Real-Time Optimal Time-Critical Target Assignment for UAVs*

Yoonsoo Kim, Da-Wei Gu, and Ian Postlethwaite

Department of Engineering, University of Leicester, Leicester, LE1 7RH, U.K.
{yk17, dag, ixp}@le.ac.uk

Abstract. In the literature, e.g. [10], one can find the so-called *basic* UAV mission target assignment in which m UAVs each with a capacity limit q visit n targets in a cooperative manner (and return to their departure points) such that the cost incurred by each UAV's travel is minimized. In [10], we proposed a mixed integer linear program (MILP) formulation which exactly solves the problem, as well as four alternative MILP formulations which are computationally less intensive (and therefore suited for real-time purposes) yet yield a theoretically guaranteed sub-optimal solution. In this chapter, we further consider *timing constraints* imposed on some p of the targets, so-called *prime targets*. This consideration is often required for scenarios in which prime targets must be visited in a pre-defined time interval, and mathematically results in the addition of several integer linear constraints to the previous MILP formulation making the problem computationally intractable. We propose a novel procedure of adding these cumbersome timing constraints to the previous MILP formulation, in order to avoid increasing too much computational cost under practically valid assumptions. We first show that the proposed procedure still guarantees the previously claimed theoretical solution quality associated with the basic mission. We then show through extensive numerical simulations that under certain conditions, our algorithms return solutions which are still computationally manageable.

Keywords: Unmanned aerial vehicles (UAV); target assignment; mixed integer linear program; timing constraints.

1 Introduction and Problem Statement

In [10], for a given number m of UAVs U_i ($i = 1, 2, \ldots, m$, $m \geq 2$) at corresponding positions T_0^i, and a number n of targets T_j ($j = 1, 2, \ldots, n$, $n \geq m$) within a terrain $\mathbf{X}$, we consider a mission in which the UAVs visit all the targets in a cooperative manner (and return to where they departed from) such that the cost (reflecting UAV operating time and risk) incurred by each UAV's travel is

* This research has been supported by the UK Engineering and Physical Sciences Research Council and BAE Systems. The authors truly appreciate the valuable comments and suggestions by the editors and anonymous reviewers.

M.J. Hirsch et al. (Eds.): Adv. in Cooper. Ctrl. & Optimization, LNCIS 369, pp. 265–280, 2007.
springerlink.com

minimized whilst keeping the number of targets visited by a single UAV below a certain limit q. More precisely, we would like to calculate $\mathbf{T}^*$

$$\mathbf{T}^* \equiv \min_{j} \max_{\mathcal{A}(j)_i} \mathbf{T}(j)_i^*, \tag{1}$$

where $\mathbf{T}^*$ is the least maximum cost among all UAVs in visiting their assigned targets (and returning to their departure points), $\mathcal{A}$ is the set of feasible target assignments to UAVs, $\mathcal{A}(j)$ ($\in \mathcal{A}$) is one of the feasible assignments, $\mathcal{A}(j)_i$ is the sub-assignment given to the ith UAV within $\mathcal{A}(j)$ and finally $\mathbf{T}(j)_i^*$ is the optimal cost of completing the sub-assignment $\mathcal{A}(j)_i$ by the ith UAV. We note that the objective functional to be minimized is appropriate for balancing workloads across UAVs. In this chapter, we further consider the same problem with practical *timing constraints*. This is basically due to the frequent presence of so-called *prime targets* that must be visited in a fixed time interval in many UAV applications. As a result, we require that the solution assignment to (1) be chosen such that a UAV visits a prime target T_k within a given time window $[t_k^\alpha, t_k^\beta]$.[1] In addition, the total number of prime targets is limited by p, and the maximum number of prime targets which a UAV is capable of handling is limited by q' ($\leq q$), in order to increase the probability of mission success.

There is a large number of papers dealing with various target assignment problems. These include Weapon-target assignment [1,14], timetabling [20,22], the celebrated Travelling Salesman Problem [19] and more generally capacity-limited vehicle routing problems [9,17]. We note that these problems are slightly different from the problem in the present context, in that (i) we may not require UAVs to return to their starting positions; (ii) we minimize the individual *tour* cost for balanced workload, not the total cost incurred by the whole mission; (iii) UAVs do not necessarily depart from the same depot. There is also much literature available on coordinated target assignment of UAVs, for example [2,3,4,5], and some of which add the timing and precedence constraints to the original problem [3,11,15,16]. As the underlying problem is known to be NP-hard, it is often fruitless to approach the problem in a direct or exact manner. Nevertheless, as many papers have shown, direct MILP formulations offer a promising way forward in terms of providing an optimal solution to the problem in spite of the computational demands [3,22]. As an example of the MILP approaches, we note the petal algorithm introduced in [2,5]. It considers all feasible task assignments (so-called petals), i.e. identifies all possible sequences of waypoints, for every UAV subject to its capabilities, and subsequently constructs the shortest paths connecting the waypoints as well as avoiding threats. Then, a MILP formulation is employed to find the best assignment in terms of the underlying cost. As implied by the numerical tests shown in [5], the petal algorithm becomes computationally problematic for a large number of targets, e.g. $n > 12$, due to the exhaustive consideration of all feasible petals. As an alternative method,

[1] In the literature, one may find *precedence constraints*, i.e. some target must be visited before other targets. However, we here assume that this can be viewed as a special case of the aforementioned timing constraints.

tabu search based methods are also useful for this purpose [17,18]. When time is critical, heuristic or non-exact methods have been considered, even if global optimality may not be achieved [4,9]. Among many heuristics, we note the *Iterated Optimal Tour Partitioning* (IOTP) algorithm proposed in [9,13], mainly for multi-vehicle-single-depot routing with capacity constraints. With the IOTP algorithm, it is claimed that one can obtain a tour whose cost is at most $2 - 1/q$ times of the optimal tour cost, where q is the capacity of vehicles.

In conclusion, what would be desirable is a direct MILP formulation combined with a non-exact method in such a way that the advantages of each are enjoyed. As inspired by the solution strategy introduced in [10] for the target assignment problem without timing constraints, we reduce the possibly large MILP associated with the original time-critical target assignment problem down to smaller MILPs while minimizing the loss of optimality, followed by solving the smaller MILPs exactly. Since this approach involves only small MILPs, it is therefore computationally tractable. We note that a similar approach has been recently proposed in [21] in which the problem in question is interpreted as a mixed version of the minimum cost network flow problem and the travelling salesman problem. The two problems are then approached by a promising heuristic employing linear programming, MILP and tabu search in order to reduce the computational complexity. However, no formal analysis on the algorithm performance, for example something like (16), is given, and therefore the algorithm may not be suitable for particular problem parameters. Our main challenges (contributions) are to ensure the previously claimed performance bounds in [10] and to allow only a slight increase of computational cost even after adding timing constraints. To this end, we first briefly describe two algorithms, which were introduced in [10], and their performance bounds in Section 2.1. We then show in Section 2.2 and 2.3 how the timing constraints in question can be incorporated into the existing MILPs with the performance bounds unchanged. In Section 3, we examine the computational aspect of our modified algorithms through extensive simulations. Concluding remarks are presented in Section 4.

2 Algorithms

2.1 Two Algorithms

The two algorithms (denoted by $\mathbf{H}_2$ and $\mathbf{H}_3$) introduced in [10] need a feasible network of UAVs' flying routes as their input. One can create such a feasible network of UAVs' flying routes by defining significant waypoints and links connecting the waypoints, and assigning cost (again, reflecting UAV operating time and risk) to the links. For illustration, Fig. 1 shows two UAV starting positions (T_0^1 and T_0^2), two targets (T_1 and T_2) and two obstacles (dashed objects). In order to identify significant waypoints, the two obstacles are approximated by two rectangles. The corner points of the rectangles then become part of the set of waypoints along with T_0^1, T_0^2, T_1 and T_2. The feasible network of the two UAVs' flying routes is the set of links connecting the waypoints. Each link carries the cost of travel based on its length and possible risks on it.

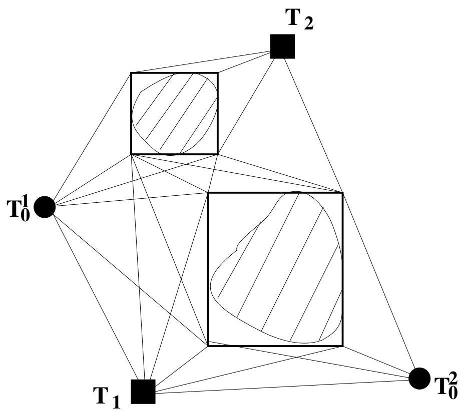


Fig. 1. A rectangular cover of two objects (the dashed area) and the corresponding construction of links

Given the network of UAVs' flying routes, the algorithm $\mathbf{H}_2$ (respectively, $\mathbf{H}_3$) is devised for the problem without (respectively, with) the UAV's return constraint. The basic principle behind the algorithms is that the original problem of possibly large size is handled by two solution steps each of which involves a problem of relatively small size. The first step in the present context solves a MILP for finding m groups of targets such that each group is disjoint and contains exactly one UAV and less than or equal to q targets. The second step finds an optimal order of visiting the assigned targets for each UAV by solving another MILP. These two MILPs are computationally manageable as long as q is small (≤ 4). The first grouping step is crucial in terms of both solution quality and computational complexity. The algorithms $\mathbf{H}_2$ and $\mathbf{H}_3$ employ the objective functionals which minimize the T_j-to-T_k cost and the T_0^i-to-T_j-to-T_k cost for each j, k, respectively, where T_j or T_k is the starting position of the ith UAV or a target position to be covered by the ith UAV. The following are the formal descriptions of $\mathbf{H}_2$ and $\mathbf{H}_3$. For detailed explanation on the constraints in the MILPs below, see [10].

Step 1. *Sub-optimal partitioning*: Consider an optimization problem $\mathcal{F}_2$ (respectively, $\mathcal{F}_3$), as shown below, that solves for x_{ij} ($x_{ij} = 1$ if the jth target is assigned to the ith UAV) to partition the underlying set of targets into m subsets $\mathcal{T}_i$ ($i = 1, 2, \ldots, m$) such that (i) each $\mathcal{T}_i$ contains at most q elements; (ii) the travelling cost T_j-to-T_k (respectively, T_0^i-to-T_j-to-T_k) for each j, k is minimized, where T_j or T_k is the starting position of the ith UAV or belongs to $\mathcal{T}_i$; (iii) each target is covered by exactly one UAV.

Step 2. *Optimal path-planning*: For each $\mathcal{T}_i$, consider $\mathcal{F}_E$ (respectively, $\mathcal{F}_{Eret}$), as shown below, that solves for a_{ij}^k ($a_{ij}^k = 1$ if the ith UAV visits the kth target after $j-1$ targets, so that the ith UAV visits $T_{x_{i1}}, T_{x_{i2}}, \ldots$ in turn, where

$x_{ij} = \sum_{k=1}^{n} a_{ij}^k$) to obtain the optimal path of visiting all the targets contained in $\mathcal{T}_i$ by the ith UAV.

$\mathcal{F}_E$: minimize r

subject to

$$\sum_{k=1}^{n} a_{ij}^k \leq 1 \quad \forall i, j \tag{2}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{q} a_{ij}^k = 1 \quad \forall k \tag{3}$$

$$a_{ij}^k \in \{0, 1\} \quad \forall i, j, k \tag{4}$$

$$\sum_{k=1}^{n} a_{i(j+1)}^k \leq \sum_{k=1}^{n} a_{ij}^k \quad \forall i, j \tag{5}$$

$$a_{ij}^v + a_{i(j+1)}^w + a_{ij}^w + a_{i(j+1)}^v = 2\, y_{i\eta(v,w)}^j + \tilde{y}_{i\eta(v,w)}^j \quad \forall i, j, v, w \tag{6}$$

$$y_{i\eta(v,w)} = \sum_j y_{i\eta(v,w)}^j \quad \forall i, v, w \tag{7}$$

$$y_{i\eta(v,w)}^j \in \{0, 1\}, \quad \tilde{y}_{i\eta(v,w)}^j \in [0, 1] \quad \forall i, j, v, w \tag{8}$$

$$\sum_{k=1}^{n} C_0(i,k)\, a_{i1}^k + \sum_{v=1}^{n-1} \sum_{w=v+1}^{n} c(\eta(v,w))\, y_{i\eta(v,w)} \leq r \quad \forall i \tag{9}$$

where $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, q\}$ for (2)-(4) or $j \in \{1, 2, \ldots, q-1\}$ for (5)-(8), $k \in \{1, 2, \ldots, n\}$, v, $w \in \{1, 2, \ldots, n\}$ $(v < w)$, $\eta(v,w) = (v-1)n - v(v-1)/2 + w - v$. In (9), $C_0(i,k)$ (respectively, $c(\eta(v,w))$) is the travelling cost from T_0^i to T_k (respectively, from T_v to T_w).

$\mathcal{F}_{Eret}$: minimize r

subject to (2)–(8) and

$$a_{i1}^k = b_{i1}^k \quad \forall i, k$$

$$b_{i(j-1)}^k - \sum_{k=1}^{n} a_{ij}^k \leq b_{ij}^k \leq b_{i(j-1)}^k + \sum_{k=1}^{n} a_{ij}^k \quad \forall i, j, k$$

$$a_{ij}^k \leq b_{ij}^k \leq a_{ij}^k + (1 - \sum_{k=1}^{n} a_{ij}^k) \quad \forall i, j, k$$

$$\sum_{k=1}^{n} C_0(i,k)\, (a_{i1}^k + b_{iq}^k) + \sum_{v=1}^{n-1} \sum_{w=v+1}^{n} c(\eta(v,w))\, y_{i\eta(v,w)} \leq r \quad \forall i$$

where $i \in \{1, 2, \ldots, m\}$, $j \in \{2, 3, \ldots, q\}$, $k \in \{1, 2, \ldots, n\}$.

$\mathcal{F}_2$: minimize r

subject to

$$\sum_{j=1}^{n} x_{ij} \leq q \quad \forall i \tag{10}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \forall j \tag{11}$$

$$y_{i\eta(j,k)} \leq \frac{x_{ij} + x_{ik}}{2} \leq y_{i\eta(j,k)} + \frac{1}{2} \quad \forall i, j, k \ (j < k) \tag{12}$$

$$C_0(i,j)\, x_{ij} \leq r \quad \forall i \tag{13}$$

$$c(\eta(j,k))\, y_{i\eta(j,k)} \leq r \quad \forall i, j, k \ (j < k) \tag{14}$$

$$x_{ij},\, y_{i\eta(j,k)} \in \{0,\, 1\} \quad \forall i, j, k \ (j < k) \tag{15}$$

where $i \in \{1, 2, \ldots, m\}$, $j,\, k \in \{1, 2, \ldots, n\}$ and $\eta(j,k) = (j-1)n - j(j-1)/2 + k - j$.

$\mathcal{F}_3$: minimize r

subject to

$$\sum_{j=1}^{n} x_{ij} \leq q \quad \forall i$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \forall j$$

$$y_{i\eta} \leq \frac{x_{ij} + x_{ik}}{2} \leq y_{i\eta} + \frac{1}{2} \quad \forall i, j, k \ (j < k)$$

$$2\, C_0(i,j)\, x_{ij} \leq r \quad \forall i, j$$

$$C_0(i,j)\, x_{ij} + c(\eta(j,k))\, y_{i\eta(j,k)} \leq r \quad \forall i, j, k \ (j < k)$$

$$x_{ij},\, y_{i\eta(j,k)} \in \{0,\, 1\} \quad \forall i, j, k \ (j < k)$$

where $i \in \{1, 2, \ldots, m\}$, $j, k, l \in \{1, 2, \ldots, n\}$ and $\eta(j,k) = (j-1)n - j(j-1)/2 + k - j$.

The performance bound for $\mathbf{H}_2$ is obtained by the two facts that (i) $\mathbf{T}^*$ is less than the T_j-to-T_k cost for any i, j; (ii) one can create a UAV's feasible path which sequentially visits all the (at most q) targets within the assigned group from the first grouping step. Similarly, the performance bound for $\mathbf{H}_3$ is due to the facts that (i) $\mathbf{T}^*$ is less than the T_0^i-to-T_j-to-T_k cost for any i, j, k when the UAV's return constraint is imposed; (ii) one can create a feasible path such that a UAV goes back to its departure point every time after it sequentially visits two targets within the associated group. As a result, $\mathbf{H}_2$ and $\mathbf{H}_3$ guarantee

$$1 \leq \frac{\mathbf{T}}{\mathbf{T}^*} \leq q \tag{16}$$

and

$$1 \leq \frac{\mathbf{T}}{\mathbf{T}^*} \leq 2 \left\lceil \frac{q}{3} \right\rceil - \kappa, \tag{17}$$

respectively, where $\mathbf{T}$ is the maximum of actual costs incurred by each UAVs' travel using $\mathbf{H}_2$ or $\mathbf{H}_3$, and $\kappa = 1$ if $q = 3k + 1$ $(k = 0, 1, \ldots)$; otherwise $\kappa = 0$. Note that the performance bound (17) is guaranteed only if every UAV can go back to its departure point every time after visiting two targets, because of the second fact used for deriving the bound. This assumption may fail when timing constraints are imposed in the next section. See the proof of Proposition 1.

2.2 Incorporation of Timing Constraints into the Existing Framework

For incorporation of timing constraints into the existing framework, we need two kinds of T_0^i-to-T_j and T_v-to-T_w costs over the same network of UAVs' flying routes. One kind, denoted by $C_0(i,j)$ and $C(v,w)$, is used for being minimized and reflects both flight time and risk information due to threats. The other kind, denoted by $\underline{C_0}(i,j)$ and $\underline{C}(v,w)$, is used in concert with timing constraints and solely contains flight time information. The latter is computationally not as cumbersome as the former because algorithms to be introduced do not require T_0^i-to-T_j and T_v-to-T_w costs for *all* j, v, w. In fact, for the ith UAV the first grouping step of the new algorithms needs $C_0(i,j)$ and $C(v,w)$ for all j, v and w, but $\underline{C_0}(i,j')$ and $\underline{C}(v',w')$ only for j', v' and w' such that $T_{j'}$, $T_{v'}$ and $T_{w'}$ are prime targets.

Clearly, the first grouping step of $\mathbf{H}_2$ or $\mathbf{H}_3$ must be modified such that prime targets are assigned to a UAV such that the UAV can actually reach the assigned prime targets within the required time intervals. For this purpose, as the targets are grouped by the same technique used for $\mathbf{H}_2$ or $\mathbf{H}_3$, we impose the additional constraint such that each UAV visits its assigned prime targets prior to non-prime targets. As a result, the new grouping step provides each UAV with a feasible assignment accounting for timing constraints, along with an explicit order of visiting prime targets. Note that the additional constraint, which forces each UAV to visit prime targets prior to non-prime targets in the first step, is however neglected in the second step in which the optimal path for each UAV to visit all the assigned targets is computed. The second step is basically the same as before, in the sense that one exactly solves the time-critical target assignment problem but now of small size. For an illustration of the new algorithm, as shown in Fig. 2, suppose the new grouping step returns the assignment such that a UAV at T_0 covers two non-prime targets, T_1 and T_2, and two prime targets, T_3 and T_4, with their associated time windows, $[0, 10]$ and $[0, 50]$, respectively. Then, although the new grouping step directs the UAV to visit T_3 and T_4 (dotted line) prior to T_1 and T_2, the second step disregards the

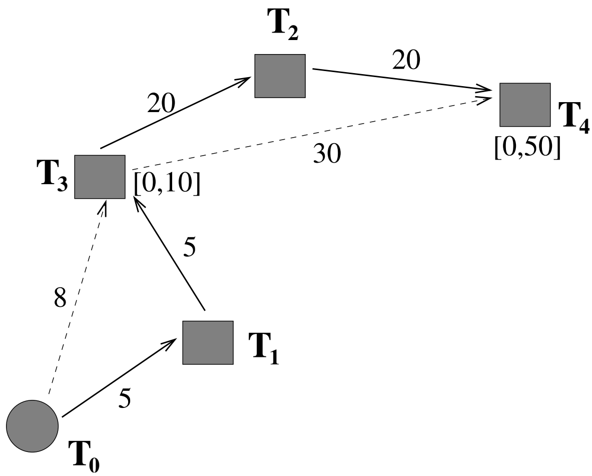


Fig. 2. An illustration of the new algorithm

direction and yields the path (solid line), $T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_4$. This is because T_0 is closer to T_1 than T_3, and also T_3 is closer to T_2 than T_4. Note that this change is allowed because the timing constraints are still not violated. Based on the brief description of the new algorithms $\widetilde{\mathbf{H}}_2$ and $\widetilde{\mathbf{H}}_3$, the following result immediately follows. The formal description of $\widetilde{\mathbf{H}}_2$ will be given later.

Proposition 1. *If $t_k^\alpha = 0$ for every k, the algorithm $\widetilde{\mathbf{H}}_2$ guarantees the bound (16) for every positive integer q', where q' is the maximum number of prime targets visited by a UAV. However, $\widetilde{\mathbf{H}}_3$ guarantees the bound (17) only for $q' < 3$.*

Proof. First note that under the condition that $t_k^\alpha = 0$, no feasible assignments are *lost* by the new grouping step. In fact, for any feasible target T_k assignment to a UAV, one can always construct a feasible path such that the UAV visits prime targets prior to non-prime targets as long as $t_k^\alpha = 0$ for all k. This immediately implies that the new algorithm $\widetilde{\mathbf{H}}_2$ guarantees the same bound as (16). However, when $q' \geq 3$, this violates the underlying assumption used to derive (17) that one can create a feasible path such that a UAV goes back to its departure point every time after the UAV sequentially visits two targets within the associated group. This proves the claim.

As noted before, it is often desirable to have small q' in order to increase the probability of mission success, in which case the new algorithms are still applicable in practice.

2.3 MILPs Including Timing Constraints

One can find the necessary (integer linear) constraints for the grouping with no timing constraints in [10]. We thus focus here on developing methods of converting timing constraints into integer linear constraints.

To begin, let us first define integer grouping variables x_{ij} ($i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$) which represent the relationship between the ith UAV and the jth target, i.e. $x_{ij} = 1$ if the jth target is assigned to the ith UAV; $x_{ij} = 0$ otherwise. For the consideration of timing constraints, we then consider q' *rooms* (numbered from 1 to q') for each UAV. Each room may hold at most one identifier (ID) of prime target and has to be filled in ascending order, so that the resultant path of the ith UAV becomes $T_0^i \to T_{z_1} \to T_{z_2} \ldots \to T_{z_{\tilde{q}'}}$, where z_j ($j \in \{1, 2, \ldots, p\}$) is the ID of the jth prime target and $\tilde{q}' \leq q'$. For this purpose, we define other integer variables a_{ij}^k (similar to the one previously defined in $\mathcal{F}_E$), where i, j and k run from 1 to m, 1 to q' and 1 to p, respectively, in order to capture the relationship between the ith UAV, its jth room and the kth prime target. Similarly, we also define integer variables z_{ij}^k, where i, j and k run from 1 to m, 1 to q' and 1 to n, respectively. The difference between a and z is that a pertains to the ID (numbered from 1 to p) assigned to a prime target amongst only p prime targets, whereas z pertains to the ID (numbered from 1 to n) assigned to a prime target amongst all n targets. This seemingly unnecessary definition of z becomes useful when the connection between a and x need to be made later. The following are the integer linear constraints for implementing the aforementioned verbal expressions:

$$\sum_{k=1}^{p} a_{ij}^k \leq 1 \quad \forall i, j \quad \text{and} \quad \sum_{i=1}^{m} \sum_{j=1}^{q'} a_{ij}^k = 1 \quad \forall k \tag{18}$$

for $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, q'\}$ and $k = \{1, 2, \ldots, p\}$, and

$$\sum_{k=1}^{p} a_{i(j+1)}^k \leq \sum_{k=1}^{p} a_{ij}^k \quad \forall i,\ j \tag{19}$$

for $i = \{1, 2, \ldots, m\}$ and $j = \{1, 2, \ldots, q' - 1\}$.

Next, we consider the flight time from T_0^i to the first room for the ith UAV, and from the jth room to the $(j+1)$th room. The former is simply $\sum_{k=1}^{p} \underline{C_0}(i,k) a_{i1}^k$, and thus we need

$$\sum_{k=1}^{p} t_k^\alpha a_{i1}^k \leq \sum_{k=1}^{p} \underline{C_0}(i,k) a_{i1}^k \leq \sum_{k=1}^{p} t_k^\beta a_{i1}^k$$

where $i \in \{1, 2, \ldots, m\}$, for satisfying the associated timing constraints. However, the latter is not trivial.[2] The MILP expression of the flight time between targets requires the introduction of the additional auxiliary variables $d_{i\eta(v,w)}^j$ and $\tilde{d}_{i\eta(v,w)}^j$ which are defined through the following equality:

$$a_{ij}^v + a_{i(j+1)}^w + a_{ij}^w + a_{i(j+1)}^v = 2\, d_{i\eta(v,w)}^j + \tilde{d}_{i\eta(v,w)}^j \tag{20}$$

$$d_{i\eta(v,w)}^j \in \{0, 1\}, \quad \tilde{d}_{i\eta(v,w)}^j \in [0, 1] \tag{21}$$

[2] The present approach to the latter is similar to the one in [10], but the derivation of the linear inequality constraints corresponding to timing constraints is novel in the chapter.

for all i, j, v, w, where $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, q'-1\}$, $v, w \in \{1, 2, \ldots, p\}$ ($v < w$) and $\eta(v, w) = (v-1)p - v(v-1)/2 + w - v$. The flight time from T_0^i to T_{z_j} (via $T_{z_1}, \ldots, T_{z_{(j-1)}}$) then becomes

$$\sum_{k=1}^{p} \underline{C_0}(i, k) a_{i1}^k + \sum_{u=1}^{j-1} \sum_{v=1}^{p-1} \sum_{w=v+1}^{p} \underline{c}(\eta(v, w))\, d_{i\eta(v,w)}^u$$

where $\underline{c}$ is a one-dimensional form of two-dimensional $\underline{C}$. Equality (20) enables $d_{i\eta(v,w)}^j$ to be 1 only if $a_{ij}^v = a_{i(j+1)}^w = 1$ or $a_{ij}^w = a_{i(j+1)}^v = 1$. In other words, $d_{i\eta(v,w)}^j$ is set to 1 only if the vth and wth targets are assigned to the consecutive jth and $(j+1)$th (or $(j+1)$th and jth) rooms of the ith UAV.

As a result, the timing constraints imposed on the targets occupying the second to last rooms for the ith UAV may be now represented as the following:

$$\sum_{k=1}^{p} t_k^{\alpha} a_{ij}^k \le \sum_{k=1}^{p} \underline{C_0}(i, k) a_{i1}^k + \sum_{u=1}^{j-1} \sum_{v=1}^{p-1} \sum_{w=v+1}^{p} \underline{c}(\eta(v, w))\, d_{i\eta(v,w)}^u \le \sum_{k=1}^{p} t_k^{\beta} a_{ij}^k \quad (22)$$

for each $j \in \{1, 2, \ldots, q'\}$. The left inequality is fine, but the right inequality causes a problem when some of the rooms for a UAV are empty, i.e. less than q' prime targets are assigned to a UAV, thereby forcing the a variables corresponding to unoccupied rooms to be zero. For this reason, we use the following method:

$$\sum_{k=1}^{p} \underline{C_0}(i, k) a_{i1}^k + \sum_{u=1}^{j-1} \sum_{v=1}^{p-1} \sum_{w=v+1}^{p} \underline{c}(\eta(v, w))\, d_{i\eta(v,w)}^u \le \sum_{k=1}^{p} t_k^{\beta} a_{ij}^k + M(1 - \sum_{k=1}^{p} a_{ij}^k) \quad (23)$$

where $M > 0$ is a large constant number. This makes the right inequality vacuous whenever a room is unoccupied, i.e. $\sum_{k=1}^{p} a_{ij}^k = 0$ for some i, j.

The final task for grouping targets is to make the relationship between variables a and x. The main difficulty in doing this is that x is defined for all n targets, but a for only p prime targets. In order to resolve this problem, we recall the integer variable z, as defined at the beginning of this section, and consider the following linear constraints:

$$\sum_{k=1}^{p} (ID)_k a_{ij}^k = \sum_{k=1}^{n} k z_{ij}^k \quad \forall\, i, j \quad (24)$$

$$\sum_{k=1}^{n} z_{ij}^k \le 1 \quad \forall\, i, j \quad (25)$$

$$\sum_{j=1}^{q'} z_{ij}^u \le x_{iu} \quad \forall\, i, u \quad (26)$$

where i, j and u run from 1 to m, 1 to q' and 1 to n, respectively, and the constant one-dimensional array $(ID)_k$ ($k \in \{1, 2, \ldots, p\}$) contains a unique identifier (number from 1 to n) for each prime target. The equality (24) and inequality (25) basically perform the function: if the ith UAV's jth room is occupied by the target with ID $(ID)_k$, then z_{ij}^k is enabled and subsequently x_{ik} is set to 1 by (26).

By putting all the aforementioned constraints together, $\widetilde{\mathbf{H}}_2$ can be described as follows:

Step 1. *Sub-optimal partitioning*: Consider an optimization problem $\widetilde{\mathcal{F}}_2$, as shown below, that solves for x_{ij} ($x_{ij} = 1$ if the jth target is assigned to the ith UAV) to partition the underlying set of targets into m subsets $\mathcal{T}_i$ ($i = 1, 2, \ldots, m$) such that (i) each $\mathcal{T}_i$ contains at most q elements; (ii) the travelling cost T_j-to-T_k for each j, k is minimized, where T_j or T_k is the starting position of the ith UAV or belongs to $\mathcal{T}_i$; (iii) each target is covered by exactly one UAV.

Step 2. *Optimal path-planning*: For each $\mathcal{T}_i$, consider $\widetilde{\mathcal{F}}_E$ (see the remark below) in order to obtain the optimal path of visiting all the targets contained in $\mathcal{T}_i$ by the ith UAV.

$\widetilde{\mathcal{F}}_2$: minimize r

subject to (10)–(15), (18)–(21), the left inequality of (22) and (23)–(26).

We do not further elaborate on the MILPs associated with the first step of $\widetilde{\mathbf{H}}_3$, in which the UAV's return constraints are considered, and the second steps of both $\widetilde{\mathbf{H}}_2$ and $\widetilde{\mathbf{H}}_3$ and their associated programs $\widetilde{\mathcal{F}}_E$ and $\widetilde{\mathcal{F}}_{Eret}$, because all these MILPs can be easily constructed using the aforementioned techniques. Instead, we remark that the numbers of binary variables newly introduced to accommodate timing constraints for the first step and the second step of both algorithms are $mq'(p(p+1)/2 + n) - mp(p-1)/2$ and 0, respectively, and the numbers of newly added constraints are approximately $mq'(p(p-1)+6)+mn+p$ and $2mp$, respectively. In the next section, we investigate via extensive numerical simulations how much these added variables and constraints affect the performance of the algorithms.

3 Numerical Simulations

We first present an introductory example showing how timing constraints change the solution to a basic target assignment problem. Figure 3 shows an example scenario and solution paths chosen when no timing constraints are considered. The scenario consists of 3 UAVs, 4 non-prime targets, 5 prime targets, and both the maximum number of targets and the maximum number of prime targets visited by a single UAV are 3, i.e. $m = 3$, $n = 9$, $p = 5$, $q = 3$ and $q' = 3$. The scenario also contains five threats which are each marked as "x". The threats create a joint probabilistic risk distribution (contour lines in the figure) and the risk is determined using the deterministic formulae found in [10]. In the figure, ten contour lines representing the risks ranged from 0.1 to 1 are plotted for each threat. The closer a line is to a threat source the higher the risk. The targets

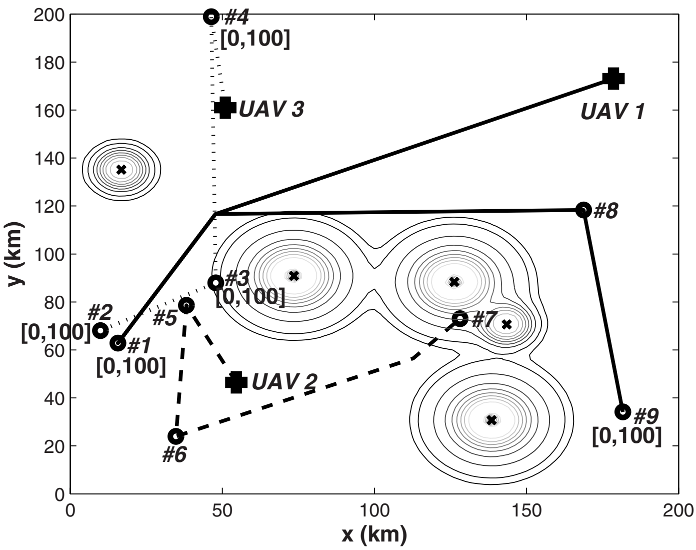


Fig. 3. Example solution paths (solid line for UAV 1, dashed line for UAV 2 and dotted line for UAV 3) when no timing constraints are considered

labeled with #1, #2, #3, #4 and #9 are supposed to be prime targets which are required to be visited by a UAV before 100 (min) after the mission starts, but their associated timing constraints are neglected at this time.

Following the aforementioned procedure of creating a network of UAVs' flying routes, we identify *nodes* including the UAV starting positions, the target position and the corner points of the smallest rectangles which cover the risky area due to the threats. We then associate each segment (joining two distinct nodes) with a cost weighting of 90% to the total risk (scaled to 1) along the segment and 10% to the length (scaled to 1) of the segment, and subsequently compute necessary travelling costs C_0, C, $\underline{C_0}$ and $\underline{C}$. Under the assumption that UAVs fly at a constant speed of 2 (km/min) and an altitude of 2 (km) in the operational range of $[0, 200] \times [0, 200]$ (km), algorithm $\mathbf{H}_2$ returns the solution paths shown in Fig. 3. The first UAV first visits #1, then #8 and finally #9, the second UAV visits #5, #6 and #7, and the third UAV #4, #3 and #2 in turn. As easily noticed, two prime targets (#2 and #9) are visited after, not before, 100 (min). However, the modified algorithm $\widetilde{\mathbf{H}}_2$ accounting for the timing requirements returns completely different solution paths, as shown in Fig. 4. This time, the first UAV first visits #8, then #9 and finally #5, the second UAV visits #3, #1 and #7, and the third UAV visits #4, #2 and #6 in turn. Note that every prime target is visited within the required time window.

Next we proceed with investigating the effect of timing constraints on the total computational time needed for executing $\widetilde{\mathbf{H}}_2$ and $\widetilde{\mathbf{H}}_3$. We first recall from the results in [10] that $\mathbf{H}_2$ and $\mathbf{H}_3$ show the performance of yielding solutions within

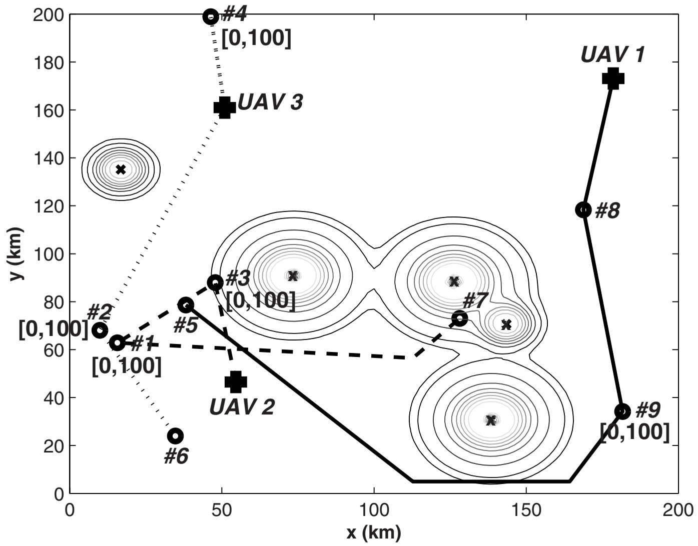
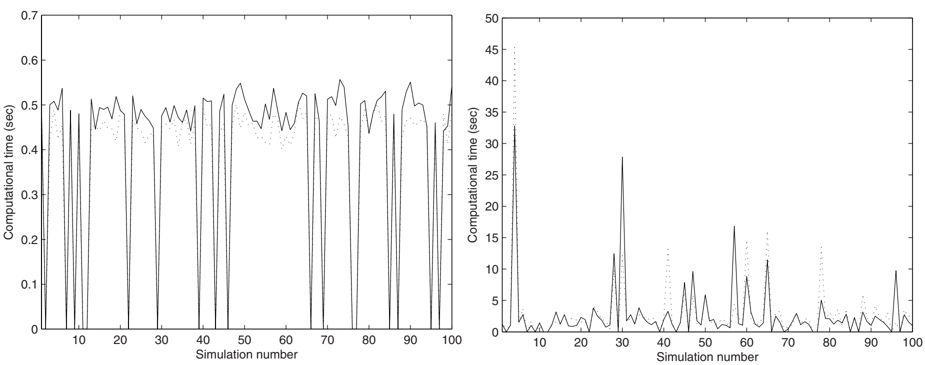


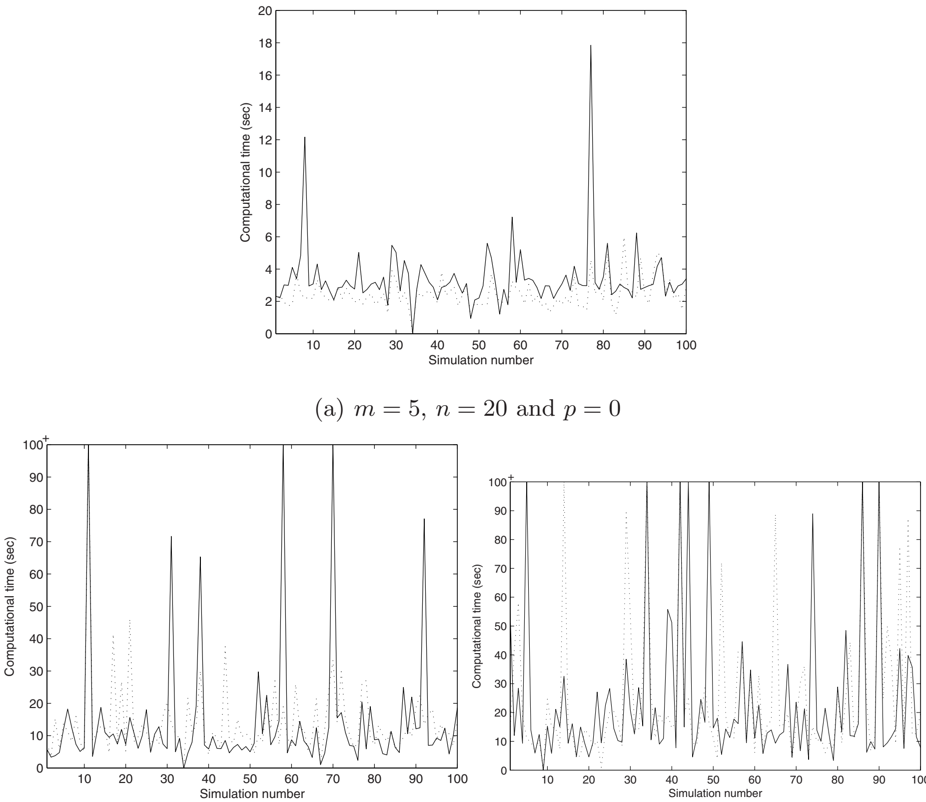
Fig. 4. Example solution paths (solid line for UAV 1, dashed line for UAV 2 and dotted line for UAV 3) when timing constraints are considered: prime targets (labeled with #1, #2, #3, #4 and #9) are required to be visited by a UAV before 100 (*min*) after the mission starts



(a) $m = 3$, $n = 10$, $p = 0$ and $q = 4$ (b) $m = 3$, $n = 10$, $p = 6$, $q = 4$ and $q' = 2$

Fig. 5. The total computation times needed for executing $\widetilde{\mathbf{H}}_2$ (solid line) and $\widetilde{\mathbf{H}}_3$ (dotted line) when $m = 3$ and $n = 10$

6 seconds on average as well as guaranteeing $\mathbf{T}/\mathbf{T}^* < 1.5$ for up to $m = 5$, $q = 4$ and $n = 20$. As timing constraints may greatly decelerate the speed of finding $\mathbf{T}^*$, i.e. solving the problem exactly, we here especially focus on investigating the computational performance of $\widetilde{\mathbf{H}}_2$ and $\widetilde{\mathbf{H}}_3$ versus various m, n and p. To this end, for fixed m, n, p, q and q' we create one hundred random scenarios with various UAV starting positions ($\in \mathbf{X}_2$ (km)), where $\mathbf{X}_h = [0, 200] \times [0, 200] \times h$ (km),



(a) $m = 5$, $n = 20$ and $p = 0$

(b) $m = 5$, $n = 20$ and $p = 5$

(c) $m = 5$, $n = 20$ and $p = 6$

Fig. 6. The total computation times needed for executing $\widetilde{\mathbf{H}}_2$ (solid line) and $\widetilde{\mathbf{H}}_3$ (dotted line) when $m = 5$ and $n = 20$; 100^+ denotes the number greater or equal to 100

the target positions ($\in \mathbf{X}_2$), the number of threats ($\in \{5, 6, \ldots, 10\}$), the threat locations ($\in \mathbf{X}_0$) and the threat ranges (7 or 25 (km)). We again assume that UAVs fly at a constant speed of 2 (km/min), and every prime target must be visited within 100 (min) from the mission starting. All numerical tests are done with a personal computer equipped with an Intel(R) Pentium 4 CPU 3.40GHz.

Figure 5(a) shows the results when the scenarios consist of 3 UAVs, 10 non-prime targets, no prime target and the maximum number of targets visited by a single UAV set to 4. The dotted (respectively, solid) line shows the total computational times when the UAV's return constraint is (respectively, not) considered. Note that all the computations are done in 0.6 (sec). However, when the number of prime targets gets increased to 6 and the maximum number of prime targets visited by a single UAV is set to 2, one can see several peaks, as shown in Fig. 5(b). Although many cases (93%) are handled within less than 10 (sec), the figure shows that timing constraints can greatly complicate the *basic* assignment problem. Similar observations follow from Fig. 6. As expected, for the cases in which 5 UAVs cover 20 targets with no timing constraints and each UAV is restricted to visit at most 4 targets, the proposed algorithms manage to deal with all the cases within 6 (*sec*) on average, as depicted in Fig. 6(a). However, when 5 of the 20 targets become prime targets and each UAV is further restricted to visit at most two prime targets, Fig. 6(b) suggests that the computational burden dramatically increases for several cases. The situation becomes worse when the number of prime targets gets increased to 6, as seen in Fig. 6(c). However, in spite of the presence of such *unpleasant* scenarios, we note that for more than 90% of the tested cases, $\widetilde{\mathbf{H}}_2$ and $\widetilde{\mathbf{H}}_3$ return solutions within 20 (*sec*) when the sum of the numbers of UAVs and targets (including prime targets) is less than or equal to 25, the maximum number of targets visited by a single UAV is less than or equal to 4 and the number of prime targets is less than or equal to 5 or 6 depending on the numbers of UAVs and targets.

4 Concluding Remarks

We have considered the UAV-to-target assignment problem especially focused on the presence of time-critical (prime) targets. Our main challenges (contributions) in doing this are to keep the previously guaranteed theoretical bounds (16) and (17) and to allow only a slight increase of computational cost after including time-critical targets. We analytically show that by means of adding several integer linear constraints to the previous MILP formulation, the bounds still hold under a mild condition ($q' < 3$). In the numerical experiments, for more than 90% of the tested cases the newly proposed algorithms returned solutions within 20 (*sec*) when the sum of the numbers of UAVs and targets (including prime targets) is less than or equal to 25, the maximum number of targets visited by a single UAV is less than or equal to 4 and the number of prime targets is less than or equal to 5 or 6 depending on the numbers of UAVs and targets.

References

1. Ahuja, R., Kumar, A., Jha, K., Orlin, J.: Exact and heuristic methods for the weapon target assignment problem. Oper. Res. (to appear)
2. Alighanbari, M., How, J.: Decentralized task assignment for unmanned aerial vehicles. Proc. 44th IEEE Conf. Decision Control (2005) 5668–5673
3. Alighanbari, M., Kuwata, Y., How, J.: Coordination and control of multiple UAVs with time constraints and loitering. Proc. IEEE Amererican Control Conf. (2003) 5311–5316
4. Beard, R., McLain, T., Goodrich, M., Anderson, E.: Coordinated target assignment and intercept for unmanned aer vehicles. IEEE Trans. Robot. Automat. **18** (2002) 911–922
5. Bellingham, J., Tillerson, M., Richards, A., How, J.: Multi-Task Allocation and Trajectory Design for Cooperating UAVs. Proc. 2nd Annual Conf. Cooperative Control Optimization. (2001) 1–19

6. Christofides, N., Mignozzi, A., Toth, P.: The vehicle routing problem. Combinatorial Optimization. Wiley and Sons (1979) 315–338
7. Gu, D.-W., Kamal, W. A., Postlethwaite, I.: A UAV Waypoint Generator. Proc. AIAA 1st Intelligent Systems Technical Conf. **AIAA-2004-6227** (2004)
8. Gu, D.-W., Postlethwaite, I., Kim, Y.: A Comprehensive Study on Flight Path Selection Algorithms. Proc. IEE Seminar Target Tracking (2006) 77–89
9. Haimovich, M., Kan, R., Stougie, L.: "Analysis of heuristics for vehicle routing problems" in Vehicle Routing: Methods and Studies. Elsevier (1988) 87-127
10. Kim, Y., Gu, D.-W., Postlethwaite, I.: Real-time optimal mission scheduling and flight path selection. IEEE Trans. Automat. Contr. (to appear)
11. Kingston, D. B., Schumacher, C. J.: Time-dependent cooperative assignment. Proc. IEEE American Control Conf. (2005) 4084–4089
12. Korte, B., Vygen, J.: Combinatorial Optimization: Theory and Algorithms. Springer (2002)
13. Li, C., Simchi-levi, D.: Worst-case analysis of heuristics for multidepot capacitated vehicle routing problem. ORSA J. Comput. **2** (1990) 64–73
14. Manne, A.: A target-assignment problem. Oper. Res. **6** (1958) 346–351
15. MacLain, T. W., Beard, R. W.: Coopertative path planning for timing-critical missions. Proc. IEEE American Control Conf. (2003) 296–301
16. Mitrovic-minic, S., Krishnamurti, R.: The multiple travelling salesman problem with time windows: bounds for the minimum number of vehicles. Simon Fraser University **TR-2002-11** (2002)
17. Marinakis, Y., Migdalas, A.: Heuristic solutions of vehicle routing problems in supply chain management. Working Paper (2001)
18. Ryan, J., Bailey, G., Moore, J., Carlton, W.: Reactive tabu search in unmanned aerial reconnaissance simulations. Proc. Winter Simulation Conf. (1998) 873–879
19. Savelsbergh, M.: Local search in routing problems with time windows. Ann. Oper. Res. **4** (1985) 285–305
20. Schaerf, A.: A survey of automated timetabling. Artif. Intell. Rev. **13** (1999) 87–127
21. Shetty, V., Sudit, M., Nagi, R.: Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. Comp. & Oper. Res. **doi: 10.1016/j.cor.2006.09.013** (2006)
22. White, C., Lee, Y., Kim, Y., Thomas, R., Perkins, P.: Creating weekly timetables to maximize employee preferences. The UMAP J. **25** (2004) 5–25