Journal of Intelligent & Robotic Systems (2019) 94:857–869
https://doi.org/10.1007/s10846-018-0910-9

Coordinated Target Assignment and UAV Path Planning with Timing Constraints

Luitpold Babel[1]

Received: 18 January 2018 / Accepted: 17 July 2018 / Published online: 24 July 2018

Abstract

The engagement of a group of autonomous air vehicles against several targets is a major challenge in mission planning. This paper addresses the problem of cooperative flight path planning where the air vehicles should arrive at the destinations simultaneously or sequentially with specified time delays, while minimizing the total mission time. This involves finding an optimal assignment of air vehicles to targets and generating trajectories in compliance with the kinematic constraints of the vehicles. The trajectories have to avoid nofly-areas, threats and other obstacles, and must prevent the air vehicles from colliding with each other. The presented algorithm for simultaneous arrival first calculates shortest flight paths between all pairs of air vehicles and targets using a network-based routing model. An optimal assignment and a critical path is found by solving a linear bottleneck assignment problem with costs corresponding to the lengths of the shortest paths. The other flight paths are prolongated to the length of the critical path by automatic insertion of waypoints. This is achieved by concatenating subpaths stored in different shortest-path-trees. Due to the special structure of the network, all concatenated flight paths are flyable and feasible. Sequential arrival at a target is realized by sorting the flight paths according to their lengths and prolongating them whenever necessary to accomplish the desired time delays. The capability of the approach is demonstrated by simulation results.

Keywords Cooperative flight path planning · Trajectory generation · Target assignment · Linear bottleneck assignment problem · Unmanned air vehicles

Mathematics Subject Classification (2010) 05C85 · 90B15 · 90C27 · 90C35 · 94C15

1 Introduction

In recent years, the deployment of cooperative teams of unmanned air vehicles (UAVs) has become a field of major interest, both in civilian and in military sectors. Applications include search and rescue, intelligence surveillance and reconnaissance, suppression of enemy air defense, combat operations, and others. It is expected that the capabilities of a joint system far exceeds the sum of its individual parts. Most missions can be accomplished more effective by cooperation and coordination of the team members. In this context, task assignment and path planning play a key role during the planning phase of a mission.

This paper deals with the engagement of a group of autonomous air vehicles against a number of ground targets. The task is to find an assignment of air vehicles to targets and flight paths to the targets such that (1) all air vehicles arrive simultaneously at the target locations and (2) the total mission time is as short as possible. The problem is further complicated by the existence of obstacles, flight restricted areas and threats endangering the success of the mission. The main motivation for simultaneous arrival is to enhance the element of surprise. Another reason is the saturation of the air defense. Attacking targets simultaneously by several missiles is more likely to be successful than consecutively, simply due to the saturation of the anti-aircraft batteries involved.

The second problem refers to a group of air vehicles that has to accomplish multiple consecutive tasks on the same

✉ Luitpold Babel
luitpold.babel@unibw.de

[1] Fakultät Betriebswirtschaft, Institut für Mathematik und Informatik, Universität der Bundeswehr, München, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany

target. Each air vehicle is able to perform each of the tasks. The objective is to find flight paths to the target such that (1) the air vehicles arrive sequentially at the target location with given time delays and (2) the total mission time is minimized. A typical application is the suppression of a hostile air defense site requiring detection, destruction, and verification (damage assessment) of the target. A further example is coordinating the landing of a fleet of aircraft on a single runway. Along with performing the tasks, the fuel consumption of the air vehicles should be kept as low as possible.

A crucial issue for flight path generation is that all paths must be *flyable*, *feasible* and *safe*. The first property refers to the kinematic capabilities of the air vehicle. The vehicle must be able to follow the path, i.e. the maneuvers must be in accordance with the acceleration limits of the aircraft. The second property states that the flight path must avoid all obstacles, restricted areas and threats. Safety means to guarantee that air vehicles are not colliding with each other. This is realized by maintaining a safety distance between the vehicles.

There are quite a lot of publications dealing with various aspects of cooperating UAVs. This includes (A) task allocation problems, (B) combined task allocation/path planning problems, and (C) cooperative flight path planning problems with predefined task assignments.

The Weapon Target Assignment Problem is a combinatorial optimization problem that has been studied for several decades. It consists of optimally assigning n weapons to m targets such that the total expected damage to the targets is maximized. It can be formulated as a nonlinear integer programming problem and is known to be NP-complete. Numerous exact methods for small-size problems and heuristic methods for large-size problems have been proposed (see e.g. Ahuja et al. [1]).

The Stochastic Weapon Target Assignment Problem where numbers and locations of targets are not known a priori has been studied, among others, by Murphey [35]. A similar problem for UAVs, considering failure probabilities of the vehicles and providing a rudimentary flight path planning, has been investigated by Bellingham et al. [7]. The authors pose the problem as a mixed-integer linear program and solve it using a commercial software package.

A multiple task allocation problem where UAVs with different capabilities have to perform a set of tasks has been discussed by Bellingham et al. [5]. Using straight-line path approximations, the problem is solved using a formulation as a special type of knapsack problem. Other approaches applying mixed-integer linear programming include the work of Schumacher et al. [40] and Weinstein and Schumacher [45]. Tabu search heuristics have also been applied to the assignment problem, see e.g. Alighanbari et al. [2].

A combined target assignment and path planning problem is discussed in Maddula et al. [32]. The authors present heuristic methods for assigning a group of UAVs to multiple targets, thereby limiting the threat faced by each UAV and minimizing the maximum path length. The idea is to create potential subpaths by means of a Voronoi tessellation, to construct a graph containing these subpaths and to calculate short paths between targets and UAVs in the graph. An initial assignment of UAVs to targets obtained by a greedy heuristic is successively improved by exchanging subpaths. A similar problem with timing constraints for simultaneous attack and multiple consecutive tasks has been investigated and solved using genetic algorithms by Eun and Bang [18]. However, both approaches do not consider the maneuverability of the air vehicles and possible collisions between UAVs.

Flight properties of air vehicles are taken into account by Shima et al. [42]. The authors study the problem where each UAV is required to fulfill three consecutive tasks and where the cumulative length of all paths has to be minimized. The problem is formulated as a large combinatorial optimization problem and solved by a genetic algorithm, using a tree representation of the possible assignments. The flight paths are realized as Dubins paths consisting of straight lines and circle segments. Unfortunately, the algorithm neglects obstacles and the risk of mutual collisions.

Approaches for cooperative flight path planning with given target assignment include an algorithm for fuel-optimal path planning presented by Schouwenaars et al. [39]. A fleet of UAVs has to move from predefined initial states to predefined final states without colliding with each other and with other stationary or moving obstacles, such that the total fuel consumption is as small as possible. The problem is rewritten as a linear program with mixed integer/linear constraints and solved using a commercial optimization package. However, it seems that flight properties and safety requirements of the UAVs are not sufficiently considered. A number of enhancements improving the efficiency of the approach have been reviewed and compared by Melor et al. [34].

Several papers have been published dealing with the planning of threat-avoiding trajectories with simultaneous arrival at the targets. The algorithm proposed by McLain and Beard [33] determines initial paths from the start positions to the target positions from a graph search through the edges of a Voronoi diagram. These paths are discretized into fixed-length segments and subsequently optimized to meet the desired length by adding or taking away segments and smoothing the outcome. Another method based on Voronoi diagrams, with special emphasis on the development of paths within the dynamic capabilities of the UAVs, is due to Chandler et al. [11]. However, the algorithms do not consider the risk of mutual collisions of air vehicles.

Collisions with other UAVs and with static obstacles are taken into account by the algorithm of Shanmugavel et al. [41]. The three-step procedure starts with the generation of flyable paths in the form of Dubins paths with clothoid arcs. These paths are then modified to safe paths by manually including waypoints. The third step is to produce paths of equal length by changing the curvature of the paths. The method seems to be working well for not overly complex scenarios with few and small obstacles and if the initial path lengths are not very different from each other.

The intention of this work is to develop an algorithm combining task assignment and coordinated flight path planning for a fleet of UAVs. The algorithm should allow timing constraints for target arrival and thoroughly consider flyability, feasibility and safety of the flight paths, thereby trying to eliminate weaknesses of previous strategies. The algorithm should be fully automatic, without the need for intervention by an operator, be reasonably simple to implement with modest computer running time, and effective to handle also challenging scenarios.

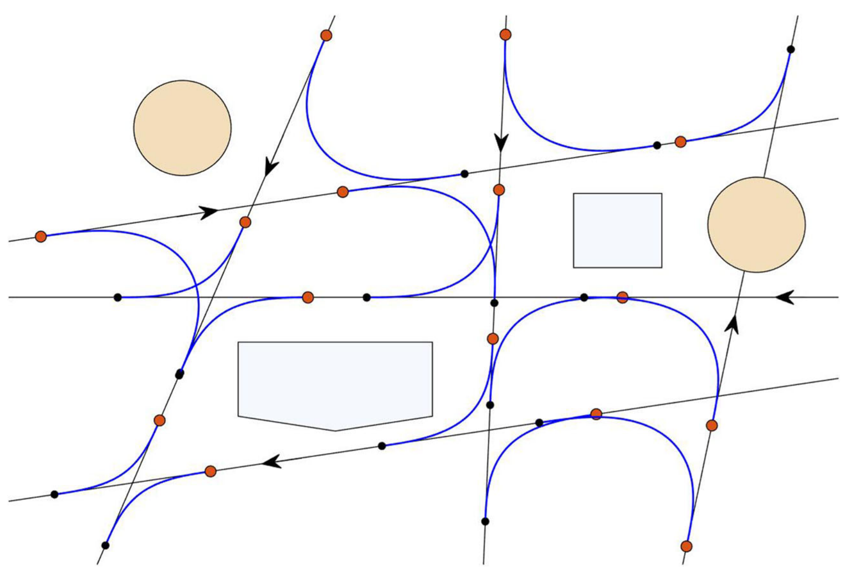
We propose an algorithm that is divided into three phases. In the first phase, shortest flight paths are computed between all pairs of air vehicles and targets. This is done by generating a sophisticated network discretizing the configuration space and applying standard graph search methods. The second phase is to find an optimal assignment of UAVs to targets or tasks. In case of simultaneous arrival at different targets, one has to solve a linear bottleneck assignment problem where the costs correspond to the lengths of the shortest paths. In case of sequential arrival at a common target, it suffices to sort the flight paths according to their lengths and reschedule the flight times. In the third phase, flight paths have to be prolongated to the desired lengths and adapted to eliminate possible conflicts between UAVs. The idea is to insert suitable waypoints. The modified flight paths are obtained by concatenating the subpaths to the waypoints which are stored in different shortest-path-trees. Due to the special structure of the network, it is guaranteed that all concatenated flight paths are both flyable and feasible. A sufficiently dense network contains a large number of suitable waypoints, making it possible to select a safe flight path without conflicts.

The rest of the paper is organized as follows. Section 2 sketches an algorithm for flight path planning of single vehicles that will form the basis for cooperative flight path planning. Section 3 describes the strategy for target assignment and coordinated path planning and details the technical realization. Simulation results for various scenarios are discussed in Section 4. We conclude with final remarks in Section 5.

2 Flight Path Planning for Single Vehicles

A variety of different techniques has been developed over time to solve the flight path planning problem. This includes the potential field method (see e.g. Kim and Khosla [25], Waydo and Murray [44]), cell decomposition (Chazelle [12], Lingelbach[31]), the roadmap method (Choset [13], Kavraki et al. [24]), rapidly exploring random trees (LaValle and Kuffner [30]), mixed-integer linear programs (Schouwenaars et al. [39], Bellingham et al. [6]),

Fig. 1 Mikado network generated from randomly located Mikado jackstraws



probabilistic methods (Bertuccelli and How [8], Dogan [17], Pfeiffer et al. [38]) and diverse network-based approaches (see e.g. Babel [3], Bortoff [9], Jun and D'Andrea [23]).

Further approaches are aimed at refining the graph search algorithm of Dijkstra [16]. This ranges from the A*-algorithm (Hart et al. [22]) to the incremental A*-algorithm (Koenig and Likhachev [26]), the D*-algorithm (Stentz [43]), the D*-lite algorithm ((Koenig and Likhachev [27]), the field D*-algorithm (Ferguson and Stentz [19]), the Theta*-algorithm ((Nash et al. [36], De Filippis et al. [15]) and the L*-algorithm ((Niewola and Podsedkowski [37]). Comprehensive reviews of path planning algorithms are presented by Goerzen et al. [21], Latombe [28] and LaValle [29].

We use an enhanced version of a flight path algorithm introduced by Babel [4]. The algorithm is best suited for our purpose since it allows in a very simple and effective way to manipulate flight paths by inserting waypoints. This in turn makes it possible to extend path lengths to almost any quantity, preserving both flyability and feasibility (see Section 3).

The main idea is to create a sophisticated network discretizing the horizontal configuration space. A large set of directed lines is thrown into the operating area like *Mikado jackstraws*, see Fig. 1. The flight paths proceed along these lines. Crossing lines are connected by two smooth transitions. The branching points (indicated as red circles) are the vertices of the network, the flight path segments between the branching points are the edges. The costs of the edges are the lengths of the flight path segments. In order to include release and destination point, the network is augmented by two lines passing through these points with release and approach direction, respectively. Finally, all edges of the network passing through obstacles, restricted areas and threats are eliminated. A shortest path is obtained by applying standard methods such as Dijkstra's algorithm or the A*-algorithm.

In the original version of the algorithm all lines are thrown randomly into the plane, thus providing a purely probabilistic network. Based upon practical experience, we extend the approach by supplementing the set of randomly generated lines with a modest number of deterministically determined lines. Given two polygonal or circular obstacles, we add those lines touching but not intersecting the obstacles and, at the same time, not intersecting any other obstacle in between (see Fig. 2). The lines, one for each direction, allow to pass the obstacles in straight flight as close as possible.

Transitions between crossing lines are realized by combinations of clothoids and circle segments, see Fig. 3a. A clothoid is a curve whose curvature grows linearly with its length. The transition curve passes over from the initial straight line to a clothoid with linearly increasing curvature,

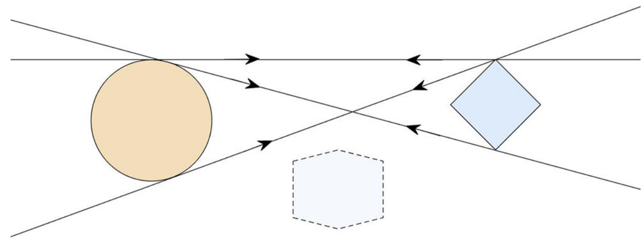


Fig. 2 Deterministically determined Mikado jackstraws

followed by a circle segment with constant curvature and a second clothoid with linearly decreasing curvature, to finally join the second straight line. This leads to a trajectory with continuous curvature and hence smooth and jerk-free lateral acceleration of the air vehicle.

The variability and number of possible flight paths in the network can be increased by integrating transitions with different maximal curvatures, corresponding to maneuvers of different strengths. An example with four possible transitions is shown in Fig. 3b. The shortest curve realizes the strongest maneuver with maximal lateral acceleration, the other curves are longer with weaker lateral acceleration.

The algorithm allows arbitrary flight directions and arbitrary turn angles as well as maneuvers of different strengths, thus fully exploiting the flight capabilities of the air vehicle. All paths in the network are flyable since transitions between straight flight path segments are smooth and in accordance with the lateral acceleration of the vehicle, and feasible since no flight path segment collides with an obstacle. A further strength of the algorithm is that the density of the network is freely adjustable according to the accuracy requirements of the solution. This is in contrast to the commonly used regular grid networks, where grid points are identified with vertices, thus leading to a limitation of the resolution.

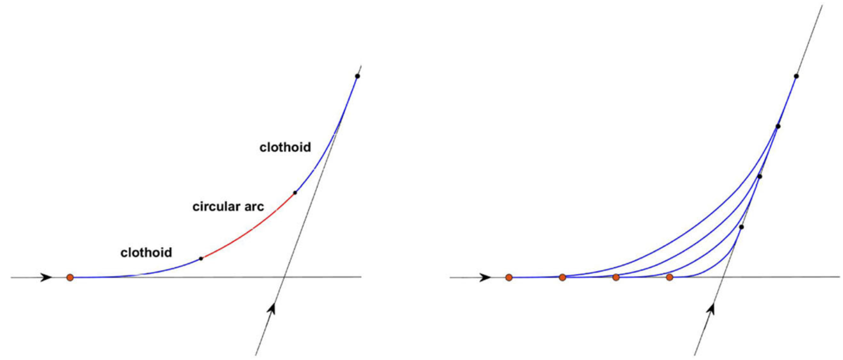
3 Coordinated Flight Path Planning

3.1 Simultaneous Arrival

We consider a group of n UAVs leaving simultaneously from a base, with the task to engage n targets. Release positions of the UAVs with release directions and target positions with approach directions are predefined. All air vehicles are of the same type and are flying at constant altitude. The scenarios may contain threats, nofly-areas and obstacles. The objective is to assign UAVs to targets (one UAV for each target) and to generate flyable, feasible and safe flight paths with simultaneous arrival at the targets and with smallest possible mission time.

The idea of coordinated mission planning is sketched in the following algorithmic scheme.

Fig. 3 Transitions between Mikado jackstraws



(a) Clothoids and circle segments (b) Differently strong maneuvers

Algorithm I

(1) *Individual flight path planning*
For each UAV compute a shortest flight path to each of the targets.

(2) *Target assignment*
Find an assignment of UAVs to targets such that the length L of the longest flight path is minimal. Let T denote the flight time of the longest path under nominal velocity.

(3) *Flight path adaptation and coordination*
Manipulate all flight paths with length smaller than L to meet the mission time T and resolve conflicts arising from UAVs coming too close to each other.

Step (1) of the algorithm requires planning of flight paths for a group of vehicles, independently of one another and without considering possible collision between the vehicles. The problem to be solved in step (2) is known in the literature as the *Linear Bottleneck Assignment Problem* (LBAP). It has been introduced by Fulkerson et al. [20] in connection with assigning jobs to parallel working machines as to minimize the latest completion time. More generally, we are given a number of agents and an equal number of tasks. Any agent can perform any task, incurring some cost that may vary depending on the agent-task assignment. All tasks have to be performed by assigning exactly one agent to each task. The goal is to find an assignment such that the maximum cost among the individual assignments is as small as possible.

The term bottleneck refers to a common type of optimization problem where the maximum cost or maximum duration of the task has to be minimized. The LBAP can be solved efficiently by applying the threshold algorithm (see Burkard et al. [10]). The algorithm runs in time $O(n^{2.5}/\sqrt{\log n})$. Clearly, for a small number of agents and tasks the problem can also be solved by exhaustive search, i.e. consider all $n!$ possible assignments of agents to tasks and choose the assignment where the maximum cost or duration is minimal.

In order to address step (3) of the algorithm, the two basic questions arise how to modify flight paths to meet the mission time T, and how to avoid possible collisions between UAVs. Strategies to solve the first problem, known from the literature, include:

– Reducing the velocity of the air vehicle, i.e. the flight path remains unchanged, but the flight time increases (see e.g. Chandler et al. [11]).
– Increasing path length by decreasing the curvature of the curves (see Shanmugavel et al. [41] where the curvature is decreased iteratively). This corresponds to decreasing the lateral acceleration of the air vehicle.
– Increasing path length by inserting loitering maneuvers (loops) or short detours (see e.g. McLain and Beard [33]).

Unfortunately, the strategies are not always successful. Air vehicles come with reserves that allow deviations from the nominal velocity. On the other hand, these deviations are rather limited for most UAVs and missiles. Hence only minor differences between the lengths of the flight paths can be compensated by adapting the velocity. Changing the curvature or lateral acceleration modifies the course of the flight path. This may lead to collisions, in particular in challenging scenarios that are densely occupied with obstacles. The same is true for inserting loops or detours. In the presence of obstacles, the feasibility of the modified paths is not guaranteed.

Potential strategies for avoiding collisions between air vehicles might be:

– Varying the velocity of the air vehicles. The flight path remains unchanged but the velocity is increased and decreased along parts of the path.
– Locally modifying the flight path (e.g. by changing the strength of flight maneuvers or inserting additional avoidance maneuvers).
– Changing the flight altitude during the whole flight or within regions where vehicles might come too close to each other.

These ideas come along with problems similarly as above. Changes of the velocity provide only limited variation options. Changing the course of the path might lead to collisions with obstacles and alters the path length and flight time, just like the insertion of climb or descent phases.

We suggest a different approach. Flight paths not meeting the desired length L are extended by automatically inserting suitable waypoints. This permits us to realize nearly any kind of prolongation, small and medium ones, but also large ones. The key issue is that the modified flight paths remain both flyable and feasible. Typically, for each flight path being shorter than L, there exists a large number of potential waypoints providing paths of length (almost) L, many of them following different routes. Smaller deviations from L can be compensated by adapting the velocity.

In step (3) of the algorithm, the flight paths that are too short are prolongated one after another. Simultaneously, it is checked whether the selected path collides with one of the previously fixed paths. This is done by calculating the minimal distance of the air vehicles in a simulated flight. If the distance is smaller than a safety threshold then there is a risk of collision and the selected path is rejected. Instead, another prolongation leading through another waypoint is chosen. The large number of alternatives usually allows to identify a safe flight path.

3.2 Technical Realization

The task in step (1) of the algorithm is to find n^2 shortest flight paths from the release points of the UAVs to the targets. For that purpose a network is created discretizing the horizontal configuration space. The generation follows the description in Section 2. Shortest-path calculations are performed with the algorithm of Dijkstra. It fixes a single vertex as the source vertex and finds shortest paths from the source to all other vertices in the graph, producing a *shortest-path-tree* (see e.g. Cormen et al. [14]). The source vertex is the root of the tree. The usual way to store the tree is to assign to each vertex its predecessor on the path from the root to the vertex. A shortest path is then retrieved by simply processing the sequence of predecessors. The length of the shortest path is the sum of its edge weights. To obtain all n^2 shortest paths we run Dijkstra's algorithm n times, with the release points of the UAVs being the source vertices. The n shortest-path-trees will be reused later when inserting waypoints into paths.

The longest (or bottleneck) flight path in the solution of the target assignment problem will be referred to as the *critical path* or *reference path*. It should be noted that, although the UAV and the target belonging to the reference path are unique, this is not the case for the assignment of the other air vehicles to the other targets. There are usually many possible assignments. If L is the length of the reference path and v_0 the nominal velocity of the air vehicles, then the total mission time is given by $T = L/v_0$.

The *waypoints* used for the prolongation of flight paths in step (3) of the algorithm are the branching points of the network (see Fig. 1). Each waypoint is characterized by a position and a direction. Obviously, a shortest path from a UAV to a target via a given waypoint consists of a shortest path from the UAV to the waypoint and a shortest path from the waypoint to the target. The length is the sum of the lengths of both subpaths. What is crucial here is that the transition between the subpaths is smooth. Hence the concatenation of the subpaths again provides a flyable and feasible flight path.

Shortest paths from the release points of the UAVs to all waypoints are implicitly given by the associated shortest-path-trees computed in step (1). Shortest paths from the waypoints to the targets can be obtained in a similar way. For that purpose, the directions of the edges in the network must be reversed. Then Dijkstra's algorithm is applied n times with the n targets being the source vertices. The resulting shortest-path-trees contain the desired paths. The schematic representation in Fig. 4 sketches parts of two shortest-path-trees rooted at the release point of a UAV (solid lines) and at a target (dashed lines), respectively, and a few waypoints.

The question remains which of the waypoints are suitable. A waypoint will be denoted *viable* if the length of the associated flight path is between $L_{\min} = T \cdot v_{\min}$ and $L_{\max} = T \cdot v_{\max}$, where $v_{\min}$ and $v_{\max}$ are the minimal

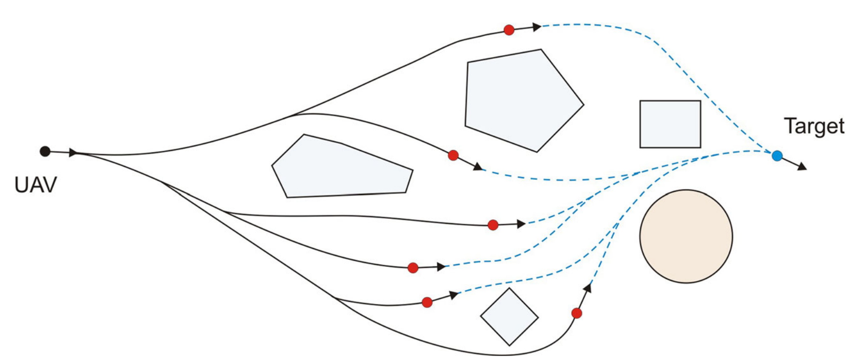


Fig. 4 Flight paths from release point to target via selected waypoints

and maximal velocity of the air vehicles, respectively. Each of these paths allows to realize the mission time T. If the path is too long then the velocity is increased, otherwise it is reduced. In the algorithm, the viable waypoints are ordered according to the gap between the associated path lengths and the length L of the reference path. The first path in the list needs the least velocity adaptation. If this path turns out to be safe then it is used. In case of a risk of collision the path is rejected and the next path from the list is analyzed.

For a more formal description of step (3), assume w.l.o.g. that the first UAV produces the critical path. The parametrized curve of the critical path will be denoted by $\gamma_1(t)$, i.e. $\gamma_1(t) = (x_1(t), y_1(t))$ is the location of the UAV at time t, with $0 \le t \le T$, and

$$\int_0^T \|\dot{\gamma}_1(t)\| \; dt = L \quad \text{and} \quad \|\dot{\gamma}_1(t)\| = v_0.$$

$\gamma_1(t)$ remains unchanged. For the other air vehicles, curves $\gamma_2(t)$, ..., $\gamma_n(t)$ have to be determined realizing mission time T and guaranteeing a safe flight. Let i^* refer to the target assigned to the i-th UAV. Let further $dist_i(p)$ denote the flight distance between the start position of the i-th UAV and a specific waypoint p, and $dist_{i^*}(p)$ the flight distance between p and the target i^*. The associated flight paths are stored in the shortest-path-trees of i and i^*, respectively. The length of the shortest path via p is then $L(p) = dist_i(p) + dist_{i^*}(p)$.

Flight path adaptation and coordination

for $i = 2, ..., n$
  Identify all viable waypoints p fulfilling $L_{min} \le L(p) \le L_{max}$. Sort the viable waypoints p according to increasing deviations $|L(p) - L|$ from the length L of the reference path and store them in a list W.
  while W is not empty
    Remove the first waypoint p from the list. Let P be the path from start position i to target i^* via waypoint p.
    Adapt the velocity of P to meet the mission time T, i.e. set $v = L(p)/T$. Let $\gamma(t)$ with $0 \le t \le T$ denote the associated parametrized curve.
    Check safety of $\gamma(t)$ with respect to $\gamma_1(t)$, ..., $\gamma_{i-1}(t)$.
    if $\gamma(t)$ is safe *then* $\gamma_i(t) = \gamma(t)$, break.
  if no safe path $\gamma_i(t)$ has been found *then* return (no solution).

The following procedure checks whether there is a risk of collision between two air vehicles flying along two curves $\gamma_i(t)$ and $\gamma_j(t)$. A risk occurs if the distance falls below a safety margin d_{safe}. The curves are discretized using a time step Δt and traversed simultaneously. The notation $\|.\|$ stands for the Euclidean norm.

Check safety

safe = *true* , $t = 0$
while $t < T$
  $t = t + \Delta t$
  $d = \|\gamma_i(t) - \gamma_j(t)\|$
  if $d < d_{safe}$
  then *safe* = *false* return.

The algorithm operates successfully except in notoriously hard scenarios with a large number of obstacles lying close together, along with a large safety distance between the air vehicles. In such exceptional cases, if no solution is found, the following strategies often resolve the problem:

- Restart of the algorithm. Due to the probabilistic nature of the network generation there is a good chance to find a solution in further runs.
- Change of the network resolution. Even slightly increasing the number of Mikado jackstraws significantly increases the number of potential flight paths, and with that the chance to find a solution.
- Change of the assignment of UAVs to targets. Except for the reference path, the assignment of UAVs to targets is not unique in most cases. Using a different assignment may lead to a solution.
- Check whether the safety margin can be decreased to some extent without jeopardizing the mission. If the safety distance between aircraft is chosen too large then, in complex scenarios, it may happen that no solution exists.

3.3 Sequential Arrival

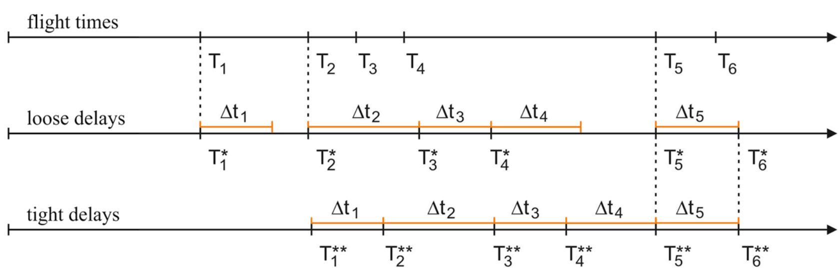
The second type of mission deals with a group of n air vehicles with multiple tasks assigned to a common destination. Each vehicle is able to perform each task. The air vehicles are supposed to arrive sequentially with specified time delays. The delays can be either loose or tight depending on whether a minimum or a precisely defined period of time must elapse between the arrivals. The objective is to find flyable, feasible and safe flight paths such that the total mission time is as short as possible.

The basic idea can be summarized as follows.

Algorithm II

(1) *Individual flight path planning*
For each UAV compute a shortest flight path to the target.
(2) *Sequence determination*
Sort the flight paths of the UAVs according to increasing lengths. Let $T_1, ..., T_n$ denote the flight times of the sorted paths under nominal velocity.

Fig. 5 Adjustment of flight times



(3) *Flight path adaptation and coordination*
Adjust the flight times $T_1, ..., T_n$ to comply with the desired time delays.
Manipulate the flight paths to meet the adjusted flight times and resolve conflicts arising from UAVs coming too close to each other.

Let $\Delta t_1, \Delta t_2, ..., \Delta t_{n-1}$ denote the time delays between the arrivals, i.e. the second UAV should arrive at least (or precisely) Δt_1 after the first UAV, the third UAV at least (or precisely) Δt_2 after the second UAV, etc. Let further $T_1^*, ..., T_n^*$ denote the adjusted flight times with loose time delays and $T_1^{**}, ..., T_n^{**}$ the adjusted flight times with tight time delays. The flight times are determined by the following procedures.

Adjust flight times

(a) *Loose time delays*
$T_1^* = T_1$
for $i = 1, ..., n-1$
  if $T_{i+1} < T_i^* + \Delta t_i$
  then $T_{i+1}^* = T_i^* + \Delta t_i$
  else $T_{i+1}^* = T_{i+1}$

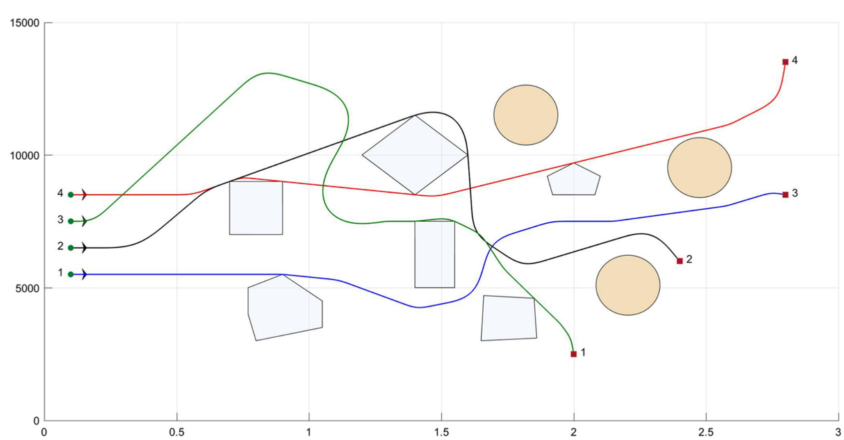
(b) *Tight time delays*
$T_n^{**} = T_n^*$
for $i = n-1, ..., 1$
  if $T_i^* + \Delta t_i < T_{i+1}^{**}$
  then $T_i^{**} = T_{i+1}^{**} - \Delta t_i$
  else $T_i^{**} = T_i^*$

The principle is illustrated by an example in Fig. 5. To realize loose time delays, we have to process the flight paths in the order defined in step (2) of the algorithm. The flight time of the first UAV remains unchanged, i.e. $T_1^* = T_1$. If $T_{i+1} \geq T_i^* + \Delta t_i$ then the flight time of the $(i+1)$-th UAV remains unchanged, i.e. $T_{i+1}^* = T_{i+1}$, since a sufficiently large period elapses between arrival of the i-th and the $(i+1)$-th UAV. Otherwise, the flight time of the $(i+1)$-th UAV has to be shifted to the right to meet the desired time delay Δt_i, i.e. $T_{i+1}^* = T_i^* + \Delta t_i$.

To realize tight time delays, we additionally have to perform part (b). The flight paths are processed in the reversed order. The flight time of the last UAV remains unchanged, i.e. $T_n^{**} = T_n^*$. Whenever there is some idle time between the arrival of the i-th and $(i+1)$-th UAV, i.e. $T_i^* + \Delta t_i < T_{i+1}^{**}$, then the flight time of the i-th UAV has to be shifted to the right to precisely meet the time delay Δt_i.

Flight paths are prolongated as described before by inserting suitable waypoints. In order to adjust the i-th flight path to the new flight time T_i^* or T_i^{**}, the waypoints are sorted according to the deviation $|L(p) - L_i|$, where

Fig. 6 Engagement of UAVs against different targets



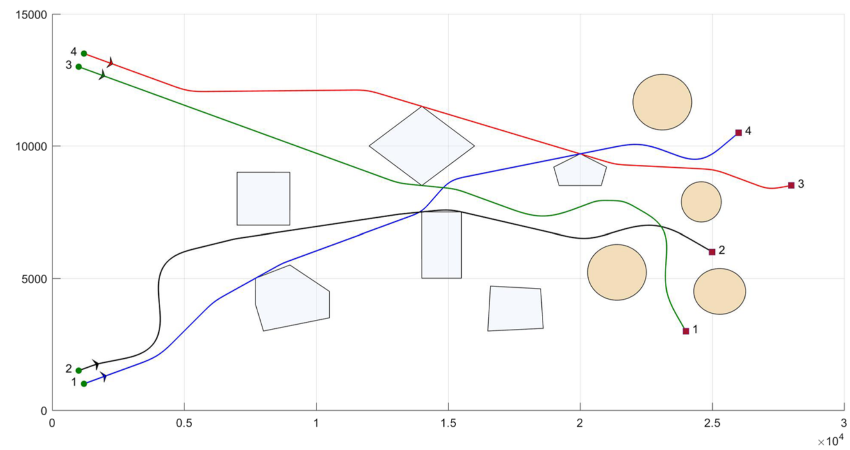


Fig. 7 Engagement of UAVs released from two directions

$L_i = T_i^* \cdot v_0$ or $L_i = T_i^{**} \cdot v_0$ is the length of the required path under nominal velocity v_0, and $L(p)$ is the length of the flight path passing the waypoint p. We select the first waypoint in the list guaranteeing a safe flight. Minor deviations from the desired flight time T_i^* or T_i^{**} are compensated by adapting the velocity of the UAV. The total mission time is the flight time T_n^* or T_n^{**} of the last UAV.

4 Performance Analysis

The algorithms have been implemented and tested in Matlab R2017a on a standard PC running Windows 10 with Intel Core i7-6600U 2.6 GHz CPU and 16 GB RAM.

In the following test cases, the air vehicles are supposed to arrive simultaneously at the targets. All air vehicles have a nominal velocity of 100 meters per second. Deviations of the actual velocity from the nominal velocity must not exceed 5%. The lateral acceleration is restricted to 10 meters per second squared. The size of the operational area is 30 km × 15 km. The scenarios contain forbidden areas and obstacles (plotted as polygons) and threats (plotted as circles). The planning process is based on 300 Mikado jackstraws resulting in a network with approximately 110.000 vertices and 160.000 edges. The transitions between crossing lines are realized by two types of maneuvers, a strong one consisting of two clothoids with maximal lateral acceleration of 10 m/s^2 and a weak one with lateral acceleration 5 m/s^2.

The first test case consists of four air vehicles advancing in formation flight from the west and four widely scattered targets with predefined approach directions. Figure 6 shows the flight paths calculated by the algorithm. The safety distance between the air vehicles has been specified as 350 m. The red path from the uppermost UAV to the uppermost target is the reference path (critical path). The length of the path is 28.756 m. The lengths of the other three prolongated paths differ from the reference path by less than 10 m. The deviations are indeed negligible and make it unnecessary to modify the velocity of the vehicles. This represents the typical performance of the algorithm, also for other similar scenarios.

The second test case contains other targets and threats. The air vehicles are approaching from different directions,

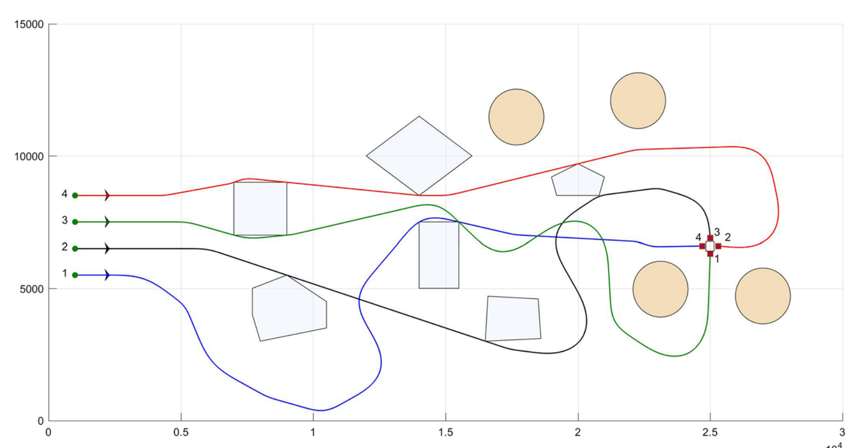


Fig. 8 Engagement of UAVs against single target

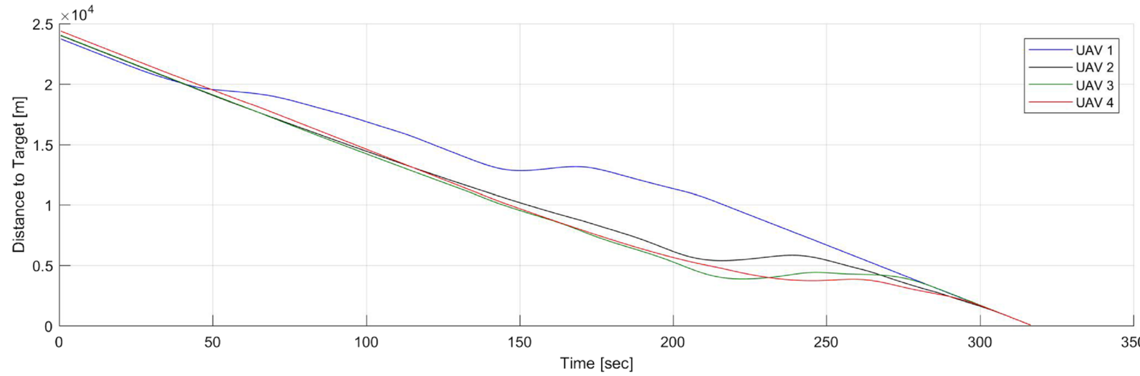


Fig. 9 Distance of UAVs to target vs. time

two of them from northwest and the other two from southwest, see Fig. 7. The red path from UAV #4 to target #3 is critical with path length 27.578 m. The lengths of the other prolongated paths deviate by less than 8 m.

The third scenario in Fig. 8 illustrates a fleet of air vehicles combating a high value target. The target must be approached from different directions (north, south, east, and west). The critical path is the red path belonging to UAV #4. The length is 31.654 m. The deviation of the other three path lengths is less than 7 m. Figure 9 shows the distance of the air vehicles to the target. The four vehicles simultaneously arrive at the target after 316 seconds of flight. For instance, the distance of UAV #1 to the target slightly increases after 150 seconds since the vehicle flies in a loop to prolongate the path.

Figure 10 shows the engagement of a fleet of six UAVs against three targets. Three air vehicles are approaching from northwest, the other three from southwest. Each target must be attacked by two air vehicles. The critical path is the red path from UAV #6 to the rightmost target. In contrast to the previous test cases there is a perceptible deviation of the path lengths. The reason is simply that more UAVs cause more possible interactions and narrow the range of possible routes. The length of the critical path is 26.721 m. The maximal deviation of the other path lengths is 523 m or 1.9%. The deviation is compensated by appropriately adapting the velocity of the air vehicles.

Generally, the path planning problem gets more difficult with increasing number of UAVs. In our experience, with up to four UAVs the maximal deviation of the path lengths does not exceed 0.1%. For six air vehicles, the maximal deviation is smaller than 2%. One option to reduce the deviation is to increase the number of Mikado jackstraws and hence the size of the network, resulting in a significantly larger number of potential paths. This is, of course, at the expense of increased computer running time. Another option is to apply the algorithm several times. Due to the random nature of the network generation, multiple runs of the algorithm provide different results.

A further aspect is the safety distance between the air vehicles. Increasing the safety distance substantially increases the difficulty of finding collision-free paths. For example, in the scenario shown in Fig. 10, changing the safety distance from 350 m to 500 m produces deviations of the path lengths of up to 2.5%. The running time of the algorithm for a network generated from 300 Mikado jackstraws is 30-85 seconds, depending on the number of involved UAVs. Increasing the number of lines to 400

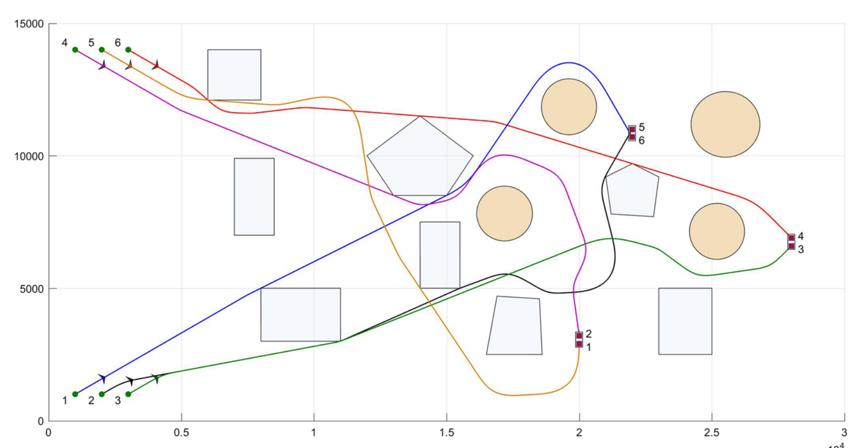


Fig. 10 Engagement of a fleet of UAVs against three targets

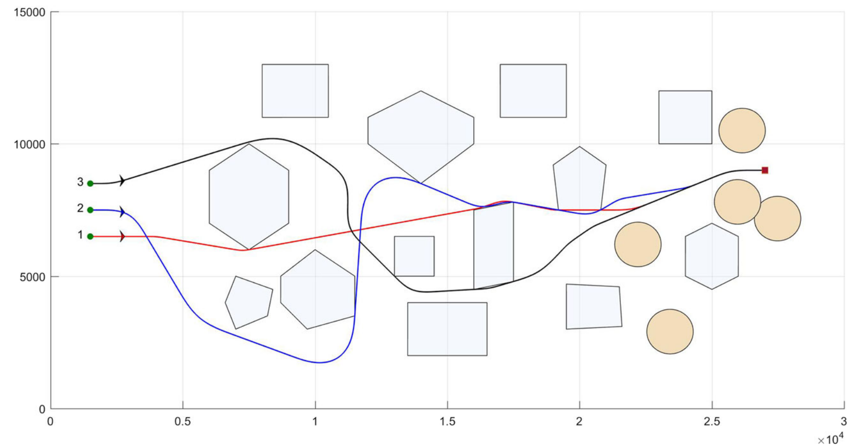


Fig. 11 Sequential arrival of a group of UAVs

provides a network with approximately 190.000 vertices and 270.000 edges, thereby roughly doubling the running time of the algorithm. On the other hand, the increased effort provides only minor improvements of the solutions.

The evaluation is completed by a representative test case for sequential arrival of UAVs at a common target. The scenario illustrated in Fig. 11 is densely occupied with obstacles and threats. The safety distance between the air vehicles has been specified as 400 m. The time delays must be tight with $\Delta t_1 = \Delta t_2 = 40$ s. The lengths of the flight path in the solution are 26.068 m, 30.069 m and 34.066 m respectively, i.e. the deviations from the desired lengths are smaller than 3 m. The distance of the air vehicles to the target as a function of the flight time is shown in Fig. 12.

5 Conclusion

Cooperation of air vehicles is crucial for conducting successful missions in surveillance and combat scenarios. The main issues of mission planning include the assignment of UAVs to targets or tasks and the generation of suitable flight paths. Due to spatial constraints, timing constraints, and limited flight capabilities, these are difficult and strongly coupled optimization problems.

This paper discusses the engagement of a group of UAVs which is supposed to arrive at the targets simultaneously or sequentially with specified time delays. The planning must consider threats and obstacles, as well as flight characteristics and the risk of mutual

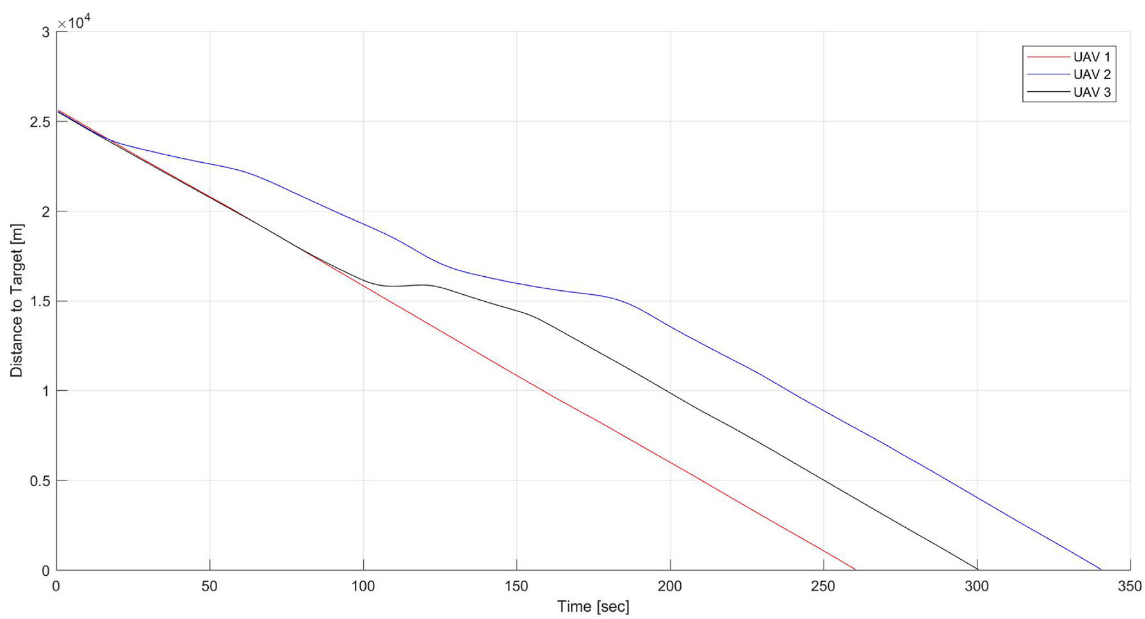


Fig. 12 Distance of UAVs to target vs. time

collision of UAVs, and must minimize the total mission time. The presented algorithms do not decouple task assignment and flight path planning but solve the problems concurrently. Task assignment uses a formulation as a linear bottleneck assignment problem with costs of the tasks corresponding to lengths of shortest flight paths. For sequential arrival, the assignment depends on a sorting of the path lengths. Flight paths of prescribed lengths are generated by concatenating subpaths leading through intermediate waypoints. A sophisticated network guarantees all flight paths to be feasible and flyable and allows to find safe paths without the risk of mutual collision of UAVs.

The algorithms have been designed for UAVs of the same type. However, they can easily be extended for a heterogeneous fleet of vehicles with different velocities and lateral accelerations. For this purpose, several networks have to be generated, each one adapted to the maneuverability of the special type of aircraft.

References

1. Ahuja, R., Kumar, A., Jha, K., Orlin, B.: Exact and heuristic algorithms for the Weapon-Target assignment problem. Oper. Res. **55**(6), 1136–1146 (2007)
2. Alighanbari, M., Kuwata, Y., How, J.: Coordination and control of multiple UAVs with timing constraints and loitering. In: Proceedings of the American control conference, vol. 6, pp. 5311-5316 (2003)
3. Babel, L.: Three-Dimensional Route planning for unmanned aerial vehicles in a risk environment. Journal of Intelligent & Robotic Systems **71**(2), 255–269 (2013)
4. Babel, L.: Flight path optimization with application to in-flight replanning to changing destinations. To appear in aircraft engineering and aerospace technology (2018)
5. Bellingham, J., Tillerson, M., Richards, A., How, J.: Multi-task allocation and path planning for cooperating UAVs. In: Cooperative control: models, applications and algorithms. Conference on coordination, control and optimization, pp. 1-19 (2001)
6. Bellingham, J., Richards, A., How, J.: Receding horizon control of autonomous aerial vehicles. In: Proceedings of the 2002 American control conference, Vol.5, pp.3741-3746 (2002)
7. Bellingham, J., Tillerson, M., Alighanbari, M., How, J.: Cooperative path planning for multiple uavs in dynamic and uncertain environments. In: Proceedings of the 41st IEEE conference on decision and control, pp. 2816-2822 (2002)
8. Bertuccelli, L., How, J.: Robust UAV search for environments with imprecise probability maps. In: Proceedings of IEEE Conference of decision and control, pp.5680-5685 (2005)
9. Bortoff, S.A.: Path planning for UAVs. In: Proceedings of the 2000 American control conference, vol. 1, pp. 364-368 (2000)
10. Burkard, R., Dell'Amico, M., Martello, S.: Assignment problems - revised reprint. SIAM, Philadelphia (2012)
11. Chandler, P.R., Rasmussen, S., Pachter, M.: UAV Cooperative path planning. In: Proceedings of AIAA Guidance, Navigation and Control Conference. AIAA-2000-4370, Denver, Co, pp. 1255-1265 (2000)
12. Chazelle, B.: Approximation and Decomposition of Shapes. In: Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics. Schwartz J.T., Yap C.K., Eds., Lawrence Erlbaum Associates, Pp. 145-185 (1987)
13. Choset, H.: Sensor based motion planning: The hierarchical generalized voronoi graph. California institute of technology, Ph.D thesis (1996)
14. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. Introduction to Algorithms, 3rd. MIT Press, Cambridge (2009)
15. De Filippis, L., Guglieri, G., Quagliotti, F.: Path Planning Strategies for UAVs in 3D Environments. Journal of Intelligent & Robotic Systems **65**(1), 247–264 (2012)
16. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
17. Dogan, A.: Probabilistic Path Planning for UAVs. In: Proceedings of 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations. San Diego, CA (2003)
18. Eun, Y., Bang, H.: Cooperative Task Assignment/Path Planning of Multiple Unmanned Aerial Vehicles using Genetic Algorithms. J. Aircr. **46**(1), 338–342 (2009)
19. Ferguson, D., Stentz, A.: Using interpolation to improve path planning: The field d* algorithm. Journal of Field Robotics **23**(2), 79–101 (2006)
20. Fulkerson, D., Glicksberg, I., Gross, O.: A production line assignment problem. Technical report RM-1102. The rand corporation, santa monica CA (1953)
21. Goerzen, C., Kong, Z., Mettler, B.: A survey of motion planning algorithms from the perspective of autonomous UAV guidance. Journal of Intelligent & Robotic Systems **57**(1), 65–100 (2010)
22. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on System Science and Cybernetics **4**(2), 100–107 (1968)
23. Jun, M., D'Andrea, R.: Path planning for unmanned aerial vehicles in uncertain and adversarial environments. Cooperative control: Models. Applications and Algorithms **1**, 95–110 (2003)
24. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**(4), 556–580 (1996)
25. Kim, J.O., Khosla, P.K.: Real-Time Obstacle avoidance using harmonic potential functions. IEEE Trans. Robot. Autom. **8**(3), 338–349 (1992)
26. Koenig, S., Likhachev, M.: Incremental A*. In: Advances in Neural Information Processing Systems, Pp. 1539-1546 (2002)
27. Koenig, S., Likhachev, M.: D*-lite. In: Proceedings of the AAAI Conference of Artificial Intelligence, pp. 476-483 (2002)
28. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Boston (1991)
29. LaValle, S.M.: Planning algorithms cambridge university press (2006)
30. LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. Int. J. Robot. Res. **20**(5), 378–400 (2001)
31. Lingelbach, F.: Path Planning Using Probabilistic Cell Decomposition. Ph.D.Thesis. KTH Signaler Sensorer och System, Stockholm (2005)
32. Maddula, T., Minai, A., Polycarpou, M.: Multi-Target Assignment and Path Planning for Groups of UAVs. In: Cooperative Systems: Recent Developments in Cooperative Control and Optimization, Pp. 261-272 (2004)
33. McLain, T., Beard, R.: Trajectory planning for coordinated rendezvous of unmanned air vehicles. In: Proceedings of the AAAI Conference on Guidance, Navigation and Control Conference, pp. 1-8 (2000)
34. Melor, C., Omar, R., Sabudin, E.: Recent Research in Cooperative Path Planning Algorithms for Multi-Agent using Mixed-Integer Linear Programming. Journal of Engineering and Applied Sciences **22**(4), 8921–8926 (2016)

35. Murphey, R.: An Approximate Algorithm for a Weapon Target Assignment Stochastic Program. In: Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems. Kluwer Academic Publishers, Boston (1999)
36. Nash, A., Daniel, K., Koenig, S., Felner, A.: Theta*: Any-Angle path planning on grids. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1177-1183 (2007)
37. Niewola, A., Podsedkowski, L.: L* algorithm - a linear computational complexity graph searching algorithm for path planning. Journal of Intelligent & Robotic Systems. Online ISSN **1573-0409**, 1–20 (2017)
38. Pfeiffer, B., Batta, R., Klamroth, K., Nagi, R.: Path Planning for UAVs in the Presence of Threat Zones Using Probabilistic Modeling. In: Handbook of Military Industrial Engineering. Taylor and Francis, USA (2008)
39. Schouwenaars, T., De Moor, B., Feron, E., How, J.: Mixed integer programming for Multi-Vehicle path planning. In: Proceedings European Control Conference, pp. 2603-2608 (2001)
40. Schumacher, C., Chandler, P., Pachter, M., Pachter, L.: Optimization of Air Vehicles Operations using Mixed-Integer Linear Programming. J. Oper. Res. Soc. **58**(4), 516–527 (2007)
41. Shanmugavel, M., Tsourdos, A., White, B., Zbikowski, R.: Cooperative Path Planning of Multiple UAVs using Dubins Paths with Clothoid Arcs. Control. Eng. Pract. **18**, 1084–1092 (2010)
42. Shima, T., Rasmussen, J., Sparks, A.: UAV Cooperative Multiple Task Assignments Using Genetic Algorithms. American Control Conference, pp. 2989–2994 (2005)
43. Stentz, A.: The focussed d* algorithm for Real-Time replanning. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1652-1659 (1995)
44. Waydo, S., Murray, R.M.: Vehicle Motion Planning using Stream Functions. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 2484-2491 (2003)
45. Weinstein, A.L., Schumacher, C.: UAV Scheduling via the vehicle routing problem with time windows. In: Proceedings of the AIAA Infotech Aerospace Conference and Exhibit, pp. 2839 (2007)

Luitpold Babel studied mathematics and received his diploma and doctorate degree from the Technical University Munich, Germany in 1990. Following research positions at the Old Dominion University, Norfolk, USA and University Graz, Austria, he completed his habilitation in 1997. The focus of his work was on problems of combinatorial optimization, graph theory and efficient algorithms. From 1999 to 2006 he joined the missile industry where he worked in research and development teams of different subsidiaries of EADS and MBDA. Since 2007 he is a professor of mathematics and computer science at the Universität der Bundeswehr in Munich, Germany. His research interests include mathematical methods in defence analyses, route planning of autonomous systems, and mission planning systems for missiles and unmanned air vehicles.