Utilizing Dual Information for Moving Target Search Trajectory Optimization*

Manon Raap[1], Maximilian Moll[2], Martin Zsifkovits[3], and Stefan Pickl[4]

1 **Universität der Bundeswehr München, Fakultät für Informatik, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany**
manon.raap@unibw.de

2 **Universität der Bundeswehr München, Fakultät für Informatik, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany**
maximilian.moll@unibw.de

3 **Universität der Bundeswehr München, Fakultät für Informatik, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany**
martin.zsifkovits@unibw.de

4 **Universität der Bundeswehr München, Fakultät für Informatik, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany**
stefan.pickl@unibw.de

Abstract

Various recent events have shown the enormous importance of maritime search-and-rescue missions. By reducing the time to find floating victims at sea, the number of casualties can be reduced. A major improvement can be achieved by employing autonomous aerial systems for autonomous search missions, allowed by the recent rise in technological development. In this context, the need for efficient search trajectory planning methods arises. The objective is to maximize the probability of detecting the target at a certain time k, which depends on the estimation of the position of the target. For stationary target search, this is a function of the observation at time k. When considering the target movement, this is a function of *all* previous observations up until time k. This is the main difficulty arising in solving moving target search problems when the duration of the search mission increases. We present an intermediate result for the single searcher single target case towards an efficient algorithm for longer missions with multiple aerial vehicles. Our primary aim in the development of this algorithm is to disconnect the networks of the target and platform, which we have achieved by applying Benders decomposition. Consequently, we solve two much smaller problems sequentially in iterations. Between the problems, primal and dual information is exchanged. To the best of our knowledge, this is the first approach utilizing dual information within the category of moving target search problems. We show the applicability in computational experiments and provide an analysis of the results. Furthermore, we propose well-founded improvements for further research towards solving real-life instances with multiple searchers.

1998 ACM Subject Classification G.2.3 Applications

Keywords and phrases Search Theory, UAV, Vehicle Routing, Benders Decomposition

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.1

---

* This work was supported by Airbus Defence & Space.

5th Student Conference on Operational Research (SCOR'16).
Editors: Bradley Hardy, Abroon Qazi, and Stefan Ravizza; Article No. 1; pp. 1:1–1:10
OASICS Open Access Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

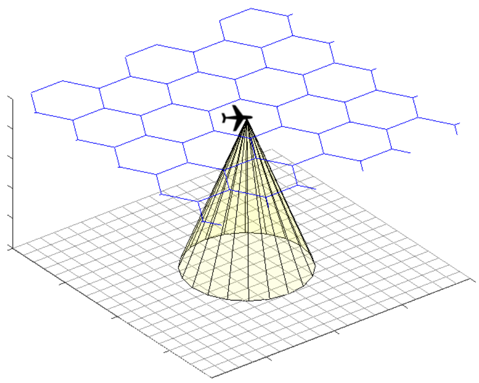

Figure 1 The heterogeneous state spaces for the target (square grid) and the platform (hexagonal grid).

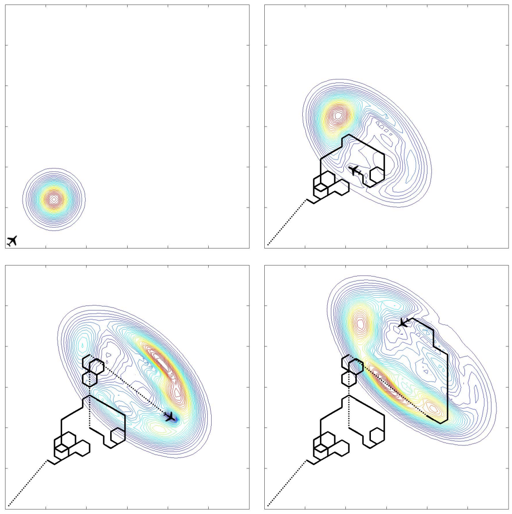

Figure 2 Snapshots of a simulation of one fixed-wing platform searching for one Markovian target. The contour plot indicates the updated probability distribution of the target location.

1 Introduction

In the spring of 2015, over 1,000 lifes were lost at sea in a single month after refugee ships sunk in the Mediterranean. These tragedies led to renewed efforts by governmental and non-governmental organizations that significantly decreased the numbers of deaths [27]. Search missions are often supported by manned aerial vehicles where individual pilots are expected to plan their optimal search trajectory by hand, which is tremendously complex; it is proven to be an $\mathcal{NP}$-complete optimization problem [26] for a single platform searching for a single *stationary* target. Planning for *moving* target search is considerably more complex in general. Executing such a complex task in a stressful situation is susceptible to result in a sub-optimal search trajectory. We therefore aim to automatize this task, with the outlook towards autonomous search missions by unmanned aerial vehicles (UAVs). We refer to an aerial sensor platform (e.g. UAV) by *platform* for short in the remainder of this article.

The contribution of this article consists of a Benders decomposition algorithm to solve a problem of Markovian target search trajectory optimization (see Figure 2). We decompose an existing mixed integer linear program (MILP) [19] that accounts for a heterogeneous state space for the target and platform. This concept is shown in Figure 1 and holds two benefits; a target specific grid allows for a more accurate estimation of the target position, whereas a platform specific grid allows for modeling more natural flight motion. By applying Benders decomposition, we are able to disconnect both networks and iteratively solve two much smaller problems until the desired optimality tolerance has been reached. A diagrammatic visualization of this decomposition is shown in Figure 3. To the best of our knowledge, this is the first method utilizing the dual information for solving a moving target search problem.

The remainder of this article is structured as follows: Section 2 provides an overview of the relevant related literature. In Section 3, the search trajectory problem is stated, followed by our methodology for solving this problem in Section 4. Computational experiments in Section 5 show the applicability of the proposed method. Finally, the conclusions are presented in Section 6.

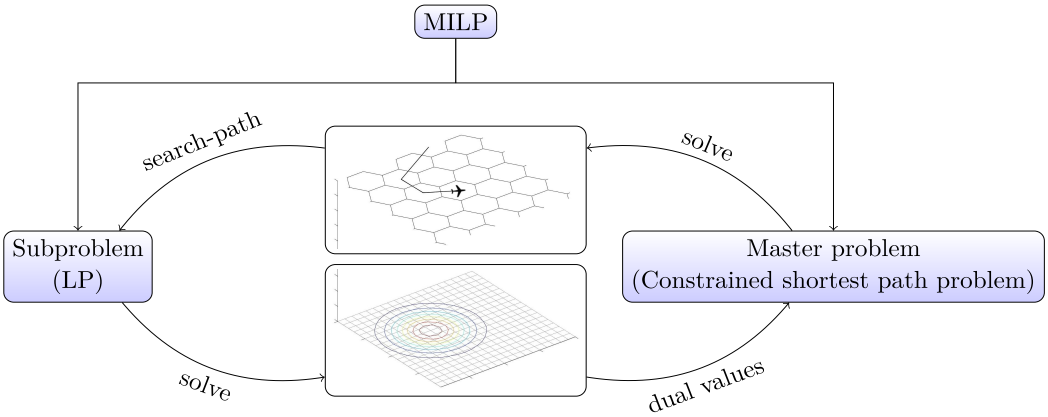

Figure 3 The MILP decomposes into a subproblem and a master problem. The master problem is a constrained shortest path problem on the platform network and the subproblem is a linear problem to calculate the probability of detection and containment on the target network, given a fixed search-path provided by the master. In turn, the subproblem provides the dual values to the master problem. As a result, the networks are disconnected and we solve two much smaller problems sequentially in iterations.

2 Related Literature

Research initiated by Koopman [13] has led to the interesting field of *Search Theory* [24, 25]. To apply search theory to real life search missions, efficient planning methods are necessary. Extensive research has led to several search strategy planning approaches, which can mainly be classified to two problem types. For the path-constrained search effort allocation problem [23], the search area is typically divided into subareas to which search effort is allocated over time. The search effort is expressed in number of platforms of a certain type and duration. For autonomous search, the kinematical properties of the sensor platform must be taken into account. The resulting problem is referred to by the search trajectory optimization problem. A depth-first branch and bound approach was presented in [23], in which lower bound approximations on the probability of non-detection are obtained by relaxing the searcher's path constraints. Most following approaches are of the branch and bound type [6, 7, 15, 17, 21], where the aim is to find a tightest true lower bound with low computational costs. Computational experiments of several branch and bound procedures are summarized in [28]. Despite vast progression over the years, computation remains intractable for larger instances. This has led to the development of heuristic approaches such as a receding horizon approach [5], cross entropy optimization [14] and constraint programming [18].

This article describes a Benders decomposition [11, 16] algorithm for a trajectory optimization problem for aerial vehicles. Also various other vehicle routing problems have been tackled by a Benders decomposition algorithm, i.e. [1, 4]. Algorithms of this type have also been developed for *combined* vehicle routing problems with allocation [2, 8], assignment [3, 9], and scheduling [22].

3 The Moving Target Search Trajectory Problem

We consider the search for a moving target in discrete time on a finite set of cells $\mathcal{C}$. This is a well known problem in literature and can be formally stated as follows. The target occupies one unknown cell $C_k \in \mathcal{C}$ at time step $k \in \mathcal{K}$. For the duration of the search mission, a probability map $\boldsymbol{pc}$ is maintained, where the probability of containment $pc_k(c) \in [0,1]$ represents the probability of the target occupying cell c at time k, without having been detected prior to time k. Although the initial position of the target is unknown, it is characterized by a known prior probability distribution $\boldsymbol{pc}_1$. The *target* path is modeled by a stochastic process $(C_1, ..., C_K)$, which we assume to be Markovian. Under this assumption, the probability map evolves due to the target motion according to

$$pc_{k+1}(c) = \sum_{c' \in \mathcal{C}} d(c', c)\, pc_k(c'), \tag{1}$$

where the transition function $d(c', c) \in [0,1]$ represents the probability that the target moves from cell c' to cell c and is assumed to be known for each pair $(c', c) \in \mathcal{C} \times \mathcal{C}$.

The *platform* is modeled to move over a finite grid of hexagonal shape. This grid is represented by a network $G = (\mathcal{V}, A)$ with the set of nodes $\mathcal{V}$ and binary adjacency matrix A. Here, an entry $a_{v,v'}$ of A is 1 if node $v \in \mathcal{V}$ is adjacent to node $v' \in \mathcal{V}$ and 0 otherwise. We assume that the considered aerial sensor platform has a sensor equipped to make observations. The glimpse probability $pg_k(v_k, c) \in [0,1]$ represents the probability of target detection, given target occupancy within cell c and platform position $v_k \in \mathcal{V}$ at time k. When observations are made, the probability map evolves according to the motion model as in Equation (1) and, in addition, evolves according to the glimpse probability due to the observations made when visiting node v_k. Therefore, Equation (1) is extended to account for observation results as follows:

$$pc_{k+1}(c) = \sum_{c' \in \mathcal{C}} d(c', c)\, pc_k(c')\, (1 - pg_k(v_k, c'))\,. \tag{2}$$

The *objective* is to determine a sequence of nodes $\boldsymbol{v} = (v_1, \ldots, v_K)$ maximizing the cumulative probability of detection over time period $\mathcal{K}$, i.e.

$$\max \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} pd_k(v_k, c), \tag{3}$$

where $pd_k(v_k, c)$ is the probability of detecting the target at time k from platform position v_k in cell c and is calculated by

$$pd_k(v_k, c) = pc_k(c)\, pg_k(v_k, c), \tag{4}$$

where the probability of containment $pc_k(c)$ is calculated through Equation (2).

3.1 Mixed Integer Linear Programming Formulation

The model for moving target search trajectory optimization under kinematical constraints can be stated as the following MILP. Here, let $\mathbb{B} = \{0, 1\}$. The decision variable $z_v^k \in \mathbb{B}$ is 1 if the platform is at node v at time k and 0 otherwise. The auxiliary decision variable $pd_c^k \geq 0$ represents the probability of detection in cell c at time k. The auxiliary decision variable $pc_c^k \geq 0$ represents the probability of containment in cell c at time k. We use *italic* font for all decision variables and normal font for input variables.

$$\text{Maximize} \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} pd_c^k \tag{5}$$

subject to

$$pd_c^k - \mathrm{pg}_{v,c}^k pc_c^k \leq 1 - z_v^k \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V}, \forall c \in \mathcal{C} \tag{6}$$

$$pd_c^k - \sum_{v \in \mathcal{V}} \mathrm{pg}_{v,c}^k z_v^k \leq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{7}$$

$$pc_c^k - \sum_{c' \in \mathcal{C}} \mathrm{d}_{c',c} pc_{c'}^{k-1} + \sum_{c' \in \mathcal{C}} \mathrm{d}_{c',c} pd_{c'}^{k-1} = 0 \qquad \forall k \in \{2, \dots, K\}, \forall c \in \mathcal{C} \tag{8}$$

$$\boldsymbol{z} \in \boldsymbol{Z} \tag{9}$$

$$pc_c^1 = \mathrm{pc}_c^1 \qquad \forall c \in \mathcal{C} \tag{10}$$

$$z_v^k \in \mathbb{B} \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V} \tag{11}$$

$$pd_c^k, pc_c^k \geq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{12}$$

The objective function (5) maximizes the cumulative probability of detection. The sets of constraints in (6) and (7) ensure calculation of pd_c^k according to the formula (4). The set of constraints in (8) ensure calculation of pc_c^k according to the formula (2) and the set of constraints in (10) ensure that the initial values of pc_c^1 are set according to the known prior probability distribution $\boldsymbol{pc}_1$. Let $\boldsymbol{Z}$ be the set of binary vectors for the z_v^k variables that yield feasible trajectories on the network G. We use this abstract notation, so that feasibility of a trajectory can be defined as desired. The set of constraints (9) ensures the binary vector $\boldsymbol{z}$ to be a feasible trajectory. The total number of decision variables, including the auxiliary decision variables pd_c^k and pc_c^k, is of order $\mathcal{O}(K|\mathcal{V}||\mathcal{C}|)$. The number of possible feasible trajectories on network G is of order $\mathcal{O}(2^{K|\mathcal{V}|})$.

4 Solution Methodology

As the number of time steps or the number of nodes increases, the problem becomes more difficult to solve. In order to decrease the rate in which computation time increases, we decompose model (5)–(12) into a pair of problems that can be solved more easily by applying Benders decomposition [10]. The primal and dual subproblem, as well as the Benders master problem are stated in the following subsections.

4.1 Primal Subproblem

Recall $\boldsymbol{Z}$ to be the set feasible trajectories on the platform network G. When fixing any binary vector $\bar{\boldsymbol{z}} \in \boldsymbol{Z}$, the original problem reduces to the following *primal subproblem* in the pc_c^k and pd_c^k variables. This linear program contains no integer decision variables and is therefore very easy to solve.

$$\text{Maximize} \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} pd_c^k \tag{13}$$

subject to

$$pd_c^k - \mathrm{pg}_{v,c}^k pc_c^k \leq 1 - \bar{z}_v^k \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V}, \forall c \in \mathcal{C} \tag{14}$$

$$pd_c^k - \sum_{v \in \mathcal{V}} \mathrm{pg}_{v,c}^k \bar{z}_v^k \leq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{15}$$

$$pc_c^k - \sum_{c' \in \mathcal{C}} \mathrm{d}_{c',c} pc_{c'}^{k-1} + \sum_{c' \in \mathcal{C}} \mathrm{d}_{c',c} pd_{c'}^{k-1} = 0 \qquad \forall k \in \{2, ..., K\}, \forall c \in \mathcal{C} \tag{16}$$

$$pc_c^1 = \mathrm{pc}_c^1 \qquad \forall c \in \mathcal{C} \tag{17}$$

$$pd_c^k, pc_c^k \geq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{18}$$

4.2 Dual Subproblem

Let $\boldsymbol{\pi} = (\pi^k_{v,c} | k \in \mathcal{K}, v \in \mathcal{V}, c \in \mathcal{C})$, $\boldsymbol{\rho} = (\rho^k_c | k \in \mathcal{K}, c \in \mathcal{C})$, $\boldsymbol{\sigma} = (\sigma^k_c | k \in \{2, ..., K\}, c \in \mathcal{C})$, and $\boldsymbol{\tau} = (\tau^1_c | c \in \mathcal{C})$ be the dual variables associated with constraints (14), (15), (16), and (17) respectively. The dual of the *primal subproblem* (13)-(18) is the following *dual subproblem*:

$$\text{Minimize} \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} \sum_{v \in \mathcal{V}} \pi^k_{v,c} \left(1 - \bar{z}^k_v\right) + \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} \rho^k_c \left(\sum_{v \in \mathcal{V}} \mathrm{pg}^k_{v,c} \bar{z}^k_v \right) + \sum_{c \in \mathcal{C}} \tau^1_c \mathrm{pc}^1_c \tag{19}$$

subject to

$$\sum_{v \in \mathcal{V}} \pi^k_{v,c} + \rho^k_c + \mathbb{1}_{\{k \le K-1\}} \sum_{c' \in \mathcal{C}} \sigma^{k+1}_{c'} d_{c,c'} \ge 1 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{20}$$

$$\sum_{v \in \mathcal{V}} \mathrm{pg}^k_{v,c} \pi^k_{v,c} - \mathbb{1}_{\{k \ne 1\}} \sigma^k_c + \mathbb{1}_{\{k \le K-1\}} \sum_{c' \in \mathcal{C}} \sigma^{k+1}_{c'} d_{c,c'} \tag{21}$$

$$-\mathbb{1}_{\{k=1\}} \tau^1_c \le 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{22}$$

$$\pi^k_{v,c}, \rho^k_c \ge 0 \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V}, \forall c \in \mathcal{C} \tag{23}$$

$$\sigma^k_c, \tau^1_c \in \mathbb{R} \qquad \forall k \in \{2, \ldots, K\}, \forall c \in \mathcal{C} \tag{24}$$

Here, function $\mathbb{1}$ denotes the indicator function so that $\mathbb{1}_{\{x\}} = 1$ if x is true and 0 otherwise. Let $\boldsymbol{\Delta}$ be the polyhedron defined by constraints (20)-(23), and let P_Δ denote the set of extreme points of $\boldsymbol{\Delta}$. Polyhedron $\boldsymbol{\Delta}$ contains no rays because the subproblem is feasible for each $\bar{\boldsymbol{z}} \in \boldsymbol{Z}$.

4.3 Benders Master Problem

We introduce the free decision variable $y_0 \in \mathbb{R}$ and reformulate the original model (5)-(12) as the following *Benders master problem*:

$$\text{Maximize } y_0 \tag{25}$$

subject to

$$\sum_{k=1}^{K} \sum_{c \in \mathcal{C}} \sum_{v \in \mathcal{V}} \pi^k_{v,c} \left(1 - \bar{z}^k_v\right) + \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} \rho^k_c \left(\sum_{v \in \mathcal{V}} \mathrm{pg}^k_{v,c} \bar{z}^k_v \right) \tag{26}$$

$$+ \sum_{c \in \mathcal{C}} \tau^1_c \mathrm{pc}^1_c \ge y_0 \qquad \forall (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\tau}) \in P_\Delta \tag{27}$$

$$\boldsymbol{z} \in \boldsymbol{Z} \tag{28}$$

$$z^k_v \in \mathbb{B} \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V} \tag{29}$$

$$y_0 \in \mathbb{R} \tag{30}$$

The set P_Δ is very large in general, resulting in many constraints in the Benders master problem. However, most of these constraints are not active in an optimal solution. We therefore do not have to include all these constraints, but generate a subset in an iterative algorithm. In each iteration, a relaxed Benders master problem is solved, which is obtained by replacing the set P_Δ with the subset P^t_Δ of extreme points available at iteration $t = 0, \ldots$. At the start, this subset is empty, i.e. $P^0_\Delta = \emptyset$. Let $(\boldsymbol{z}, y_0)$ be an optimal solution to this problem. Next, the primal subproblem (13)-(18) is solved with the values of $\boldsymbol{z}$ being fixed. This yields the dual variables $(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\tau})$ associated with constraints (14), (15), and (17) respectively.

Because the primal subproblem is feasible for each $\boldsymbol{z}$, the values of the dual variables $(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\tau})$ determine an extreme point of P_Δ and we add this point to P_Δ^{t+1}. As a consequence, one constraint is added to the relaxed Benders master problem. This process continues until its optimal value equals the optimal value of the primal subproblem.

5 Computational Experiment

In this section, we present computational experiments to show the applicability of our algorithm. First, we present the test instances of our search problem, followed by the results and a concise analysis thereof.

The experiments were performed on an Intel(R) CoreTM i7-4810MQ CPU processor with 2.80 GHz and a usable memory of 15.6 GB. The instances were generated with our simulation platform, that is written in Matlab. The original MILP and the Benders decomposition algorithm were coded in Java and solved using CPLEX with default parameters. We set the maximum number of iterations in the Benders decomposition algorithm to 3000 to avoid out-of-memory exceptions.

5.1 Description of the Instances

In the instances generated for our analysis we consider one target and one platform. The target state space $\mathcal{C}$ consists of 60 square cells, over which the initial target position is normally distributed in two dimensions with its mean in the center and with a standard deviation of 12. The target moves in north direction with 0.5 probability and in east direction with 0.5 probability as well, with a speed of one cell per time step. The platform state space $\mathcal{V}$ consists of 20 nodes and its start position is trivial, since we allow the platform to reach any node in the first time step. It then searches with the speed of one node per time step. The ratio of the lengths of a square cell and a hexagonal cell is $2:1$. For sensor characteristics, we used the typical glimpse probability function [7]:

$$pg_k(v_k, c) = 1 - \exp^{-\omega(v,c,k)}, \tag{31}$$

with $\omega(v,c,k) \geq 0$ being a measure of search effectiveness for cell c. The search effectiveness decreases with the Euclidean distance $||v_k - c||$ between cell c and the platform at node v_k, as follows:

$$\omega(v,c,k) = W\left(||v_k - c||\right)^{-1},$$

where W is some sensor quality indicator. We used $W = 0.7$ and generated four instances with these properties, for time periods $K = (4, 6, 8, 10)$.

5.2 Results and Analysis

We optimized the search trajectory on the four instances with the proposed Benders decomposition (BD) algorithm, and additionally with CPLEX on the original formulation for benchmarking. An optimal trajectory was found for all instances by both algorithms, although we noticed differences in computation time. In Table 1, we list for each instance the time period (K), the optimal value in cumulative probability of detection (PD), the number of iterations in the BD algorithm, time to solve for the BD algorithm in seconds, total and average time spend on solving the relaxed Benders master problem (RBMP) and on the primal subproblem (PSP), and finally the time needed for CPLEX to solve the original formulation (MILP).

Table 1 Test results.

K	Optimal value (PD)	Iterations (#)	BD (s)	RBMP Tot (s)	RBMP Avg (s)	PSP Tot (s)	PSP Avg (s)	MILP (s)
4	0.2786	352	65.43	62.13	0.18	2.22	0.01	0.77
6	0.3213	884	182.88	152.25	0.17	11.25	0.01	2.22
8	0.3368	1980	1738.19	1535.37	0.78	113.43	0.06	7.97
10	0.3439	3000	8740.82	8171.55	2.72	288.66	0.10	15.46

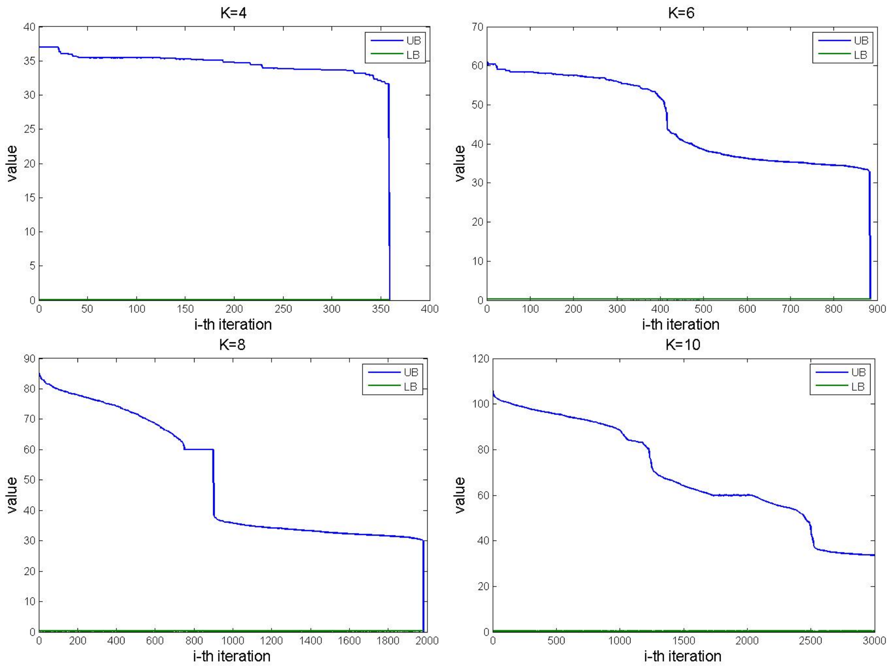

Figure 4 Convergence profiles of the instances $K = 4$ shown top-left to $K = 10$ shown bottom-right.

As the results in Table 1 show, the proposed BD algorithm is not faster than solving the original MILP. Even for the instance with $K = 10$, the cap of 3000 iterations was hit and the algorithm was terminated before it converged. The optimal solution was found, but not proved. The results however supply valuable information; the major portion of time is spend on solving the RBMP, whereas the PSP is very easily optimized. Also the convergence profiles (see Figure 4) show that many cuts are necessary to move out of local optima, of which we seem to have $(K/2)-1$ in each instance. These observations lead us to the following considerations for further research.

5.3 Considerations for Further Research

There are several improvements to be considered, which show potential to lead to improved calculation times with respect to solving the original MILP:

- One improvement is to utilize local branching for Benders decomposition [20]. The nodes resulting from a branching step each represent a smaller subregion of the feasible region, so that cuts are found more easily. This method is applicable to any binary program that can be solved by BD and may reduce the computation time significantly.

- A major difficulty lies in solving the RBMP, which is a large 0-1 linear program with one continuous variable. Improvements can be made by decreasing the computation time on this problem, which is actually a well studied resource constrained longest path problem on a directed acyclic graph. It can be solved by specific and more efficient algorithms as described in [12], such as dynamic programming and Lagrangean relaxation. Also heuristics are applicable, because it is possible to generate new cuts from any feasible integer solution.
- A major strength of our formulation is that the PSP is an easy to solve LP. This may prove to be necessary when solving the multi-platform case using Benders decomposition. The multi-platform search trajectory problem can then be decomposed into multiple RBMPs together with a single PSP providing the dual information for each platform specific RBMP.

6 Conclusion

We considered the problem of optimizing a Markovian target search trajectory under kinematical constraints. For this problem, we proposed a novel method that utilizes dual information by applying Benders decomposition. The aim of this research was to develop a method that disconnects the networks of the target and platform and thereby solving much smaller problems in iterations. Results from computational experiments show this accomplishment. Possibilities for efficiently solving this problem for the multi-platform case arise, because the master problem decomposes into multiple smaller problems for each platform. This is of importance because the original formulation grows exponentially in the number of platforms. All known exact methods for solving the multi-platform case appear intractable for very small instances already and are therefore not applicable to real-life *cooperative* search missions. The network disconnection as accomplished in this work is a first step towards an exact method for solving the multi-platform case that grows only linearly in the number of platforms. Therefore, this is a significant step towards automated cooperative search missions by unmanned aerial vehicles.

Acknowledgements. This work was supported by Airbus Defence & Space, Germany.

References

1. Tolga Bektaş. Formulations and benders decomposition algorithms for multidepot salesmen problems with load balancing. *European Journal of Operational Research*, 216(1):83–93, 2012.
2. James H. Bookbinder and Kathleen E. Reece. Vehicle routing considerations in distribution system design. *European Journal of Operational Research*, 37(2):204–213, 1988.
3. Jean-François Cordeau, François Soumis, and Jacques Desrosiers. A benders decomposition approach for the locomotive and car assignment problem. *Transportation science*, 34(2):133–149, 2000.
4. Yvan Dumas, Jacques Desrosiers, Eric Gelinas, and Marius M. Solomon. An optimal algorithm for the traveling salesman problem with time windows. *Operations research*, 43(2):367–371, 1995.
5. James N. Eagle. *The approximate solution of a simple constrained search path moving target problem using moving horizon policies.* Monterey, California. Naval Postgraduate School, 1984.
6. James N. Eagle. The optimal search for a moving target when the search path is constrained. *Operations Research*, 32(5):1107–1115, 1984.

7 James N. Eagle and James R. Yee. An optimal branch-and-bound procedure for the constrained path, moving target search problem. *Operations Research*, 38(1):110–114, feb 1990.
8 Awi Federgruen and Paul Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037, 1984.
9 Marshall L. Fisher and Ramchandran Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.
10 Arthur M. Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260, 1972.
11 John N. Hooker and Greger Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.
12 Stefan Irnich, Guy Desaulniers, et al. Shortest path problems with resource constraints. *Column generation*, 6730:33–65, 2005.
13 Bernard Osgood Koopman. *Search and screening: general principles with historical applications.* Pergamon Press, 1980.
14 P. Lanillos, E. Besada-Portas, G. Pajares, and J.J. Ruz. Minimum time search for lost targets using cross entropy optimization. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 602–609, oct 2012.
15 Haye Lau, Shoudong Huang, and Gamini Dissanayake. Discounted MEAN bound for the optimal searcher path problem with non-uniform travel times. *European Journal of Operational Research*, 190(2):383–397, 2008.
16 Richard Kipp Martin. *Large scale linear and integer optimization: a unified approach.* Springer Science & Business Media, 2012.
17 Gustavo H. A. Martins. A new branch and bound procedure for computing optimal search paths. Master's thesis, Naval Postgraduate School, Monterey, CA, 1993.
18 Michael Morin, Anika-Pascale Papillon, Irène Abi-Zeid, François Laviolette, and Claude-Guy Quimper. Constraint programming for path planning with uncertainty. In Michela Milano, editor, *Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, pages 988–1003. Springer Berlin Heidelberg, 2012.
19 M. Raap, S. Pickl, and M. Zsifkovits. Trajectory optimization under kinematical constraints for moving target search. *(working paper)*, 2016.
20 Walter Rei, Jean-François Cordeau, Michel Gendreau, and Patrick Soriano. Accelerating benders decomposition by local branching. *INFORMS Journal on Computing*, 21(2):333–345, 2009.
21 Hiroyuki Sato and Johannes O. Royset. Path optimization for the resource-constrained searcher. *Naval Research Logistics*, 57(5):422–440, 2010.
22 Thomas R. Sexton and Lawrence D. Bodin. Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. *Transportation Science*, 19(4):378–410, 1985.
23 T.J. Stewart. Search for a moving target when searcher motion is restricted. *Computers & Operations Research*, 6(3):129–140, 1979.
24 Lawrence D. Stone. *Theory of Optimal Search.* 1975.
25 Lawrence D. Stone, Johannes O. Royset, and Alan R. Washburn. *Optimal Search for Moving Targets.* Springer, 2016.
26 K. E. Trummel and J. R. Weisinger. The complexity of the optimal searcher path problem. *Operations Research*, 34(2):324–327, mar 1986.
27 UN High Commissioner for Refugees (UNHCR). The sea route to europe: The mediterranean passage in the age of refugees, 2015. Last accessed: 24 Feb 2016.
28 Alan R. Washburn. Branch and bound methods for a search problem. *Naval Research Logistics (NRL)*, 45(3):243–257, 1998.