristics of the lengths of the assigned time windows the test instances are classified in categories. The first category contains the test instance with small time windows. The average length of the time windows of the test instance in Category 1 is at most 160 fuel units, while the maximum length should be lower than 400 units. The average length of the time windows of the test instances in the second category is slightly longer than of the test instances in Category 1. Therefore, Category 2 includes three test instances with medium time windows. The test instances with both small and very large time windows are classified in Category 3. The maximum length of the time windows of the test instances in this third category is more than 1100 fuel units, while the minimum length is less than 50. In the fourth column of Table 9.2 shows the category classification of the test instances.

Test instance	Average Length	Minimum Length	Maximum Length	Category
c101	60.76	37	89	1
c102	325.69	43	1135	3
c103	588.49	43	1136	3
c104	852.94	43	1136	3
c105	121.61	75	177	1
c106	156.15	29	387	1
c107	180	180	180	2
c108	243.28	149	353	2
c109	360	360	360	2

Table 9.1: The characteristics of the assigned time windows in the test instance as well their category classification.

9.2 Stochastic fuel consumption

The average fuel usage on the flight path of target i to target j is equal to the Euclidean distance between target i and target j. We set the average fuel usage during the recording of target j equal to the recording times $\mathfrak{r}_j$ that are given in the data sets. This deterministic recording time is the same for all targets and is equal to 90 fuel units. The recording time of the depot is equal to zero. Consequently, the mean of the probability distribution function of the fuel usage on arc (i,j) is equal to $\mu_{ij} = d_{ij} + r_j$, where d_{ij} is the distance between target i and target j.

We assume that the maximum deviation $\mathfrak{s}_{ij}$ of this average usage on arc (i,j) consists of two parts. The first part is related to the distance between the two locations. We assume that this part is a fixed percentage $\mathfrak{a}$ of the given distance. The second part is related to the time spend for recording a target, which is assumed to be a fixed percentage $\mathfrak{c}$ of the recording time of a target. Summarizing, the realizations of the fuel usage on arc (i,j) are in the interval $[\mu_{ij} - \mathfrak{s}_{ij}, \mu_{ij} + \mathfrak{s}_{ij}]$, where $\mathfrak{s}_{ij}$ is equal to $\mathfrak{a}d_{ij} + \mathfrak{c}\mathfrak{r}_j$. In this case study we use $\mathfrak{a} = 0.15$ and $\mathfrak{c} = 0.25$, consistent with the settings used by Evers et al. [1].

In the reformulation of the Chance-Constrained Programming model it is assumed that the fuel usage on each individual arc is normally distributed, while in the Recourse Action of the TSOPTW is assumed that the worst case usage on each arc can be defined. Since the normal distribution ranges from negative infinity to positive infinity, we assume that the fuel usages are truncated normally distributed. To construct this truncated distribution we use a normal distribution with the mean equal to μ_{ij}, which we have defined earlier. The variance of this normal distribution is based on the triangular distribution. In contrast to the normal distribution, the triangular distribution is a continuous probability distribution with a lower limit $\mathfrak{l}$, an upper limit $\mathfrak{u}$ and a mode $\mathfrak{m}$. According to Evans et al. [49], the variance of a triangular distribution is equal to

$$\frac{\mathfrak{l}^2 + \mathfrak{u}^2 + \mathfrak{m}^2 - \mathfrak{l}\mathfrak{u} - \mathfrak{l}\mathfrak{m} - \mathfrak{u}\mathfrak{m}}{18} \tag{9.1}$$

When the limits of this triangular distribution are set equal to the limits of the interval of the fuel usage and the mode is set equal to the mean, the variance of the normal distribution of the fuel usage of arc (i,j) is equal to

$$\frac{(\mu_{ij} - \mathfrak{s}_{ij})^2 + (\mu_{ij} + \mathfrak{s}_{ij})^2 + \mu_{ij}^2 - (\mu_{ij} - \mathfrak{s}_{ij})(\mu_{ij} + \mathfrak{s}_{ij}) - \mu_{ij}(\mu_{ij} - \mathfrak{s}_{ij}) - \mu_{ij}(\mu_{ij} + \mathfrak{s}_{ij})}{18} \tag{9.2}$$

Equation (9.2) can be rewritten as

$$\frac{\mathfrak{s}_{ij}^2}{6} \tag{9.3}$$

which means that the standard deviation of this probability distribution is equal to

$$\sigma_{ij} = \frac{\mathfrak{s}_{ij}}{\sqrt{6}} \tag{9.4}$$

In figure 9.2 the probability density functions of both the triangular distribution and the normal distribution of arc (i, j) are displayed.

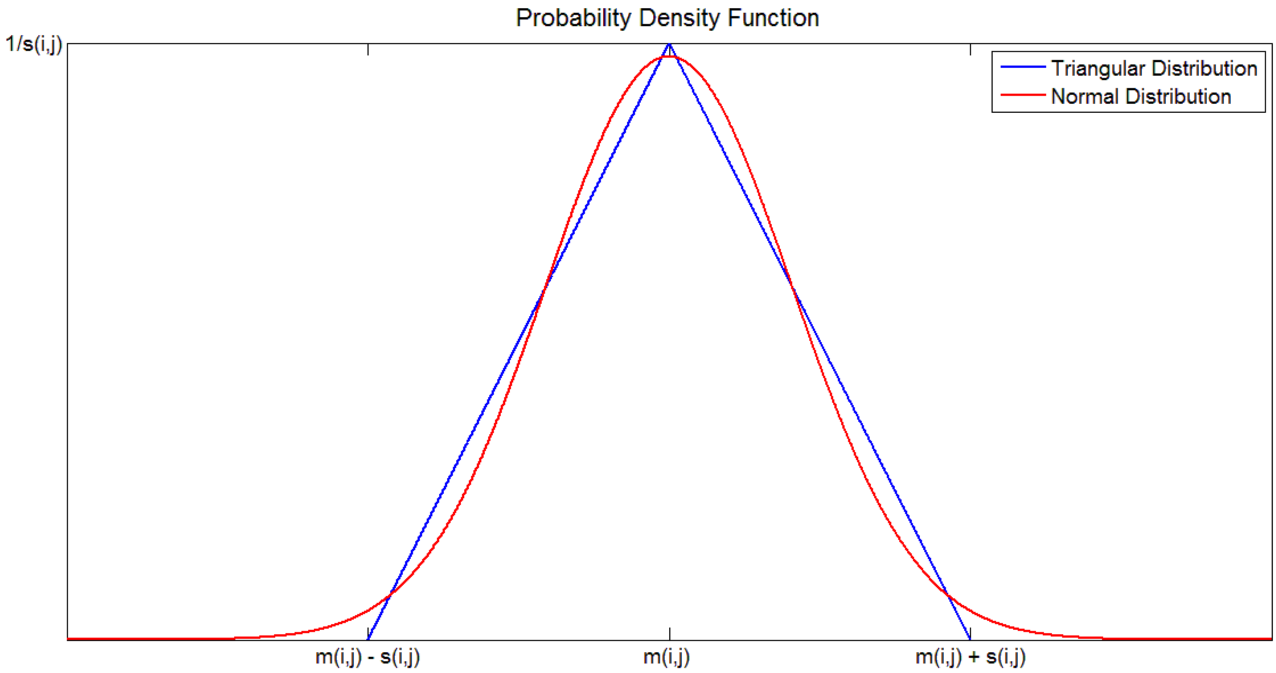

Figure 9.2: The probability density functions of both the triangular distribution and the normal distribution of arc (i, j).

To construct the truncated normal distribution of arc (i, j), this given normal distribution is truncated at the limits of the interval of the fuel usage on the arc. In the scenario construction this is done by changing all values outside the interval in a random sample of the given normal distribution by a new random number of the given distribution, until this value falls inside the interval.

By truncating the normal distribution an error is made in the CCP-models which assume that the fuel usages are normally distributed. Since at both sides of the normal distribution less than one percent is truncated, this error will be small. For illustration in Figure 9.3 depicts the cumulative density functions of the right tail of both the normal distribution and the corresponding truncated normal distribution of arc (i, j).

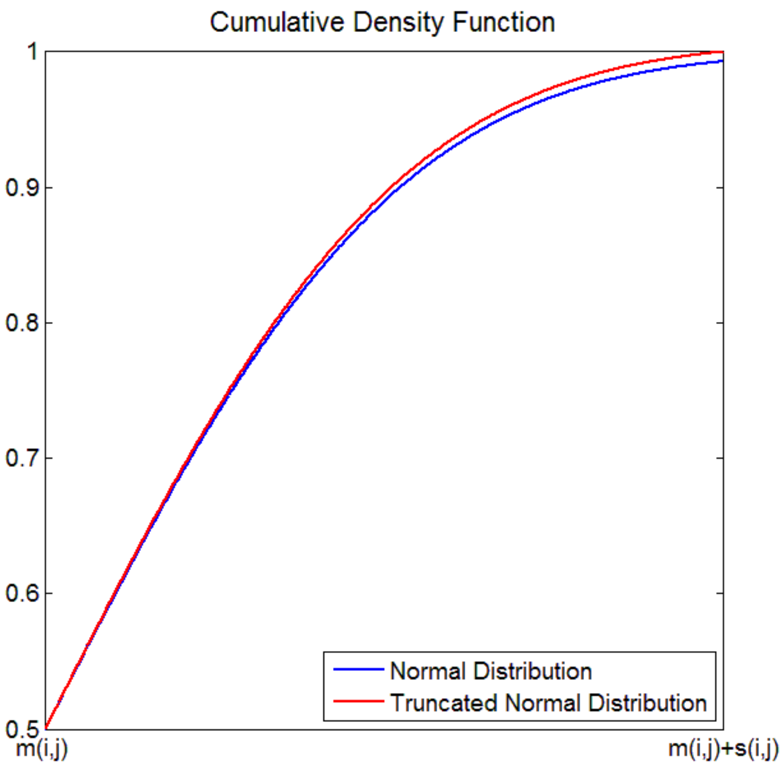

Figure 9.3: The cumulative density functions of the right tail of both the normal distribution and the truncated normal distribution of arc (i, j).

10. Experimental Settings

In this chapter we will discuss the model settings of the different approaches used in this case study. In the first section the settings of both the hybrid approach and the One-Step-Ahead-Routing approach are discussed. After that, the performance measures used in this case study are presented in the second section.

10.1 Model Settings

In this section the parameters of hybrid approach are discussed. We will describe the parameter settings of the TSB-heuristic as well as the parameter settings of both the different offline approaches and online strategies. Subsequently, the parameter settings of the OSAR-approach will be discussed.

10.1.1 Hybrid Approach

In this subsection we will discuss consecutively the parameter settings of the TBS, the offline methods which are used to construct an initial tour and the online strategies which are used to adjust the tour during the flight.

Tabu-Search Based Heuristic

In our case study we use the TSB-heuristic which is presented in Section 6.3 to construct an initial tour for the hybrid approach of the SOPTW. The TSB requires some settings defining the experimental design. In the first part of this subsection we will discuss which values for the parameters are chosen.

For all different approaches to construct an initial tour, we fix the maximum computation time to one minute. This results in a limitation of the total number of solution evaluations during the construction by the TSB-heuristic. For the different approaches, the total number of solution evaluations which can be done within one minute is displayed in the first column of Table 9.2.

Recall that in every iteration of the TSB heuristic $Nmax$ candidate solutions are constructed and evaluated. The total number of solution evaluations should be divided over the iterations. For example, all solution evaluations can be done within one iteration. In that case, the number of iterations T_{max} is equal to one, while the number of candidate solutions N_{max} is equal to the total number of solutions evaluations. Also, the opposite is possible where every iteration only one candidate solution is constructed. From earlier experiments, the best division of the total number of solution evaluations is determined and the corresponding values of T_{max} and N_{max} are displayed in the second and third column of Table 9.2.

Approach	# of solution evaluations	T_{max}	N_{max}
TSOPTW	60000	2400	25
CCP-model	120000	960	125
POPTW with Penalty Function 1	300000	625	480
POPTW with Penalty Function 2	300000	1500	200
POPTW with Penalty Function 3	300000	32	9375
POPTW with Penalty Function 4	300000	250	1200

Table 9.2: Parameter settings for the TSB for the different approaches.

Preliminary experimental testing showed that good results are obtained when the prohibited moves remain tabu for as much as at least 10 iterations with a maximum of 15 iterations. We therefore use $\underline{\theta} = 10$ and $\overline{\theta} = 15$, where $\underline{\theta}$ and $\overline{\theta}$ are the minimum and maximum values of the random interval of the tabu structure. These values are the same for all different approaches.

For each test instance an initial tour is constructed using the TSB-heuristic. Since the TSB-heuristic is dependent of random numbers, for each test instance and for some approaches, N initial tours are individually constructed for each threshold or parameter setting as well. The selected initial tour is the tour with the highest score in the pool of N initial tours. In our case study we use $N = 20$ for the test instances of Category 1, for the test instance of Category 2 we use $N = 30$, while for the test instances of Category 3 $N = 50$ is used.

The Offline Methods

Also the different offline methods, which are presented in Chapter 4 and 5, require some parameter settings. Both the CCP-model and the TSOPTW are dependent of some thresholds or predefined uncertainty levels. Since the values of these thresholds have a large influence on the initial tour and therefore also on the final route, we use different values of uncertainty in this case study. In all variants of the CCP-model we need to define the threshold α, which represents the probability that the depot is reached within the available fuel capacity. This value is chosen equal to cumulative density of the maximum limit in the assumed normal distribution of the fuel usage on arc (i, j), which is

$$\alpha = P(q_{ij} \leq \mu_{ij} + \mathfrak{s}_{ij})$$

where $q_{ij} \sim N(\mu_{ij}, \sigma_{ij})$, introduced in Section 9.2 and $\mathfrak{s}_{ij}$ is the maximum deviation of the realized fuel usage of its average. Note that this probability is close to one.

The first variant of the CCP-model, which is the variant with the deadline and waiting time constraint, requires also the settings of the thresholds γ and δ. Threshold γ represents the minimum probability that the UAV does not have to wait for a target, while the minimum probability that the UAV departs from a target before its deadline is given by δ. Since the expected waiting time is not taken into account in this first variant, we restrict the probability to wait for a target to 0.5. Therefore, the value of threshold γ is in our case study equal to 0.5. For threshold δ we use different values to investigate the influence of this parameter on the final route. We use in our case study $\delta \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

The second variant of the CCP-model has a time window constraint, which means that the probability that the UAV arrives at and departs from a target within its time window should be at least as high as the fixed threshold β. For this threshold we use the same values as used for threshold δ in the first variant.

The third variant of the CCP-model has only a deadline constraint, but the expected waiting time is taken into account. For threshold δ, which represents the minimum probability the UAB departs a target before its deadline, is the same as in the first variant and therefore also the same values are used. Parameter κ is the fraction of the waiting time which is included in the objective function to minimize the total expected waiting time. Since the realized waiting time does not affect the realization of the objective during the execution of the case study, this parameter is chosen equal to zero. The time the UAV should wait for a target is already lost time. This is due to the fact that spending fuel on waiting does not have any value for the objective function. In this third variant also two parameters are introduced to compensate the error which is made by considering the expected waiting time. The value of these parameters, which are θ and λ, is in our case study equal to one.

In the TSOPTW we need to define an uncertainty level α. Note that this α represents a different probability than the threshold α in the first variant of the CCP-model. This level of uncertainty represents the minimum probability a target could be reached before its deadline if it should not be skipped in the final route. For this parameter we use in our case study the same values as we use for threshold δ in the first and third variant of the CCP-model. Note that this could result in six different initial tours, one for each value of α.

For the POPTW we have introduced four different penalty functions to determine the incurred penalty for a late departure from a target. All these penalty functions require some parameter settings. In our case study we assume that the time windows of the targets are hard, which means that recording a target outside its time window yields nothing. Therefore, we use for parameter pf in Penalty function 1 a value of 50, which results in a nonpositive yield for late recording of a target. For parameter pf_i in Penalty function 2 we use the value of 1, through this the yield of a late recording is equal to zero. In both smooth penalty functions a scale parameter is included. For the linear smooth penalty function we set scale parameter b equal to 5, while for the quadratic smooth penalty function scale parameter c is chosen to be equal to 2.5.

Furthermore, in the construction of the initial tour of both the TSOPTW and the POPTW a sample set of $|\Omega|$ scenarios is required to determine the score of a solution. We use for both approaches the same set of scenarios. In this case study during the construction of the initial tour of both recourse models a sample set of 1000 scenarios is used, therefore $|\Omega| = 1000$.

The Online Strategies

Also two of the online strategies that were introduced in Chapter 7 require some parameter settings. First, the recourse action is developed to prevent that the UAV departs from a lot of targets later than the deadline of the time window, which yields no information value. The decision to skip a target is based on a predefined uncertainty level α. We use in our case study uncertainty level $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, which are the same values as for threshold δ in the CCP-model and as for the level of uncertainty in the TSOPTW.

10.1.2 One-Step-Ahead Routing

Also the Repeated Tabu-Search Based Heuristic Approach presented in Section 8.2 requires some parameter settings. In this thesis we have executed at each step of the RTSBA a Tabu-Search Based heuristic with a fixed number of iterations $Tmax$ equal to 50 in which 75 candidate solutions are constructed. Consequently, $Nmax = 75$. The upper and lower bound of the random interval of the tabu structure are set to $\underline{\theta} = 5$ and $\overline{\theta} = 10$ in this RTSBA.

10.2 Performance Measures

To compare the different approaches which are used to construct an initial tour and the different adaptive routing strategies which are used to adjust the initial tour to the fuel realizations, for each combination the constructed initial tour and the strategy are executed for S scenarios. In this case study we use $S = 10000$. Based on these executions we can calculate some performance measures.

The first performance measure we use to evaluate the different approaches introduced is the average realized profit. This value is calculated by taking the mean over the realized profits of all S executed scenarios. This measure is an estimation of the expected realized profit for the used approach and strategy.

The second measure is the absolute gap between the average realized profit and the sum of the profits of the targets in the initial tour, while the third measure is the percentage gap between the average realized profit and the planned profit in the initial tour. These two measures indicate both how close the expected realized profit is to the planned profit.

The average number of skipped targets is the fourth performance measure. The value of this measure is calculated by taking the mean over the differences between the number of targets in the initial tour and the number of visited targets in the final route of all S executed scenarios. Since skipping a target means that the initial tour has to be adjusted, it is preferred that the average number of skipped targets during the executions of the tour is small. This performance measure is used, because it is an indication of the endurance of the initial constructed tour. In military settings a predictable route of the flight could be desirable.

The fifth and last performance measure we use to evaluate the different approaches is the average percentage of the Profit by Complete Information (PCI). The PCI could be calculated for each scenario by solving a deterministic OPTW where the fuel consumption on the flight paths f_{ij} is equal to the fuel realizations of that scenario. The objective value of this OPTW is the total profit which could be gathered in this scenario if all fuel realizations were available before the flight. Therefore, the average realized percentage of this value should be large. Note that the deterministic OPTW is proven to be NP-hard [8]. Therefore, due to the very large calculation time this performance measure is only determined for the test instance in Category 1.

For the One-Step-Ahead Routing approaches only the first and the fifth performance measures are used to evaluate these approaches. In OSAR there is no initial tour planned before the flight, therefore the difference between the planned tour and the executed tour cannot be determined.

11. Evaluation of the Hybrid Approach

In this chapter we will present the results of the hybrid approach of the SOPTW. We will compare the different methods to construct an initial tour. In the second part of this thesis we have introduced 9 different methods to construct an initial tour. The first was the optimal solution of the deterministic OPTW. Using the CCP-model three different initial tours could be constructed. Besides these methods, also the TSOPTW could be used to construct an initial tour. Finally, based on the different penalty functions, the last four initial tours can be constructed using the POPTW.

Preliminary experimental testing has shown that both the first variant of the CCP-model with the deadline and waiting time constraint and the second variant with the time window constraint provide poor performances compared to the third version of the CCP-model. The main reason for these undesirable outcomes is that the expected waiting time is not taken into account, while the execution of the initial tour is affected by waiting time. The third variant considers the expected waiting time and therefore this variant outperforms the first two variants of the CCP-model. Besides that, for some test instances of the first and second category it is not possible to construct an initial tour which is feasible for the second variant of the CCP-model with a certain value for threshold β. This is due to the fact that in these instances there is no target which can be reached after its earliest time with a minimum probability larger than 0.5. Consequently, for these values of threshold β this variant cannot be used. For these two reasons we will only present the results of the third variant of the CCP-model in this chapter.

Furthermore, due to the parameter settings of the first penalty function of the POPTW, the yield of a late recording of a target is zero or in some cases even negative. For the given parameter settings of the second penalty function, the yield of a late recording is always equal to zero. This could result in a shorter initial tour for the first penalty function, since the penalties of some targets are higher than when Penalty function 2 is applied. Therefore, during the execution, the realized profit of the initial tour of the second penalty function is the same or more than the realization of the first penalty function. Except this, since we have assumed that the time windows are hard in our case study, the yield of a late recording is always equal to zero. In Penalty function 2 this is also the case, therefore the results of the POPTW combined with Penalty function 1 are left out in this chapter.

For the POPTW we also consider two different smooth penalty functions. By these penalty functions the penalty costs are dependent on the late period, which means that the yield of a late recording is not always equal to zero. Since the yield of a late recording could be positive, the time windows are soft in this case. However, during the execution of the initial tour hard time windows are considered. For the linear penalty function the penalty for late recording increases less rapidly than the penalty of the quadratic penalty function. This results in a better approximation of the hard time windows which we consider during the execution for the linear penalty function. Therefore, the results of the POPTW combined with the quadratic smooth penalty function are also left out in this chapter.

We will combine the remaining five methods with different adaptive routing strategies. In the first section we will evaluate the different offline methods combined with the recourse action, while in the second section the results of the execution of the initial tours adjusted by the Alternative Recourse Action are presented.

11.1 Recourse Action

In this section we will present the results of the different methods combined with the recourse action. The complete table of remaining results can be found in Table A.1 of the appendix. To discuss the results, we will use an illustrative instance, since all test instances of all categories follow approximately the same pattern. Table 11.1 shows the results of Test instance c107 of Category 2 for different levels of uncertainty α: the minimum probability level to continue to the next target, applied in the recourse action. Note that these uncertainty levels are also used in the construction of the initial tour of both the CCP-model and the TSOPTW. For the five considered offline methods the values of the introduced performance measures are given. In the first column the sum of the profits of the targets in the initial tour is displayed. The profit which is realized by evaluation of this initial tour combined with the recourse action is given in the second column. Column 3 and 4 show the gap between the planned profit and the realized profit. The third column shows the absolute gap between those two values, while in the fourth column the percentage gap is displayed. Column 5 contains the average number of skipped targets of the initial tour due to the recourse action.

The initial tour of the deterministic OPTW is the same for all levels of uncertainty, since this method does not take uncertainty into account. Therefore, also the planned profit is the same for all uncertainty levels. The average realized profit is the largest for a level of uncertainty of 0.6. Both smaller and larger uncertainty levels provide a lower average profit during the execution. When the level of uncertainty is equal to 1, the average realized profit is the lowest. This is caused by the fact that at this level, the number of targets that should be skipped due to the recourse action is the highest out of all levels of alpha. The average absolute gap is between 37 and 45 units of profit, which corresponds to a percentage gap between 10 and 12 percent.

The constructed initial tours of the CCP-models are shorter than the optimal deterministic tour. Therefore, the planned profit of these tours is also less than the planned profit of the deterministic tour. Note that in the optimal deterministic tour all targets can be reached with probability 0.5, since we have assumed that the deterministic fuel consumption is equal to the means of the truncated normal distributions. However, due to the total fuel constraint of the CCP-model, this optimal deterministic tour can be not feasible for the CCP-model with waiting time. Since in the CCP-model the uncertainty of the whole tour is considered, the initial tour which is feasible for uncertainty level 1 is shorter than the other initial tours. However, the gap between the realized profit and the planned profit is smaller for this level of uncertainty compared to both the absolute and the percentage gap of the other uncertainty levels. The realized profit is lower for larger levels of uncertainty caused by the larger number of targets of the initial tour that should be skipped due to the recourse action. With respect to the realized profit, the CCP-model outperforms the deterministic OP approach. Furthermore, both the average gap between the planned and the realized profit and the average number of skipped targets are smaller than the gaps of the first considered method.

For the TSOPTW the constructed initial tours are longer than all other initial tours and therefore also the planned profit is larger. This is due to the fact that in the construction of the initial tour of the TSOPTW the recourse action is already taken into account, which means that the possibility to skip targets is considered. Consequently, both the average number of skipped targets and the average gap are much larger than for the other methods. However, for all levels of uncertainty the average realized profit for the TSOPTW is larger than the average realized profit for all other methods.

Recourse Action

c107	Deterministic OPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	370	332.308	37.692	10.187	0.8525
0.6	370	332.884	37.116	10.031	0.8768
0.7	370	332.840	37.160	10.043	0.9006
0.8	370	332.250	37.750	10.203	0.9308
0.9	370	330.790	39.210	10.597	0.9668
1.0	370	324.796	45.204	12.217	1.0634
	CCP-model				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	350	347.839	2.161	0.617	0.0208
0.6	350	347.695	2.305	0.659	0.0265
0.7	350	347.345	2.655	0.759	0.0377
0.8	350	346.667	3.333	0.952	0.0560
0.9	350	344.682	5.318	1.519	0.1036
1.0	330	329.382	0.618	0.187	0.0218
	TSOPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	380	348.700	31.300	8.237	1.8805
0.6	410	349.304	60.696	14.804	2.8834
0.7	410	349.765	60.235	14.691	2.8829
0.8	410	349.951	60.049	14.646	2.8848
0.9	410	349.406	60.594	14.779	2.8915
1.0	470	340.113	129.887	27.636	7.0113
	POPTW - Fixed Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	350	347.839	2.161	0.617	0.0208
0.6	350	347.695	2.305	0.659	0.0265
0.7	350	347.345	2.655	0.759	0.0377
0.8	350	346.667	3.333	0.952	0.0560
0.9	350	344.682	5.318	1.519	0.1036
1.0	350	329.252	20.748	5.928	0.4214
	POPTW - Smooth Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	350	347.812	2.188	0.625	0.0243
0.6	350	347.650	2.350	0.671	0.0301
0.7	350	347.282	2.718	0.777	0.0413
0.8	350	346.572	3.428	0.979	0.0606
0.9	350	344.537	5.463	1.561	0.1096
1.0	350	329.056	20.944	5.984	0.4280

Table 11.1: The results of Instance c107 for different methods combined with the recourse action.

The constructed initial tours of both variants of the POPTW have a lower planned profit than the sum of the profits of the targets in the optimal deterministic tour. This profit is the same for all levels of uncertainty, because the values of α are not taken into consideration in the POPTW. For most of the levels of uncertainty, the values of the performance measures of both variants of the POPTW are comparable with the results of the CCP-model, except for the case when the uncertainty level is equal to 1. In that case both the gap between the planned profit and the realization and the average number of skipped targets is larger than for the CCP-model. This is due to the fact that the initial tour of the CCP-model is adjusted to the uncertainty level, while the initial tours of the POPTW are the same for all levels of uncertainty.

11.2 Alternative Recourse Action

The execution of the initial tours constructed using the five considered methods combined with the alternative recourse action is evaluated in this section. Recall that the alternative recourse action prescribes that all planned targets are visited as long as the UAV is still able to return to the depot in worst case. Contrary to the recourse action, in none of the offline methods the alternative recourse action is taken into account. Furthermore, during the execution the flight has to be adjusted at most one time, since only targets at the end of the tour might be skipped.

The levels of uncertainty which are taken into consideration in the construction of the initial tours of the CCP-model and the TSOPTW are not applied in the alternative recourse action. Therefore, we consider for each of the five different offline methods just one of the available initial tours. For the deterministic OPTW and both variants of POPTW only one initial tour is constructed for each test instance. However, for the TSOPTW and the CCP-model an initial tour is available for each of the six different levels of uncertainty. In this section we evaluate for the TSOPTW the initial tour with the smallest average number of skipped targets due to the recourse action in Section 11.1. This selected tour had the least adjustments during the flight. Consequently, when the possibility to skip targets due to their time windows is left out, the realized profit of this tour will be larger than the realized profit of the other initial TSOPTW-tours associated to other levels of alpha. For the TSOPTW the initial tour which is constructed based on the uncertainty level equal to 0.5 has for all test instances the smallest average number of skipped targets out of all initial tours. In the construction of the initial tour of the CCP-model, it is assumed that the tour cannot be adjusted during the flight. Due to the deadline constraint the tours are longer for smaller threshold. Therefore, we select for each test instance the initial tour which is based on a threshold equal to 0.5.

In Table 11.2 the results for two illustrative test instances of the execution of the initial tours combined with the alternative recourse action are given. The results for the other instances can be found in Table A.2 of the appendix. The other test instances of Category 1 and 2 show similar results as illustrative case c105 and the results of the other instances of Category 3 are similar to the results of illustrative case c103.

For Test instance c105 of Category 1 we see that the realized profit of the TSOPTW is lower than the realized profit of the other methods. The time windows are small in this test instance and therefore the probability to depart from a target after its deadline is larger. This probability is increased by the fact that the possibility to skip targets due to their time windows is considered during the construction of the initial tour, but this possibility does not exists in the alternative recourse action. In contrast to Test instance c105, half of the time windows of Test instance c103 are very large. The probability to reach a target within its deadline is therefore much larger, even when it is not possible to skip targets during the flight. Consequently, the realized profit of the TSOPTW is the largest for test instances with large time windows.

For both instances the deterministic optimal tour gathered on average fewer profit during the execution than the CCP-model and both POPTW variants. This means that when the stochasticity of the fuel consumption is taken into account, this will result in a larger realized profit. For Test instance c103 the average gathered profit during the execution of the initial tour of the CCP-model is larger than the realized profit of both POPTW tours, while for the small time windows of c105 the realized profit of the CCP-model is equal to the realization of the POPTW with a fixed penalty. The POPTW with a smooth penalty function has in both test instances a lower realized profit than the CCP-model and the POPTW with a fixed penalty.

The gap between the planned and the realized profit of the deterministic OPTW and the TSOPTW is much larger than the gap of the other three methods. For Test instance c105 the realized profit of the TSOPTW is only 48% of the planned profit. For the very large time windows of Test instance c103 the gap between the realized and planned profit is the smallest for the POPTW with a fixed penalty, while for the small time windows of Instance c105 this gap is the smallest for the POPTW with a smooth penalty function.

Also the average number of skipped targets during the execution of the initial tour of the TSOPTW is much larger than for the other models. For both instances the method with the smallest gap is also the method with the lowest average number of skipped targets. Note that in Test instance c105 there is no target skipped during the execution of the initial tour of the POPTW with a smooth penalty function, but the realized profit is not equal to the planned profit. This is due to the fact that some targets are recorded outside their time window and therefore the yield of the recording was equal to zero.

c103	**Alternative Recourse Action**					
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	390	331.434	58.566	15.017	0.9428	-
CCP-model (α=0.5)	370	363.897	6.103	1.649	0.1175	-
TSOPTW (α=0.5)	430	376.521	53.479	12.437	2.5613	-
Fixed Penalty	360	359.980	0.020	0.006	0.0005	-
Smooth Penalty	360	355.338	4.662	1.295	0.1554	-
c105						
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	340	291.819	48.181	14.171	0.0704	0.8671
CCP-model (α=0.5)	330	327.930	2.070	0.627	0.0045	0.9488
TSOPTW (α=0.5)	420	199.885	220.115	52.408	5.0832	0.5788
Fixed Penalty	330	327.930	2.070	0.627	0.0045	0.9488
Smooth Penalty	320	319.794	0.206	0.064	0.0000	0.9261

Table 11.2: The results of Test instances c103 and c105 for different methods combined with the alternative recourse action.

12. Evaluation of the One-Step-Ahead Routing Approach

In this chapter we will evaluate the different One-Step-Ahead-Routing Approaches. In Chapter 8 we have introduced four different OSAR approaches. The first two approaches are Best-Neighbor approaches, where the first is the Deterministic Best Neighbor Approach (DBNA) which does not take uncertainty into account and the second is the Stochastic Best Neighbor Approach (SBNA) which considers the stochasticity of the fuel consumption. Both these approaches do not take the remainder of the flight into consideration in the determination of the location to visit next. The other two approaches on the other hand, do consider the remainder of the flight in the determination of the next location by constructing a whole route for the remainder of the flight and selecting the first location in this route as next location. These two approaches execute repeatedly a tabu-search based heuristic. The first Repeated Tabu-Search-Based heuristic Approach is the Deterministic RTSBA, where uncertainty is ignored. In the constructing of a route for the remainder of the flight, the Stochastic RTSBA considers the stochasticity of the fuel usages.

We have evaluated these four approaches for the same S scenarios as the scenarios which we used for the evaluation of the five offline methods combined with the recourse actions. In OSAR there is no initial tour constructed before the flight which should be adaptive during the flight. After recording a target, the next location of the flight is determined using an OSAR approach. Therefore, only the realized profit and the percentage of the PCI can be used as performance measures in this case.

In Table 12.1 contains the realized profit and the percentage of the PCI for the four OSAR approaches: DBNA, SBNA, RDTSBA and RSTSBA.

For the Best-Neighbor approaches, we see that the realized profits of the deterministic variant are less than the realized profit of the stochastic variant for the test instances with small or medium time windows. The realized profit of the stochastic variant of the test instances with very large time windows is smaller than the realization of the deterministic variant. Taking into account the stochasticity of the fuel usages is more important for small time windows, since the probability to arrive at the target after its deadline is much larger than for large time windows. In instances with very large time windows it is less important to take the uncertainty into account in the

	DBNA		**SBNA**		**RDTSBA**		**RSTSBA**	
	Realized Profit	% of the PCI	Realized Profit	% of the PCI	Realized Profit	% of the PCI	Realized Profit	% of the PCI
c101	299.274	0.9273	299.376	0.9231	307.124	0.9492	309.560	0.9580
c102	322.418	-	320.628	-	335.307	-	338.359	-
c103	375.994	-	374.635	-	376.648	-	376.362	-
c104	388.160	-	387.764	-	388.094	-	389.238	-
c105	317.115	0.9353	317.981	0.9254	324.719	0.9472	328.276	0.9548
c106	315.309	0.8989	322.225	0.9183	320.017	0.9146	329.760	0.9433
c107	312.755	-	317.922	-	336.721	-	335.975	-
c108	327.456	-	328.156	-	339.608	-	341.248	-
c109	343.534	-	345.773	-	354.553	-	354.029	-

Table 12.1: The results of the One-Step-Ahead Routing Approaches for all considered test instances.

determination of the next location. Furthermore, in the SBNA the targets which can be reached before their deadline with a probability equal to one are overrated in the determination of next target. In case there are a lot of targets with the same profit, which have a probability to be reached before their deadline equal to one, the nearest of these targets is selected as the next location. This selection could have a negative impact on the realized profit.

For the most test instances the realized profits of the repeated TSB approaches are for both variants significant larger than the realizations of the profit of the BNA. We can see that the realized profit of the stochastic variant is for the most instances larger than the deterministic variant. However, for a few test instances the total average gathered profit of the stochastic variant is smaller than the realizations of the profit for the deterministic variant, but these differences are small. The improvement of the realized profit by considering of the stochasticity of the fuel usages is the largest by the test instances with the smallest time windows. This is due to the fact that the probability a small time window is reached after its deadline by higher fuel usages is much larger than when the width of the time windows is larger. Therefore, taking into account the stochasticity of the fuel consumption is more recommended in these test instances.

Compared to the results of the hybrid approach, presented in Chapter 11, we can see that the realized profits of the OSAR approaches are smaller than the highest realized profits of the hybrid approach for all test instances. These differences are the smallest for the test instances of Category 1, where adjusting the flight to the realizations of the fuel usages is the most profitable because of the small time windows.

Part V

Conclusions

In this part we will summarize our research done in this thesis. In Chapter 13 an overview of advantages and disadvantages of the presented methods and strategies is given, while a summary and a short conclusion are provided in Chapter 14. This part will end with some ideas for further research in Chapter 15.

13. Overview Solution Strategies

The diagram below contains the advantages and disadvantages of the methods and approaches.

Hybrid Approach			
Offline Methods			
Approach/Method	**Advantages**	**Disadvantages**	**Section**
Deterministic OPTW	• "Optimal" solution can be found for most instances	• Uncertainty not taken into account	4.1
CCP-model – Deadline and Waiting Constraint	• Maximal probability to wait for a target is fixed	• Not feasible for some test instances • Expected waiting time not included	4.3.2
CCP-model – Time Window Constraint	• Minimal probability to reach within time window is fixed	• Not feasible for some test instances • Expected waiting time not included	4.3.3
CCP-model – Waiting Time	• Expected waiting time included • Small gap between planned and realized profit	• No exact distribution of sum of fuel consumption and waiting time can be determined	4.3.4
TSOPTW	• Recourse action taken into account • Less specific assumptions on probability distribution required • Largest realized profit	• Large average number of skipped targets during the execution • Large gap between planned and realized profit	5.2
POPTW – fixed penalty function	• Less specific assumptions on probability distribution required • Both soft or hard time windows can be assumed • Small gap between planned and realized profit	• Loss in potential profit due to disregarding possibility of skipping targets	5.3
POPTW – smooth penalty function	• Less specific assumptions on probability distribution required • Small gap between planned and realized profit	• Assumed soft time windows • Loss in potential profit due to disregarding possibility of skipping targets	5.3
Online Strategies			
Recourse Action	• Optimized for time windows	• Less predictable first part of the flight by skipping intermediate targets during the flight.	7.1
Alternative Recourse Action	• More predictable first part of the flight by skipping only targets at the end of the flight.	• Not optimized for time windows	7.2
OSAR approach			
Best Neighbour approach	• Fast determination of next location	• Local focus on only the best next location	8.1
Repeated TSB approach	• Also targets close to the next location taken into account	• Longer determination of next location	8.2

14. Summary and Conclusion

In this thesis we have addressed the UAV-mission planning problem with time windows and stochastic fuel consumption. In the area of operations, several targets are identified, which each have their own information value (represented by a profit value in our models). Since the fuel capacity of the UAV is fixed, during a mission only a subset of these targets can be recorded. Furthermore, the fuel usage on the flight paths between each two targets is not fixed, but only known a priori probabilistically. This means that before the flight only a probability distribution function is known for the fuel usage on each flight path. Besides that, to all targets a time windows is assigned, consisting of an earliest and a latest time. When recording takes place within the assigned time window, information is collected and a profit is gathered. After the deadline of a target there is no information to collect and therefore the gathered profit is equal to zero. If the UAV arrives at a target before the earliest time, it should wait until the target is 'open'. In this thesis we have presented methods to construct a route which maximizes the total collected information value, taking into account the stochasticity of the fuel consumption. This route is restricted by the fuel capacity of the UAV and the time windows of the targets.

We have detected that this problem can be modeled as a stochastic orienteering problem with time windows, which to the best of our knowledge has not yet been investigated in the literature. However, literature about related problems, such as the stochastic vehicle routing problem with time windows, can be found in literature.

Based on this found literature, we have decided to focus on two different approaches to solve the addressed problem: a hybrid approach and a One-Step-Ahead Routing approach. In the hybrid approach an initial route is constructed before the flight, which can be adjusted during the flight to the fuel realizations. The first method to construct an initial tour is a chance-constrained programming model, which prescribes a minimum probability that a target in the tour should be reached before its deadline. The minimum probability that the depot should be reached before the fuel capacity is completely used is also set in this model. The second and the third offline methods are both stochastic programming models with recourse, which means that a tour is constructed in the first stage, while in the second case the recourse costs of this tour are determined. The first recourse model is the two-stage orienteering problem with time windows. In this model a recourse action is applied to prevent the UAV both for running out of fuel and for missing a lot of time windows. In the second recourse model a penalty is incurred for late recordings and for a late arrival at the depot. In this thesis we have introduced four different penalty functions.

To construct an initial tour using one of these methods, a tabu-search based heuristic is presented. In each iteration of the heuristic a fixed number of candidate solutions are constructed by applying one of the six different neighbor operators to the current solution. The candidate solution with the highest score is selected as new current solution. This process is repeated until the maximum number of iterations is reached. The solution with the best overall score is the output of the heuristic.

To adjust the initial tour to the realizations of the fuel consumption during execution of the tour, two recourse actions are introduced. The first recourse action is the same recourse action which is considered in the TSOPTW, while the second is based on a problem without time windows.

The second approach to our UAV-mission planning problem is the One-Step-Ahead Routing (OSAR) approach. In this approach no initial tour is constructed before the flight. At the departure from a target the next location is determined by an OSAR approach. In this thesis we have presented four different of such approaches. The first two approaches are best neighbor approaches, where in the first the uncertainty of the fuel usages is not taken into account, while in the second variant the stochasticity of the fuel consumption is considered. In the third and fourth approach not only the best next location is determined but also the best direction of the next location. This is done by constructing an entire route for the remaining part of the flight using a variant of the Tabu-Search based heuristic. From this route only the first location is considered. In the third approach the uncertainty of the fuel usages is not considered, while this is done in the fourth approach.

We have a case study to compare these different approaches and the methods within these approaches. In this case study we have extended some of the deterministic orienteering problem with time window test instances from literature to stochastic test instances. We have presented the results of the case study using five performance measures.

From this case study we can overall conclude that taking into account the uncertainty of the fuel consumption leads to better results. If the initial tours are adjusted during the flight using the recourse action, the realizations of the gathered profit are the largest for the TSOPTW. Also the initial tour of the CCP-model and the POPTW combined with the recourse action gather more profit during the execution than the deterministic optimal tour for the most test instances. Besides that, for the CCP-model the average gap between the planned profit in the initial tour and the realized profit during the execution and the average number of skipped targets are both the smallest out of all models. However, during the execution of the TSOPTW initial tour the largest average number of targets is skipped and therefore also the gap between the planned and the realized profit is the largest for the TSOPTW.

If the alternative recourse action is used to adjust the initial tour during the flight, the TSOPTW gathers again the most profit during the execution of the flight for the instances with very large time windows. For the test instances with smaller time windows the TSOPTW performs poorly, while the CCP-model and the POPTW with a fixed penalty give the best results. Also in this case, realization of the profit of the execution of the deterministic optimal tour is fewer than the realizations of the other stochastic programming methods.

Furthermore, also for the OSAR approaches we can conclude that taking into account the stochasticity of the fuel consumption increases the average total gathered profit during the flight for most test instances. However, for the BNA the realization of the profit of the test instance with very large time windows is larger when the uncertainty of the fuel usages is not considered. Furthermore, we can also conclude that considering the whole route of the remaining part of the flight leads to larger realizations of the total gathered profit.

Overall, we can conclude that the hybrid approach leads to better realizations of the gathered profit for the most test instances. Moreover, this approach is suitable to the military setting, since tours are more predictable. That is, planned tour can only be changed by skipping targets, while by the OSAR approach the next location is unknown until the recording of the current location is finished.

15. Further Research

In this research we have addressed the problem situation in which the location of the targets is assumed to be fixed and known beforehand. Further research could focus on target locations that change during execution of the tour. This is for example the case when the targets represent individuals or groups of people that move from one place to the other. This uncertainty in the target locations will require an approach that deals with this dynamic situation, probably in a different way than the methods we have introduced in this thesis. However, if the movement of the targets is limited, our OSAR approach might still perform well since it takes into account all current available information at each step of the determination of the next location.

A similar reasoning applies to the situation where new targets appear during the flight. In this situation an initial tour could be used, based on target information that is available before the flight. During the flight the route could be reoptimized based on the current information on both fuel usages and new targets. Since the OSAR can also take this current information into account, this approach could be used as well. Depending on the expected number of new targets that appear during the flight, additional approaches might be required to take these dynamics into account.

Recall that we have assumed that no profit can be gathered outside the time window. Usually this is suitable to the military setting. However, in some cases part of profit might still be obtainable when arrive some time before or after the time window. In that case, we expect our penalized recourse models to perform well.

Part

Appendices

This part includes the list of references and the appendix with the results of the remaining test instances for both the recourse action and the alternative recourse action.

References

[1] L. Evers, T. Dollevoet, A.I. Barros, H. Monsuur. Robust UAV mission planning. *Econometric Institute Report,* EI 2011-07, 1-17, 2011.

[2] I.-M. Chao, B.L. Golden, E.A. Wasil. Theory and methodology – the team orienteering problem. *European Journal of Operational Research*, 88(3), 464-474, 1996.

[3] S. Lorini, J.-Y. Potvin, N. Zufferey. Online vehicle routing and scheduling with dynamic travel times. *Computers and Operations Research,* 38(7), 1086-1090, 2011.

[4] A. Garcia, O. Arbelaitz, P. Vansteenwegen, W. Souffriau, M.T. Linaza. Hybrid approach for the public transportation time dependent orienteering problem with time windows. *Lecture Notes in Computer Science*, 6077, 151-158, 2010.

[5] S. Gao and H. Huang. Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks. *Transportation Research Part C: Emerging Technologies,* 21(1), 196-213, 2012.

[6] T. Tsiligrides. Heuristic methods applied to orienteering. *Journal of the Operations Research Society*, 35(9), 797-803, 1984.

[7] B. Golden, Q. Wang, R. Vohra. The orienteering problem. *Naval Research Logistics* 34(3), 307-318, 1987.

[8] M.G. Kantor and M.B. Rosenwein. The orienteering problem with time windows. *Journal of the Operations Research Society*, 43(6), 629-635, 1992.

[9] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, D. Van Oudheusden. Iterated local search for the team orienteering problem with time windows. *Computers and Operations Research*, 36(12), 3281-3290, 2009.

[10] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1), 1-10, 2011.

[11] S.Y. Teng, H.L. Ong, H.C. Huang. An integer L-shaped algorithm for the time-constrained traveling salesman problem with stochastic travel times and service times. *Asia-pacific Journal of Operational Research*, 21(2), 241-257, 2004.

[12] H. Tang and E. Miller-Hooks. Algorithms for a stochastic selective travelling salesperson problem. *Journal of the Operational Research Society,* 56(4), 439-452, 2005.

[13] A.M. Campbell, M. Gendreau, B.W. Thomas. The orienteering problem with stochastic travel and service times. *Annals of Operations Research,* 186(1), 61-81, 2011.

[14] L. Evers, K. Glorie, S. van der Ster, A. Barros, H. Monsuur. The orienteering problem under uncertainty – Stochastic programming and robust optimization compared. Submitted to *Computers and Operations Research,* 2012.

[15] J.-Y. Potvin and J.-M. Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research,* 66(3), 331-340, 1993.

[16] F.V. Fomin and A. Lingas. Approximation algorithms for time-dependent orienteering. *Information Processing Letters,* 83(2), 57-62, 2002.

[17] X. Wang and L. Tang. A hybrid meta-heuristic for the prize-collecting single machine scheduling problem with sequence-dependent setup times. *Computers and Operations Research,* 37(9), 1624-1640, 2010.

[18] A. Del Bimbo and F. Pernici. Saccades planning with kinetic tsp for distant targets identification. *IEE Conference Publication,* 11033, 145-149, 2005.

[19] J. Li, Q. Wu, X. Li, D. Zhu. Study on the time-dependent orienteering problem*. International Conference on E-Product E-Service and E-Entertainment*, art. no. 5660232, 2010.

[20] J. Li. Research on team orienteering problem with dynamic travel times. *Journal of Software,* 7(2), 249-255, 2012.

[21] A. Garcia, P. Vansteenwegen, O. Arbelaitz, W. Souffriau, M.T. Linaza. Integrating public transportation in personalised electronic tourist guides. *Computers and Operations Research*, doi:10.1016/j.cor.2011.03.020, 2011.

[22] E. Baker. Vehicle routing with time window constraints. *Logistic and Transportation Review,* 18(4), 385-401, 1982.

[23] J. Desrosiers, F. Soumis, M. Desrochers, M. Sauve. Vehicle routing and scheduling with time windows. *Mathematical Programming Studies*, 26, 249-251, 1983.

[24] M.M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research,* 35(2), 254-265, 1987.

[25] N.A. El-Sherbeny. Vehicle routing with time windows: An overview of exact, heuristic and meta-heuristic methods. *Journal of King Saud University – Science,* 22(3), 123-131, 2010.

[26] G. Laporte, F. Louveaux, H. Mercure. The vehicle routing problem with stochastic travel times. *Transportation Science,* 26(3), 161-170, 1992.

[27] T.V. Woensel, L. Kerbache, H. Peremans, N. Vandaele. A queuing framework for routing problems with time-dependent travel times. *Journal of Mathematical Modelling and Algorithms,* 6(1), 151-173, 2007.

[28] R.D. Connors and A. Sumalee. A network equilibrium model with travellers' perception of stochastic travel times. *Transportation Research Part B: Methodological,* 43(6), 614-624, 2009.

[29] J.C.F. Wong, J.M.Y. Leung, C.H. Cheng. On a vehicle routing problem with time windows and stochastic travel times: model, algorithms and heuristics. *Technical Report, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong,* SEEM2003-03, 2003.

[30] N. Ando and E. Taniguchi. Travel time reliability in vehicle routing and scheduling with time windows. *Network and Spatial Economics*, 6(3-4), 293-311, 2006.

[31] J. Gao. Model and algorithm of vehicle routing problem with time windows in stochastic traffic network. *International Conference on Logistics Systems and Intelligent Management,* art. no. 5461065, 2010.

[32] X. Li, P. Tian, S.C.H. Leung. Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm. *International Journal of Production Economics,* 125(1), 137-145, 2010.

[33] C.I. Hsu, S.F. Hung, H.C. Li. Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering,* 80(2), 465-475, 2006.

[34] R.A. Russell, T.L. Urban. Vehicle routing with soft time windows and Erlang travel times. *Journal of the Operational Research Society*, 59(9), 1220-1228, 2008.

[35] D. Taş, N. Dellaert, T. van Woensel, T. de Kok. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers and Operations Research,* 40(1), 214-224, 2013.

[36] A. Charnes and W.W. Cooper. Chance-Constrained Programming. *Management Science*, 6(1), 73-79, 1959.

[37] A.M. Campbell and B.W. Thomas. Probabilistic Traveling Salesman Problem with Deadlines. *Transportation Science*, 42(1), 1-21, 2008.

[38] C.E. Miller, A.W. Tucker, R.A. Zemlin. Integer Programming Formulation of Traveling Salesman Problems. Journal of the ACM, 7(4), 326-329, 1960.

[39] L. Galli and A.N. Letchford. Reformulating Mixed-Integer Quadratically Constrained Quadratic Programs. *Working Paper, The Department of Management Science, Lancaster University,* 49041, 2011.

[40] G.B. Dantzig. Linear Programming under Uncertainty. *Management Science*, 1(3/4), 197-206, 1955.

[41] A. Shapiro and T. Homem-de-Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81(3), 301-325, 1998.

[42] W.-K. Mak, D.P. Morton, R.K. Wood. Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters,* 24 (1), 47-56, 1999.

[43] A.J. Kleywegt, A. Shapiro, T. Homem-De-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization,* 12(2), 479-502, 2002.

[44] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research,* 13(5), 533-549, 1986.

[45] M. Gendreau, A. Hertz, G. Laporte. A tabu search for the vehicle routing problem. *Management Science*, 40(10), 1276-1290, 1994.

[46] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part II: metaheuristics. *Transportation Science*, 39(1), 119-139, 2005.

[47] H. Tang and E. Miller-Hooks. A TABU search heuristic for the team orienteering problem. *Computers & Operations Research,* 32(6), 1379–1407,2005.

[48] C. Archetti, A. Hertz, M.G. Speranza. Metaheuristics for the team orienteering problem. *Journal of Heuristics,* 13(1), 49-76, 2007.

[49] M. Evans, N. Hastings, B. Peacock. Statistical Distributions, Chapter 40 Triangular Distribution. New York: Wiley, pp. 187-188, 2000.

Appendix

Recourse Action

c101	Deterministic OPTW					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	320	294.554	25.446	7.952	0.6436	0.9003
0.6	320	294.003	25.997	8.124	0.7363	0.9034
0.7	320	292.803	27.197	8.499	0.8387	0.8994
0.8	320	290.602	29.398	9.187	0.9778	0.8899
0.9	320	286.604	33.396	10.436	1.1830	0.8851
1.0	320	275.714	44.286	13.839	1.7116	0.8486
	CCP-model					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	310	306.955	3.045	0.982	0.0421	0.9475
0.6	310	306.615	3.385	1.092	0.0703	0.9469
0.7	310	306.049	3.951	1.275	0.1116	0.9463
0.8	300	298.506	1.494	0.498	0.0398	0.9208
0.9	300	297.871	2.129	0.710	0.0682	0.9195
1.0	240	240.000	0.000	0.000	0.0000	0.7404
	TSOPTW					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	420	309.282	110.718	26.361	5.0558	0.9538
0.6	420	309.750	110.250	26.250	5.0652	0.9567
0.7	420	310.552	109.448	26.059	5.1009	0.9697
0.8	420	310.830	109.170	25.993	5.0999	0.9619
0.9	420	308.807	111.193	26.475	5.1138	0.9509
1.0	340	297.348	42.652	12.545	3.1812	0.9157
	POPTW - Fixed Penalty					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	310	306.955	3.045	0.982	0.0421	0.9475
0.6	310	306.615	3.385	1.092	0.0703	0.9469
0.7	310	306.049	3.951	1.275	0.1116	0.9463
0.8	310	304.878	5.122	1.652	0.1885	0.9386
0.9	310	302.137	7.863	2.536	0.3404	0.9344
1.0	310	289.846	20.154	6.501	0.9357	0.8923
	POPTW - Smooth Penalty					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	300	298.845	1.155	0.385	0.0148	0.9235
0.6	300	298.789	1.211	0.404	0.0199	0.9235
0.7	300	298.691	1.309	0.436	0.0278	0.9235
0.8	300	298.506	1.494	0.498	0.0398	0.9208
0.9	300	297.871	2.129	0.710	0.0682	0.9195
1.0	300	294.044	5.956	1.985	0.2258	0.9059

Recourse Action

c102	Deterministic OPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	360	326.071	33.929	9.425	0.9001
0.6	360	326.212	33.788	9.386	0.9194
0.7	360	325.974	34.026	9.452	0.9492
0.8	360	324.923	35.077	9.744	0.9923
0.9	360	322.037	37.963	10.545	1.0679
1.0	360	310.576	49.424	13.729	1.3136
	CCP-model				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	340	329.040	10.960	3.224	0.1343
0.6	340	328.034	11.966	3.519	0.1919
0.7	340	326.444	13.556	3.987	0.2668
0.8	320	319.996	0.004	0.001	0.0002
0.9	320	319.991	0.009	0.003	0.0003
1.0	320	319.902	0.098	0.031	0.0022
	TSOPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	380	339.545	40.455	10.646	2.5934
0.6	400	339.620	60.380	15.095	3.2992
0.7	440	339.719	100.281	22.791	4.7973
0.8	510	339.526	170.474	33.426	6.8029
0.9	510	337.978	172.022	33.730	6.8392
1.0	410	335.688	74.312	18.125	4.1569
	POPTW - Fixed Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	340	331.143	8.857	2.605	0.1037
0.6	340	329.894	10.106	2.972	0.1575
0.7	340	328.016	11.984	3.525	0.2321
0.8	340	324.803	15.197	4.470	0.3388
0.9	340	318.606	21.394	6.292	0.5175
1.0	340	299.945	40.055	11.781	1.0013
	POPTW - Smooth Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	330	329.763	0.237	0.072	0.0105
0.6	330	329.764	0.236	0.072	0.0106
0.7	330	329.761	0.239	0.072	0.0107
0.8	330	329.758	0.242	0.073	0.0108
0.9	330	329.711	0.289	0.088	0.0122
1.0	330	328.632	1.368	0.415	0.0472

Recourse Action

c103	Deterministic OPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	390*	348.298	41.702	10.693	0.9946
0.6	390	349.973	40.027	10.263	0.9969
0.7	390	351.210	38.790	9.946	1.0026
0.8	390	352.132	37.868	9.710	1.0095
0.9	390	352.264	37.736	9.676	1.0336
1.0	390	339.071	50.929	13.059	1.2726
	CCP-model				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	370	366.550	3.450	0.932	0.0378
0.6	370	366.387	3.613	0.976	0.0484
0.7	370	366.024	3.976	1.075	0.0654
0.8	360	359.838	0.162	0.045	0.0054
0.9	360	359.838	0.162	0.045	0.0054
1.0	350	349.769	0.231	0.066	0.0077
	TSOPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	430	379.884	50.116	11.655	2.1089
0.6	480	380.004	99.996	20.833	4.9391
0.7	480	380.138	99.862	20.805	4.9441
0.8	480	380.188	99.812	20.794	4.9476
0.9	430	380.068	49.932	11.612	2.1152
1.0	430	379.881	50.119	11.656	2.1486
	POPTW - Fixed Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	360	359.368	0.632	0.176	0.0158
0.6	360	359.368	0.632	0.176	0.0158
0.7	360	359.368	0.632	0.176	0.0158
0.8	360	359.368	0.632	0.176	0.0158
0.9	360	359.368	0.632	0.176	0.0158
1.0	360	359.368	0.632	0.176	0.0158
	POPTW - Smooth Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	360	359.352	0.648	0.180	0.0216
0.6	360	359.352	0.648	0.180	0.0216
0.7	360	359.352	0.648	0.180	0.0216
0.8	360	359.352	0.648	0.180	0.0216
0.9	360	359.352	0.648	0.180	0.0216
1.0	360	359.352	0.648	0.180	0.0216

** Stopped after 6 hours of calculation time*

Recourse Action

c104	Deterministic OPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	410*	379.012	30.988	7.558	0.9742
0.6	410	378.858	31.142	7.596	0.9779
0.7	410	378.658	31.342	7.644	0.9820
0.8	410	378.271	31.729	7.739	0.9886
0.9	410	377.401	32.599	7.951	0.9980
1.0	410	372.901	37.099	9.049	1.0233
	CCP-model				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	380	379.076	0.924	0.243	0.0020
0.6	380	379.053	0.947	0.249	0.0028
0.7	380	378.822	1.178	0.310	0.0096
0.8	380	378.409	1.591	0.419	0.0209
0.9	380	376.874	3.126	0.823	0.0560
1.0	370	367.359	2.641	0.714	0.0689
	TSOPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	460	397.051	62.949	13.685	2.1570
0.6	460	397.051	62.949	13.685	2.1570
0.7	460	397.051	62.949	13.685	2.1570
0.8	460	397.050	62.950	13.685	2.1570
0.9	460	397.050	62.950	13.685	2.1570
1.0	460	396.961	63.039	13.704	2.1585
	POPTW - Fixed Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	380	379.079	0.921	0.242	0.0019
0.6	380	379.074	0.926	0.244	0.0021
0.7	380	378.854	1.146	0.302	0.0091
0.8	380	378.412	1.588	0.418	0.0211
0.9	380	376.868	3.132	0.824	0.0562
1.0	380	365.469	14.531	3.824	0.2907
	POPTW - Smooth Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	380	379.043	0.957	0.252	0.0025
0.6	380	379.038	0.962	0.253	0.0027
0.7	380	378.822	1.178	0.310	0.0096
0.8	380	378.389	1.611	0.424	0.0214
0.9	380	376.843	3.157	0.831	0.0566
1.0	380	365.467	14.533	3.824	0.2907

** Stopped after 6 hours of calculation time*

Recourse Action

c105	Deterministic OPTW					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	340	309.008	30.992	9.115	0.6429	0.9069
0.6	340	309.056	30.944	9.101	0.6961	0.9087
0.7	340	308.641	31.359	9.223	0.7495	0.9064
0.8	340	307.783	32.217	9.476	0.8077	0.8989
0.9	340	305.602	34.398	10.117	0.8776	0.8938
1.0	340	299.946	40.054	11.781	1.0027	0.8683
	CCP-model					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	330	327.982	2.018	0.612	0.0352	0.9497
0.6	330	327.811	2.189	0.663	0.0465	0.9497
0.7	330	327.605	2.395	0.726	0.0591	0.9477
0.8	330	327.141	2.859	0.866	0.0799	0.9459
0.9	330	326.291	3.709	1.124	0.1129	0.9438
1.0	300	300.000	0.000	0.000	0.0000	0.8683
	TSOPTW					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	420	328.662	91.338	21.747	4.9948	0.9521
0.6	420	328.690	91.310	21.740	4.9973	0.9533
0.7	420	328.728	91.272	21.731	4.9986	0.9521
0.8	460	329.224	130.776	28.430	6.0589	0.9596
0.9	460	329.358	130.642	28.400	6.0510	0.9561
1.0	430	326.234	103.766	24.132	6.0212	0.9427
	POPTW - Fixed Penalty					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	330	327.982	2.018	0.612	0.0352	0.9497
0.6	330	327.811	2.189	0.663	0.0465	0.9497
0.7	330	327.605	2.395	0.726	0.0591	0.9477
0.8	330	327.141	2.859	0.866	0.0799	0.9459
0.9	330	326.291	3.709	1.124	0.1129	0.9438
1.0	330	321.200	8.800	2.667	0.2673	0.9330
	POPTW - Smooth Penalty					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	320	319.777	0.223	0.070	0.0011	0.9261
0.6	320	319.741	0.259	0.081	0.0022	0.9261
0.7	320	319.694	0.306	0.096	0.0044	0.9250
0.8	320	319.547	0.453	0.142	0.0092	0.9250
0.9	320	319.295	0.705	0.220	0.0170	0.9202
1.0	320	316.567	3.433	1.073	0.0872	0.9108

Recourse Action

c106	Deterministic OPTW					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	340	321.621	18.379	5.406	0.4806	0.9147
0.6	340	321.135	18.865	5.549	0.5255	0.9125
0.7	340	320.177	19.823	5.830	0.5744	0.9114
0.8	340	318.321	21.679	6.376	0.6331	0.9088
0.9	340	315.378	24.622	7.242	0.7091	0.9037
1.0	340	306.690	33.310	9.797	0.8805	0.8767
	CCP-model					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	330	328.182	1.818	0.551	0.0614	0.9299
0.6	330	327.999	2.001	0.606	0.0749	0.9293
0.7	330	327.766	2.234	0.677	0.0919	0.9287
0.8	330	327.361	2.639	0.800	0.1183	0.9270
0.9	320	319.702	0.298	0.093	0.0086	0.9050
1.0	300	299.803	0.197	0.066	0.0115	0.8495
	TSOPTW					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	380	331.956	48.044	12.643	2.1182	0.9476
0.6	380	332.565	47.435	12.483	2.1284	0.9468
0.7	380	332.622	47.378	12.468	2.1419	0.9489
0.8	380	332.613	47.387	12.470	2.0789	0.9461
0.9	380	331.904	48.096	12.657	2.0762	0.9447
1.0	380	327.662	52.338	13.773	2.0743	0.9351
	POPTW - Fixed Penalty					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	330	328.182	1.818	0.551	0.0614	0.9299
0.6	330	327.999	2.001	0.606	0.0749	0.9293
0.7	330	327.766	2.234	0.677	0.0919	0.9287
0.8	330	327.361	2.639	0.800	0.1183	0.9270
0.9	330	326.684	3.316	1.005	0.1552	0.9248
1.0	330	323.432	6.568	1.990	0.2948	0.9160
	POPTW - Smooth Penalty					
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
0.5	320	319.747	0.253	0.079	0.0088	0.9061
0.6	320	319.716	0.284	0.089	0.0098	0.9052
0.7	320	319.682	0.318	0.099	0.0112	0.9052
0.8	320	319.641	0.359	0.112	0.0128	0.9052
0.9	320	319.498	0.502	0.157	0.0171	0.9033
1.0	320	318.189	1.811	0.566	0.0533	0.8956

Recourse Action

c108	Deterministic OPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	370	335.493	34.507	9.326	0.8354
0.6	370	336.023	33.977	9.183	0.8344
0.7	370	336.093	33.907	9.164	0.8343
0.8	370	335.577	34.423	9.304	0.8356
0.9	370	334.340	35.660	9.638	0.8351
1.0	370	328.757	41.243	11.147	0.8547
	CCP-model				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	350	348.776	1.224	0.350	0.0165
0.6	350	348.754	1.246	0.356	0.0161
0.7	350	348.294	1.706	0.487	0.0315
0.8	350	348.173	1.827	0.522	0.0319
0.9	350	347.031	2.969	0.848	0.0576
1.0	330	329.983	0.017	0.005	0.0006
	TSOPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	410	353.185	56.815	13.857	2.8633
0.6	410	353.431	56.569	13.797	2.8690
0.7	410	353.477	56.523	13.786	2.8752
0.8	410	353.481	56.519	13.785	2.8513
0.9	410	353.059	56.941	13.888	2.8499
1.0	500	348.384	151.616	30.323	5.9445
	POPTW - Fixed Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	350	348.838	1.162	0.332	0.0134
0.6	350	348.754	1.246	0.356	0.0161
0.7	350	348.562	1.438	0.411	0.0213
0.8	350	348.173	1.827	0.522	0.0319
0.9	350	347.031	2.969	0.848	0.0576
1.0	350	335.802	14.198	4.057	0.2857
	POPTW - Smooth Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	350	348.776	1.224	0.350	0.0165
0.6	350	348.688	1.312	0.375	0.0194
0.7	350	348.490	1.510	0.431	0.0249
0.8	350	348.104	1.896	0.542	0.0354
0.9	350	346.959	3.041	0.869	0.0612
1.0	350	335.766	14.234	4.067	0.2875

Recourse Action

c109	Deterministic OPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	380	360.186	19.814	5.214	0.7473
0.6	380	360.343	19.657	5.173	0.7473
0.7	380	360.401	19.599	5.158	0.7473
0.8	380	360.419	19.581	5.153	0.7473
0.9	380	359.720	20.280	5.337	0.7708
1.0	380	355.943	24.057	6.331	0.8902
	CCP-model				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	370	366.678	3.322	0.898	0.0774
0.6	370	366.461	3.539	0.956	0.0911
0.7	370	366.042	3.958	1.070	0.1123
0.8	370	365.432	4.568	1.235	0.1381
0.9	370	364.223	5.777	1.561	0.1829
1.0	360	358.159	1.841	0.511	0.0503
	TSOPTW				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	400	372.517	27.483	6.871	1.7347
0.6	400	372.558	27.442	6.861	1.7350
0.7	450	372.732	77.268	17.171	3.7094
0.8	450	372.821	77.179	17.151	3.7082
0.9	450	372.640	77.360	17.191	3.7058
1.0	450	370.000	80.000	17.778	3.7398
	POPTW - Fixed Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	370	366.052	3.948	1.067	0.0796
0.6	370	365.960	4.040	1.092	0.0947
0.7	370	365.736	4.264	1.152	0.1142
0.8	370	365.270	4.730	1.278	0.1380
0.9	370	364.223	5.777	1.561	0.1829
1.0	370	359.564	10.436	2.821	0.3366
	POPTW - Smooth Penalty				
alpha	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets
0.5	360	359.597	0.403	0.112	0.0055
0.6	360	359.571	0.429	0.119	0.0069
0.7	360	359.533	0.467	0.130	0.0091
0.8	360	359.438	0.562	0.156	0.0130
0.9	360	359.180	0.820	0.228	0.0211
1.0	360	357.308	2.692	0.748	0.0713

Table A.1: The results of the test instance for different methods combined with the recourse action.

c101	Alternative Recourse Action					
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	320	283.572	36.428	11.384	0.0028	0.8689
CCP-model	310	307.084	2.916	0.941	0.0000	0.9482
TSOPTW	420	192.524	227.476	54.161	5.1970	0.5942
Fixed Penalty	310	307.084	2.916	0.941	0.0000	0.9482
Smooth Penalty	300	298.800	1.200	0.400	0.0000	0.9235
c102						
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	360	324.288	35.712	9.920	0.8471	-
CCP-model	340	329.329	10.671	3.139	0.0006	-
TSOPTW	380	335.680	44.320	11.663	2.6829	-
Fixed Penalty	340	332.107	7.893	2.321	0.0139	-
Smooth Penalty	330	329.597	0.403	0.122	0.0188	-
c104						
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	410	377.948	32.052	7.818	0.9964	-
CCP-model	380	379.087	0.913	0.240	0.0016	-
TSOPTW	460	377.540	82.460	17.926	2.8115	-
Fixed Penalty	380	379.078	0.922	0.243	0.0019	-
Smooth Penalty	380	379.075	0.925	0.243	0.0015	-
c106						
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	340	310.550	29.450	8.662	0.3622	0.8896
CCP-model	330	328.399	1.601	0.485	0.0017	0.9310
TSOPTW	380	268.172	111.828	29.428	2.1735	0.7648
Fixed Penalty	330	328.399	1.601	0.485	0.0017	0.9310
Smooth Penalty	320	319.775	0.225	0.070	0.0059	0.9061
c107						
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	370	298.438	71.562	19.341	0.6826	-
CCP-model	350	348.074	1.926	0.550	0.0062	-
TSOPTW	380	333.999	46.001	12.106	1.8212	-
Fixed Penalty	350	348.074	1.926	0.550	0.0062	-
Smooth Penalty	350	348.069	1.931	0.552	0.0088	-

c108	Alternative Recourse Action					
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	370	296.699	73.301	19.811	0.7520	-
CCP-model	350	348.951	1.049	0.300	0.0081	-
TSOPTW	410	317.656	92.344	22.523	2.5558	-
Fixed Penalty	350	349.003	0.997	0.285	0.0056	-
Smooth Penalty	350	348.951	1.049	0.300	0.0081	-
c109						
Method	Planned profit	Realized profit	Absolute Gap	Percentage Gap	# of skipped targets	% of the PCI
OPTW	380	353.099	26.901	7.079	0.8722	-
CCP-model	370	366.282	3.718	1.005	0.0418	-
TSOPTW	400	369.506	30.494	7.624	1.8798	-
Fixed Penalty	370	365.425	4.575	1.236	0.0057	-
Smooth Penalty	360	359.554	0.446	0.124	0.0038	-

Table A.2: The results of the test instances for different methods combined with the alternative recourse action.