

Real-Time Optimal Mission Scheduling and Flight Path Selection

Yoonsoo Kim, Da-Wei Gu, and Ian Postlethwaite

Abstract—We consider a mission in which m UAVs with a capacity limit q each visit n ($\leq mq$) targets in a hostile environment in a cooperative manner (and return to where they departed from) such that the cost reflecting operating time and risk exposed is minimized. We first propose a mixed-integer linear programming (MILP) formulation which exactly solves the problem and then propose four alternative MILP formulations which are computationally less intensive and, therefore, suited for real-time purposes, but yield a theoretically guaranteed suboptimal solution. The main contributions of this note are an exact but compact MILP formulation, using a *room* concept, and four nonexact formulations, each one of which ensures a certain level of solution quality and relatively fast computational time for the considered real-time mission scheduling problem.

Index Terms—Mixed-integer linear programming (MILP), real-time target assignment, unmanned air vehicle (UAV).

I. INTRODUCTION AND PROBLEM STATEMENT

For m identical unmanned air vehicles (UAVs) U_i ($i = 1, 2, \dots, m, m \geq 2$) at corresponding starting positions T_0^i , a battle field flight mission is considered. The terrain is assumed to cover an area \mathbf{X} and is defended by enemy air defences, radar/SAM (surface-to-air-missile) units (so-called threats). There are several threats, some of which are “pop-ups,” and, thus, their locations are not known *a priori*. The UAVs are required to visit n spots T_j ($j = 1, 2, \dots, n, n \geq m$) within the terrain \mathbf{X}^1 and possibly to return to where they departed from. In this note, 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 exposed) incurred by any single UAV’s travel is minimized while keeping the number of targets visited by a single UAV below a certain limit q . More precisely, we would like to calculate \mathbf{T}^*

$$\mathbf{T}^* \stackrel{\text{def}}{=} \min_k \max_{\mathcal{A}(k)_i} \mathbf{T}(k)_i^* \quad (1.1)$$

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, $\mathcal{A}(k) \in \mathcal{A}$ is one of the feasible assignments, $\mathcal{A}(k)_i$ is the subassignment given to the i th UAV within $\mathcal{A}(k)$, and, finally, $\mathbf{T}(k)_i^*$ is the *optimal* cost of completing the subassignment $\mathcal{A}(k)_i$ by the i th UAV. We note that the objective functional to be minimized is appropriate for balancing workloads across UAVs. We first propose a mixed-integer linear program (MILP) formulation which exactly solves the problem (1.1), and then we propose four alternative MILP formulations which are computationally less intensive but yield a theoretically guaranteed suboptimal solution.

Manuscript received February 13, 2006; revised August 1, 2006 and January 24, 2007. Recommended by Associate Editor L. Lygeros. This work was supported by the U.K. Engineering and Physical Sciences Research Council and BAE Systems.

The authors are with the Department of Engineering, University of Leicester, Leicester LE1 7RH, U.K. (e-mail: yk17@le.ac.uk; dag@le.ac.uk; ixp@le.ac.uk).

Digital Object Identifier 10.1109/TAC.2007.899048

¹We assume that the n targets are sufficiently spread out so that UAVs can easily follow the reference (straight-line) paths which will be constructed in Section II.

Many research directions have been explored regarding the problem in question. These include Weapon-target assignment [1], [10], timetabling [14], the celebrated Traveling Salesman Problem [13], and, more generally, capacity-limited vehicle routing problems [7], [11]. We note that these problems are slightly different from the problem in the present context, in that 1) we may not require UAVs to return to their starting positions; 2) we minimize the individual *tour*² cost for balanced workload, not the total cost incurred by the whole mission; 3) UAVs do not necessarily depart from the same depot. There is also much literature available on coordinated target assignment of UAVs, for example, [2]–[5], and some of which add the timing and precedence constraints to the original problem [3]. However, here we focus more on the computational aspect of the *basic* task assignment problem which is also of important practical significance to planning a mission in a hostile as well as uncertain environment. As the underlying problem is known to have a membership in the class of NP-hard problems, it is often unhelpful 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 its computational demand [3]. As an alternative approach, tabu search-based methods are also useful for this purpose [11], [12]. When time is critical, heuristic or nonexact methods have been considered, even if global optimality may not be achieved [4], [7]. Among many heuristics, we note the *iterated optimal tour partitioning* (IOTP) algorithm proposed in [7], [9], mainly for multivehicle-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 nonexact method in such a way that the advantages of each are enjoyed. In this note, we have partially achieved this, and subsequently we can handle the target assignment problem effectively in terms of optimality and computational complexity. We propose four MILP formulations each of which ensures a certain level of solution quality and relatively fast computational time for the considered problem. More usefully, suboptimality bounds are derived for each of them. The note is organized as follows. In Section II, we create a feasible network of routes over which a UAV can operate based on a variation of the visibility line method introduced in [3] and [6] for UAV applications. In Section III-A, we then proceed to present an exact but compact MILP formulation solving the problem with and without the UAVs’ return constraint using a *room* concept. In Section III-B, we then propose four nonexact methods, which are computationally fast, still using MILP to obtain a suboptimal assignment along with a worst-case analysis. Finally, numerical examples are given in Section IV to support the use of the proposed formulation for real-time purposes.

II. CONSTRUCTION OF FEASIBLE FLIGHT PATHS

Based on the exact or estimated knowledge of the location of possible threats, we first calculate the probabilistic risk distribution $P(w)$ ($w = (x_0, y_0, z_0) \in \mathbf{X}$) over \mathbf{X} . In this note, we regard the principal source of danger as M enemy missile units with short R_s (7 Km), medium R_m (25 Km), or long R_l (65 Km) ranges, and subsequently calculate $P(w)$ from the following formula, as given in [6], $P(w) = 1 - \prod_{i=1}^M (1 - P_i(w))$, where $P_i(w) = (1 - \text{Step}(d, R_{\{s,m,l\}}, k_1)) \times \text{Step}(d, 0.1 \times R_{\{s,m,l\}}, k_2) \times \text{Step}(\sin^{-1}(z_0/d), \gamma, k_3)$, where $\text{Step}(a, b, c) = (1 + (a - b)/\sqrt{c^2 + (a - b)^2})/2$, d is the distance from w to the i th missile, γ (assumed to be 0.17rad) is the lower

²In this note, by *tour*, we denote a sequence of moves made by U_i starting at T_0^i (and ending at T_0^i , when required).

coverage angle of the radar of each missile system, and k_1, k_2, k_3 are the softness parameters of the *step* function with the values of 5, 1, 0.1 in this note, respectively. One can then construct a network \mathcal{G} of relatively *safe* routes that allow UAVs to visit all the targets while guaranteeing a certain degree of safety. Such a network \mathcal{G} can be designed and updated in real-time through various path planning methods using, for example, Voronoi diagrams [4] or visibility lines [3], [6].

More precisely, we first consider a cover $\mathcal{C}(\mathcal{P}_\rho)$ of the set $\mathcal{P}_\rho = \{w \mid P(w) > \rho\}$ of unsafe points for a constant value $\rho \geq 0$. As the complexity of describing such a cover often leads to a heavy computational workload, we approximate $\mathcal{C}(\mathcal{P}_\rho)$ by the union of smallest cubes or rectangles covering \mathcal{P}_ρ for real-time use. We assume that the location l_i and the strength (range) s_i of all threats t_i can be estimated or informed in real-time, and $\mathcal{C}(\mathcal{P}_\rho)$ is the union of the smallest cubes covering $\cup_i \{w \mid \|w - l_i\| \leq s_i\}$. Then, given the cover $\mathcal{C}(\mathcal{P}_\rho)$, we construct a network $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where \mathcal{V} and \mathcal{E} are the sets of vertices and edges on \mathcal{G} , respectively. The elements of \mathcal{V} include the initial locations of the UAVs $T_0^i (i = 1, 2, \dots, m)$, the locations of the targets $T_j (j = 1, 2, \dots, n)$, and the vertices of $\mathcal{C}(\mathcal{P}_\rho)$ (the corner points of each cube). Each edge $e \in \mathcal{E}$ is a straight path connecting two distinct vertices $v_i, v_j \in \mathcal{V}$ except the pairs $(T_0^i, T_0^j) (i, j = 1, 2, \dots, m)$, along which the UAV's traveling cost $\text{co}(v_i, v_j)$ (reflecting the length as well as the total risk of the path) is less than some fixed constant α . As a result, it is unlikely that paths strictly passing through $\mathcal{C}(\mathcal{P}_\rho)$ belong to \mathcal{E} . We calculate the cost $\text{co}(v_i, v_j)$ as follows:

$$\text{co}(v_i, v_j) = \kappa \|v_i - v_j\| + (1 - \kappa) \times \left(\sum_{k=1}^{\beta(v_i, v_j) - 1} \frac{P(y_{k+1}) + P(y_k)}{2} \right) \|v_i - v_j\| \quad (2.2)$$

where $\kappa \in [0, 1]$ is a parameter that allows us to make a compromise between the length and the risk of the path, e.g., $\kappa = 0.2$ in our examples, $P(y_k)$ is the risk at equally spaced intermediate points $y_k (k = 1, 2, \dots, \beta(v_i, v_j))$, $y_1 = v_i, y_{\beta(v_i, v_j)} = v_j$ on the path between v_i and v_j , and integer $\beta(v_i, v_j)$ is proportional to $\|v_i - v_j\|$, e.g., $\beta(v_i, v_j) = \max\{3, \lceil \lceil \|v_i - v_j\| / (km) \rceil \rceil\}$. We then finally construct a matrix $C_0 = [C_0(i, j)] \in \mathbf{R}^{m \times n}$ and a vector $c = [c(\eta(v, w))] \in \mathbf{R}^{(n-1)/2}$ such that $C_0(i, j)$ (respectively, $c(\eta(v, w))$) is the cheapest cost of traveling from T_0^i to T_j (respectively, T_v to T_w) over the network $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}, v, w \in \{1, 2, \dots, n\} (v < w), \eta(v, w) = (v-1)n - v(v-1)/2 + w - v$ and $c(\eta(v, w)) = c(\eta(w, v))$. Dijkstra's algorithm in [8] could be used for generation of C_0 and c .

III. OPTIMAL TARGET ASSIGNMENT

In this section, we will suggest several MILP formulations. The first MILP (denoted by \mathbf{E} when ignoring UAV return or by \mathbf{E}_{ret} when considering UAV return) solves the problem (1.1) exactly. Unfortunately, as one will see in Section IV, this approach is computationally very expensive. This motivates us to propose nonexact algorithms (for short, $\mathbf{H}_1, \dots, \mathbf{H}_4$) which first solve

$$r_{\min} \stackrel{\text{def}}{=} \min_k \max_{\mathcal{A}(k)_i} \tilde{\mathbf{T}}(k)_i^* \quad (3.3)$$

where a substitute $\tilde{\mathbf{T}}(k)_i^*$ less than $\mathbf{T}(k)_i^*$ is used so that $r_{\min} \leq \mathbf{T}^*$. The nonexact algorithms involve lower computational complexity. Methods \mathbf{H}_1 and \mathbf{H}_2 (\mathbf{H}_3 and \mathbf{H}_4) are meant for the problem that does not (does) demand all the UAVs to return to where they departed from. Each nonexact algorithm employs different $\tilde{\mathbf{T}}(k)_i^*$ in (3.3). The result obtained by solving (3.3) can be further improved through a so-called Phase II procedure, which will be introduced in Section III-B.

A. Exact MILP Formulation

1) *With No UAV Return:* As each UAV visits at most q distinct targets, we consider the problem as if each UAV has q rooms (numbered from 1 to q) to hold the identifiers (IDs) of targets. This concept leads to defining integer variables x_{ij} 's ($\in \{0, 1, 2, \dots, n\}$) and binary variables a_{ij}^k 's ($\in \{0, 1\}$) as $x_{ij} = \sum_{k=1}^n k a_{ij}^k$, where $i \in \{1, 2, \dots, m\}$ and $j \in \{1, 2, \dots, q\}$. The variable a_{ij}^k represents the relationship between the j th room of the i th UAV and the k th target. With these variables, we need

$$\sum_{k=1}^n a_{ij}^k \leq 1 \quad \forall i, j \quad \text{and} \quad \sum_{i=1}^m \sum_{j=1}^q a_{ij}^k = 1 \quad \forall k \quad (3.4)$$

where $i = \{1, \dots, m\}, j = \{1, \dots, q\}$ and $k = \{1, \dots, n\}$, in order to guarantee that each room of a UAV holds at most one target's ID and each target is covered by exactly one UAV.

Second, we force the rooms of a UAV to be filled in ascending order of room number, i.e.,

$$\sum_{k=1}^n a_{i(j+1)}^k \leq \sum_{k=1}^n a_{ij}^k \quad \forall i, j \quad (3.5)$$

where $i = \{1, \dots, m\}$ and $j = \{1, \dots, q-1\}$, so that the resultant path for the i th UAV becomes $T_0^i \rightarrow T_{x_{i1}} \rightarrow T_{x_{i2}} \rightarrow \dots \rightarrow T_{x_{iq'}}$, where $q' (\leq q)$ is the largest nonzero (nonempty) room number of the i th UAV. Next, we associate an edge, i.e., a path between T_k 's or between T_0^i and T_k , with its corresponding cost. Recalling the cost matrix C_0 and the vector c defined in Section II, the cost corresponding to the edge between T_0^i and $T_{x_{i1}}$ is $\sum_{k=1}^n C_0(i, k) a_{i1}^k$.

On the other hand, the MILP expression of the traveling cost between targets requires the introduction of additional auxiliary variables $y_{i\eta(v, w)}$ (as well as $\tilde{y}_{i\eta(v, w)}^j$ and $\tilde{y}_{i\eta(v, w)}^j$) which are defined through the following inequalities:

$$a_{ij}^v + a_{i(j+1)}^w + a_{ij}^w + a_{i(j+1)}^v = 2y_{i\eta(v, w)}^j + \tilde{y}_{i\eta(v, w)}^j \quad \forall i, j, v, w \quad (3.6)$$

$$y_{i\eta(v, w)} = \sum_j y_{i\eta(v, w)}^j \quad \forall i, v, w \quad (3.7)$$

$$y_{i\eta(v, w)}^j \in \{0, 1\}, \quad \tilde{y}_{i\eta(v, w)}^j \in [0, 1] \quad \forall i, j, v, w \quad (3.8)$$

where $i \in \{1, \dots, m\}, j \in \{1, \dots, q-1\}, v, w \in \{1, 2, \dots, n\} (v < w)$ and $\eta(v, w) = (v-1)n - v(v-1)/2 + w - v$. The corresponding traveling cost from $T_{x_{i1}}$ to $T_{x_{iq'}}$ (via $T_{x_{i2}}, \dots, T_{x_{i(q'-1)}}$) becomes $\sum_{v=1}^{n-1} \sum_{w=v+1}^n c(\eta(v, w)) y_{i\eta(v, w)}$. The inequality (3.6) enables $y_{i\eta(v, w)}^j$ and subsequently $y_{i\eta(v, w)}$ in (3.7) 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, $y_{i\eta(v, w)}$ is set to 1 only if the v th and w th targets are assigned to the consecutive j th and $(j+1)$ th (or $(j+1)$ th and j th) rooms of the i th UAV. Finally, putting together all the necessary constraints, we have the program \mathcal{F}_E : minimize r subject to (3.4)–(3.8) and

$$\sum_{k=1}^n C_0(i, k) a_{i1}^k + \sum_{v=1}^{n-1} \sum_{w=v+1}^n c(\eta(v, w)) y_{i\eta(v, w)} \leq r \quad \forall i. \quad (3.9)$$

$C_0(i, k)$ [respectively, $c(\eta(v, w))$] is the traveling cost from T_0^i to T_k (respectively, from T_v to T_w). We note that the total number of binary variables used in \mathcal{F}_E and the total number of constraints are both of the order of mn^2q .

2) *With UAV Return:* The UAVs' return constraint requires us to add an additional term that accounts for the return cost to the left side of (3.9). The constraint (3.5) facilitates the identification of the first

target covered by each UAV and to calculate the corresponding traveling cost, i.e., $\sum_{k=1}^n C_0(i, k) a_{i1}^k$. But simultaneously it complicates the traveling cost from the final target covered by a UAV to its departure point in terms of existing variables in \mathcal{F}_E . For this reason, we introduce additional variables b_{ij}^k 's satisfying the following inequalities:

$$a_{i1}^k = b_{i1}^k \quad \forall i, k \quad (3.10)$$

$$b_{i(j-1)}^k - \sum_{k=1}^n a_{ij}^k \leq b_{ij}^k \leq b_{i(j-1)}^k + \sum_{k=1}^n a_{ij}^k \quad \forall i, j, k \quad (3.11)$$

$$a_{ij}^k \leq b_{ij}^k \leq a_{ij}^k + \left(1 - \sum_{k=1}^n a_{ij}^k\right) \quad \forall i, j, k \quad (3.12)$$

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

The variable b_{ij}^k is a copy of a_{ij}^k , but $b_{ij}^k = 1$ for all $j \in \{q' + 1, \dots, q\}$ if the i th UAV visits T_k last and covers only $q' (< q)$ targets. Then, the original problem with the UAVs' return constraint can be handled by solving \mathcal{F}_E with the inclusion of (3.10)–(3.12) but with the replacement of (3.9) by $\sum_{k=1}^n C_0(i, k)(a_{i1}^k + b_{iq}^k) + \sum_{v=1}^{n-1} \sum_{w=v+1}^n c(\eta(v, w)) y_{i\eta(v, w)} \leq r$ for every i . By noting that $\sum_{k=1}^n a_{ij}^k \in \{0, 1\}$, one can readily prove that (3.11)–(3.12) guarantees $b_{ij}^k = b_{i(j-1)}^k$, if $\sum_{k=1}^n a_{ij}^k = 0$; otherwise, $b_{ij}^k = a_{ij}^k$. For instance, suppose that the i th UAV covers T_1, T_3 and T_5 in turn and $q = 5$. This implies that $a_{i1}^1 = a_{i2}^3 = a_{i3}^5 = 1$ but $a_{ij}^k = 0$ for all other j, k 's. On the other hand, by (3.10)–(3.12), we have $b_{i1}^1 = b_{i2}^3 = b_{i3}^5 = b_{i4}^5 = b_{i5}^5 = 1$ but $b_{ij}^k = 0$ for all other j, k 's. We note that the order of binary variables and constraints used for this UAV's return case are the same as the no-return case.

B. Nonexact Algorithms

As indicated earlier, the previous MILP formulation requires the order of mn^2q binary variables and the constraints are highly restrictive. This motivates us to seek a formulation which is numerically tractable but not perfectly optimal. In this section, we will still use MILP, but with fewer restrictions. The solution obtained via the MILP is to be utilized to construct a suboptimal assignment and to yield a corresponding worst-case bound that enables us to evaluate the quality of the proposed algorithm.

1) *With No UAV Return*: In this section, we propose two MILP formulations. One (\mathbf{H}_1) is computationally faster but less optimal than the other (\mathbf{H}_2). Our first strategy \mathbf{H}_1 is stated as follows. **Phase I. Suboptimal partitioning**: Consider an optimization problem \mathcal{F}_1 (see below) that partitions the underlying set of targets into m subsets \mathcal{T}_i ($i = 1, 2, \dots, m$) such that 1) each \mathcal{T}_i contains at most q elements; 2) the traveling cost from T_i^0 to the *farthest*, i.e., most expensive, target, with respect to T_i^0 , in \mathcal{T}_i is minimized; 3) each target is covered by exactly one UAV. **Phase II. Optimal path-planning**: For each \mathcal{T}_i , solve \mathcal{F}_E to obtain the optimal path of visiting all the targets contained in \mathcal{T}_i by the i th UAV.

The following program \mathcal{F}_1 creates a partition as described in the first phase of \mathbf{H}_1 : minimize r subject to

$$\sum_{j=1}^n x_{ij} \leq q \quad \forall i, \quad \sum_{i=1}^m x_{ij} = 1 \quad \forall j, \quad x_{ij} \in \{0, 1\} \quad \forall i, j \quad (3.13)$$

$$\max_j C_0(i, j) x_{ij} \leq r \quad \forall i \quad (3.14)$$

where (3.14) can be rewritten as $C_0(i, j) x_{ij} \leq r$, where UAV index $i \in \{1, \dots, m\}$ and target index $j \in \{1, \dots, n\}$. The program \mathcal{F}_1 that solves for x_{ij} is another MILP but involves much fewer binary variables than \mathcal{F}_E . The variable x_{ij} tells us whether the i th UAV covers

the j th target, i.e., the j th target belongs to \mathcal{T}_i covered by the i th UAV, where $\mathcal{T}_i \subseteq \mathcal{T} = \{T_1, T_2, \dots, T_n\}$. We note that \mathcal{F}_1 does not consider the traveling cost between targets. Instead, it just tries to minimize the worst cost [i.e., $\tilde{\mathbf{T}}_i^*$ in (3.3)] of traveling between a UAV, say initially at T_i^0 , and the farthest target, with respect to T_i^0 , to be reached by the UAV while satisfying the capacity constraint.

As mentioned earlier, one favourable fact of the optimization problem \mathcal{F}_1 is that the resultant objective function value r of \mathcal{F}_1 can serve as a lower bound on the actual optimal solution of the original problem. In other words, if we let r_{\min} be the solution to \mathcal{F}_1 , then $r_{\min} \leq \mathbf{T}^*$, where \mathbf{T}^* is defined in (1.1). In addition, we have $\mathbf{T}_i^* \leq (2q - 1) r_{\min} \leq (2q - 1) \mathbf{T}^*$, where \mathbf{T}_i^* is the i th UAV's optimal cost of visiting all the targets in \mathcal{T}_i obtained by solving \mathcal{F}_1 . The first inequality is due to the fact that for every i , \mathcal{T}_i has at most q elements in it. Thus, we obtain $\mathbf{T}/\mathbf{T}^* \leq 2q - 1$, where $\mathbf{T} = \max_i \mathbf{T}_i^*$. Last, we note that the number of binary variables and, respectively, constraints for **Phase I** are both of the order of mn .

Our next strategy \mathbf{H}_2 employs the same strategy as \mathbf{H}_1 but solves the following \mathcal{F}_2 in place of \mathcal{F}_1 in **Phase I**: minimize r subject to (3.13), (3.14)

$$y_{i\eta(j, k)} \leq \frac{x_{ij} + x_{ik}}{2} \leq y_{i\eta(j, k)} + \frac{1}{2}, \quad y_{i\eta(j, k)} \in \{0, 1\} \quad \forall i, j, k (j < k) \quad (3.15)$$

$$c(\eta(j, k)) y_{i\eta(j, k)} \leq r \quad \forall i, j, k (j < k) \quad (3.16)$$

where $i \in \{1, \dots, m\}$, $j, k \in \{1, \dots, n\}$ and $\eta(j, k) = (j - 1)n - j(j - 1)/2 + k - j$. The program \mathcal{F}_2 considers the cost of traveling between targets as well, as opposed to \mathcal{F}_1 . In this regard, $y_{i\eta}$ is introduced to represent the connectivity between two targets. The inequality (3.15) guarantees that $y_{i\eta}$ is set to 1 only if the i th UAV covers both T_j and T_k .

For further analysis on \mathbf{H}_2 , we note the fact that if a UAV departing from T_0 visits q targets at T_j ($j = 1, 2, \dots, q$), then the cheapest cost of traveling from T_0 to T_j or between T_j 's is less than the optimal cost of traveling to all the q targets by a single UAV. Therefore, since one can construct a feasible tour with q edges if a UAV needs to cover q targets, we readily see that the resultant objective function value r_{\min} of \mathcal{F}_2 satisfies the following relationship: $\mathbf{T}_i^* \leq q r_{\min} \leq q \mathbf{T}^*$, or $\mathbf{T}/\mathbf{T}^* \leq q$, where \mathbf{T}_i^* is the i th UAV's optimal cost of visiting all the targets in \mathcal{T}_i obtained by solving \mathcal{F}_2 and $\mathbf{T} \equiv \max_i \mathbf{T}_i^*$. Comparing to \mathbf{H}_1 , we note that the worst-case is tighter, but the total number of binary variables and constraints for **Phase I** are increased to the order of mn^2 .

2) *With UAV Return*: Interestingly, one can further narrow the gap between \mathbf{T} and \mathbf{T}^* with the aid of the UAVs' return constraint. As a preliminary example, we could have $2r_{\min} \leq \mathbf{T}^*$, if one further imposed the return constraint. This is because the optimal tour cost needed for a UAV should be at least the return (two-way) cost of visiting the farthest target with respect to the starting position of the UAV. Thus, if one solves the original problem including the return constraint via \mathbf{H}_1 , one can expect a solution whose corresponding cost \mathbf{T} satisfies $\mathbf{T}_i^* \leq 2q r_{\min} \leq q \mathbf{T}^*$, or $\mathbf{T}/\mathbf{T}^* \leq q$.

We now take further advantage of the UAVs' return constraint to improve the algorithm quality. We first observe that if a UAV departing at T_0 visits q targets at T_j ($j = 1, 2, \dots, q$) and returns to T_0 , and \mathbf{T}_s^* is the optimal tour cost needed for the UAV, then the cheapest cost $\text{cco}(T_0, T_k)$ of traveling from T_0 to T_k via only T_j , i.e., $\text{cco}(T_0, T_j) + \text{cco}(T_j, T_k)$, is less than \mathbf{T}_s^* . Moreover, one can show that $\text{cco}(T_i, T_j) + \text{cco}(T_j, T_k) \leq \mathbf{T}_s^* \quad \forall i, j, k$, where $i, j, k \in \{0, 1, \dots, q\}$. This observation results in two additional nonexact approaches. The first approach \mathbf{H}_3 minimizes the cost of traveling from T_0 to the second target position (possibly T_0 itself)

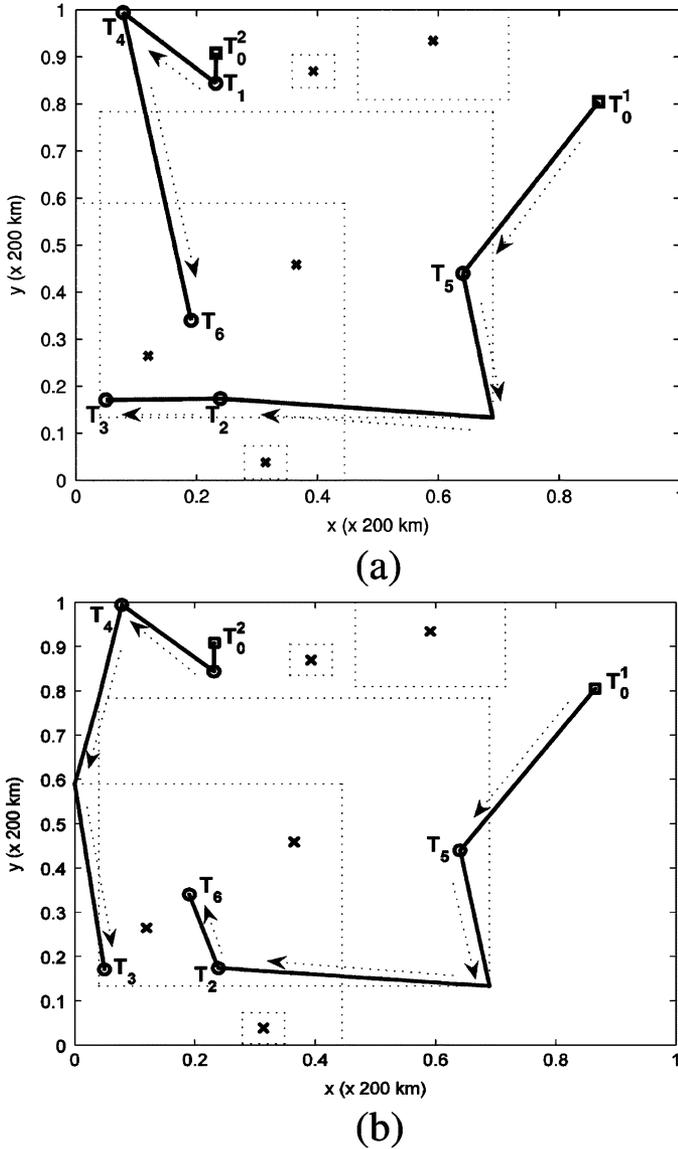


Fig. 1. Comparison of optimal and suboptimal tours. (a) Optimal tour for the two UAVs obtained by \mathbf{E} . (b) Suboptimal tour for the two UAVs obtained by \mathbf{H}_1 or \mathbf{H}_2 .

via the first target position. More generally, the second approach \mathbf{H}_4 minimizes the cost of traveling from any T_j (including T_0) to next two target positions. As a result, \mathbf{H}_3 and \mathbf{H}_4 employ the following programs, \mathcal{F}_3 : minimize r subject to (3.13)–(3.15)

$$2C_0(i, j)x_{ij} \leq r \quad \forall i, j \quad (3.17)$$

$$C_0(i, j)x_{ij} + c(\eta(j, k))y_{i\eta(j, k)} \leq r \quad \forall i, j, k (j < k) \quad (3.18)$$

and \mathcal{F}_4 : minimize r subject to (3.13)–(3.15) and

$$c(\eta(j, k))y_{i\eta(j, k)} + c(\eta(k, l))y_{i\eta(k, l)} \leq r \quad \forall i, j, k, l (j < k, k < l) \quad (3.19)$$

where $i \in \{1, \dots, m\}$, $j, k, l \in \{1, \dots, n\}$ and $\eta(j, k) = (j-1)n - j(j-1)/2 + k - j$. The terms on the left-hand sides of (3.17)–(3.19) correspond to $\text{cco}(T_0^i, T_j) + \text{cco}(T_j, T_0^i)$, $\text{cco}(T_0^i, T_j) + \text{cco}(T_j, T_k)$ and $\text{cco}(T_j, T_k) + \text{cco}(T_k, T_l)$, respectively. We note that (3.17) is not

included in \mathcal{F}_4 , because it appears to be not as useful as in \mathbf{H}_3 in terms of improving the worst-case bound.

With the scheme \mathbf{H}_3 , one can construct a feasible path with a cost of $(2(q-1)/3 + 1)r_{\min}$, if $q = 3k + 1$ ($k = 0, 1, 2, \dots$); otherwise, the cost is $2\lceil q/3 \rceil r_{\min}$, where r_{\min} is the resultant objective function value of \mathcal{F}_3 . Since $r_{\min} \leq \mathbf{T}^*$, thus, $\mathbf{T}/\mathbf{T}^* \leq 2(q-1)/3 + 1$, if $q = 3k + 1$ ($k = 0, 1, 2, \dots$); otherwise, $\mathbf{T}/\mathbf{T}^* \leq 2\lceil q/3 \rceil$. Similarly, the construction of a feasible tour may yield the bound for \mathbf{H}_4 , $\mathbf{T}/\mathbf{T}^* \leq \lceil (q+1)/2 \rceil$.

IV. NUMERICAL EXAMPLES

In this section, we present test examples to show the efficacy of the proposed algorithms. We fix the number m of UAVs, the capacity limit q and the number n of targets, while we randomly generate one hundred sets of data including the number, locations, strengths (ranges) of threats (missiles) and the locations T_0^i of UAVs in operation and the locations T_i of targets. We assume that, for the purpose of visualization, UAVs operate at a fixed altitude 2 (km) over the generated hostile area $[0, 200] \times [0, 200]$ (km), i.e., $\mathbf{X} = [0, 200] \times [0, 200] \times 2$ (km). Based on each data model, the cost matrix C_0 and the vector c in Section II are calculated. For the simulation with each set of data, we measure the computational time to obtain a solution using the proposed methods.

We first show how the solutions differ by the proposed approaches. Fig. 1(a) depicts one scenario among the one hundred data sets when $m = 2$, $q = 3$ and $n = 6$. In the figure, five missile units, marked as “x,” with different ranges are deployed in \mathbf{X} and the size of each dotted rectangle, i.e., $\mathcal{C}(\mathcal{P}_\rho)$ in Section III, around each missile unit represents the strength (range) of each unit. Using the formula given in (2.2) and Dijkstra’s algorithm, we calculate the cheapest cost of traveling between all the necessary vertices, i.e., T_0^1, T_0^2, T_j ’s ($j = 1, 2, \dots, 6$) and the four corner points of each rectangle. Given the cost, the exact approach \mathbf{E} yields the optimal tour, as shown in Fig. 1(a), for the two UAVs when the UAVs’ return requirement is ignored. As plotted in the figure, the first (second) UAV visits targets in the order of T_5, T_2 and T_3 (T_1, T_4 and T_6) and the tour cost \mathbf{T}^* is 226.31. However, different assignments demanding higher cost are obtained with nonexact method \mathbf{H}_1 (or \mathbf{H}_2), as shown in Fig. 1(b). One can see that in this case the first (second) UAV covers T_6 (T_3) instead of T_3 (T_6), which increases the tour cost to 265.71. The figures for illustrating the solutions obtained using \mathbf{E}_{ret} and \mathbf{H}_3 (or \mathbf{H}_4) are omitted for brevity. We then measure the average computation times needed and the average \mathbf{T}/\mathbf{T}^* for each of the one hundred cases with fixed m, q and n , when a personal computer equipped with an Intel(R) Pentium 4 CPU 3.40 GHz. is used. When $m = 2$, $q = 3$, and $n = 6$, the execution times (in seconds) required for $\mathbf{E}, \mathbf{H}_1, \mathbf{H}_2, \mathbf{E}_{\text{ret}}, \mathbf{H}_3$, and \mathbf{H}_4 are 0.754, 0.662, 0.681, 0.813, 0.685, and 0.748, respectively. The associated \mathbf{T}/\mathbf{T}^* are 1, 1.280, 1.124, 1, 1.050, and 1.050, respectively. When $n = 5$, $q = 4$ and $n = 20$, the execution times required for $\mathbf{E}, \mathbf{H}_1, \mathbf{H}_2, \mathbf{E}_{\text{ret}}, \mathbf{H}_3$, and \mathbf{H}_4 are 5000+, 0.931, 4.054, 5000+, 5.861, and 224.0, respectively. The associated \mathbf{T}/\mathbf{T}^* are 1+, 1.685+, 1.391+, 1+, 1.292+, and 1.292+, respectively, where $x+$ denotes a number greater or equal to x . We note that the exact optimization procedure was forced to terminate at 5000 s and returns the best solution up to that moment when the example with $n = 20$ was tested. These results show that the nonexact methods perform much faster than the exact ones, and \mathbf{H}_2 and \mathbf{H}_3 especially yield a quality solution in a reasonable time when $q \leq 4$. In all the test cases (even when $n = 20$), \mathbf{H}_2 and \mathbf{H}_3 show the performance of yielding a solution in 6 s, as well as guaranteeing $\mathbf{T}/\mathbf{T}^* < 1.5$. In contrast, the computational burden of solving \mathcal{F}_E dramatically increases as n does. These experiments also show that the nonexact methods accounting for the UAVs’ return constraint yield a solution closer to the optimal solution than the ones that ignore the UAVs’ return constraint, as the worst-case analysis suggested in the previous sections.

V. CONCLUDING REMARKS AND FUTURE WORK

We have discussed exact, as well as nonexact, methods to the target assignment problem with and without the UAVs' return constraint. We first showed how the problem can be exactly formulated in MILP which is likely to be cumbersome as the problem size increases. We then showed theoretically as well as numerically that the nonexact methods, especially \mathbf{H}_2 and \mathbf{H}_3 , perform well in terms of optimality and computational complexity when q is small (≤ 4). As mentioned at the beginning of the note, other practical constraints, e.g., timing constraints discussed in [3], could be considered as a next step.

REFERENCES

- [1] R. Ahuja, A. Kumar, K. Jha, and J. Orlin, "Exact and heuristic methods for the weapon target assignment problem," MIT Working Paper, Report No. 4464-03, Jul. 2003.
- [2] M. Alighanbari and J. How, "Decentralized task assignment for unmanned aerial vehicles," in *Proc. 44th IEEE Conf. Decision Control*, Seville, Spain, Dec. 2005, pp. 5668–5673.
- [3] M. Alighanbari, Y. Kuwata, and J. How, "Coordination and control of multiple UAVs with time constraints and loitering," in *Proc. IEEE Amer. Control Conf.*, Denver, CO, Jun. 2003, pp. 5311–5316.
- [4] R. Beard, T. McLain, M. Goodrich, and E. Anderson, "Coordinated target assignment and intercept for unmanned AER vehicles," *IEEE Trans. Robot. Automat.*, vol. 18, no. 6, pp. 911–922, Dec. 2002.
- [5] J. Bellingham, M. Tillerson, A. Richards, and J. P. How, "Multi-task allocation and trajectory design for cooperating UAVs," in *Proc. 2nd Annu. Conf. Cooperative Control Optimization*, Gainesville, FL, Nov. 2001, pp. 1–19.
- [6] D.-W. Gu, I. Postlethwaite, and Y. Kim, "A comprehensive study on flight path selection algorithms," in *Proc. IEE Semin. Target Tracking*, Birmingham, U.K., Mar. 2006, pp. 77–89.
- [7] M. Haimovich, A. H. G. Rinnooy Kan, and L. Stougie, "Analysis of heuristics for vehicle routing problems," in *Vehicle Routing: Methods and Studies*. Amsterdam, The Netherlands: Elsevier, 1988.
- [8] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 2nd ed. Berlin, Germany: Springer, 2002.
- [9] C. Li and D. Simchi-Levi, "Worst-case analysis of heuristics for multi-depot capacitated vehicle routing problem," *ORSA J. Comput.*, vol. 2, pp. 64–73, 1990.
- [10] A. Manne, "A target-assignment problem," *Oper. Res.*, vol. 6, pp. 346–351, 1958.
- [11] Y. Marinakis and A. Migdalas, "Heuristic solutions of vehicle routing problems in supply chain management," in *Combinatorial and Global Optimization*, 1st ed. London, U.K.: World Scientific, 2002.
- [12] J. Ryan, G. Bailey, J. Moore, and W. Carlton, "Reactive tabu search in unmanned aerial reconnaissance simulations," in *Proc. Winter Simulation Conf.*, Washington, DC, Dec. 1998, pp. 873–879.
- [13] M. Savelsbergh, "Local search in routing problems with time windows," *Ann. Oper. Res.*, vol. 4, pp. 285–305, 1985.
- [14] A. Schaerf, "A survey of automated timetabling," *Artif. Intell. Rev.*, vol. 13, pp. 87–127, 1999.

A Counterexample to a Conjecture of Gurvits on Switched Systems

Michael Margaliot

Abstract—We consider products of matrix exponentials under the assumption that the matrices span a nilpotent Lie algebra. In 1995, Gurvits conjectured that nilpotency implies that these products are, in some sense, simple. More precisely, there exists a uniform bound l such that any product can be represented as a product of no more than l matrix exponentials. This conjecture has important applications in the analysis of linear switched systems, as it is closely related to the problem of reachability using a uniformly bounded number of switches. It is also closely related to the concept of nice reachability for bilinear control systems. The conjecture is trivially true for the case of first-order nilpotency. Gurvits proved the conjecture for the case of second-order nilpotency using the Baker–Campbell–Hausdorff formula. We show that the conjecture is false for the third-order nilpotent case using an explicit counterexample. Yet, the underlying philosophy behind Gurvits' conjecture is valid in the case of third-order nilpotency. Namely, such systems do satisfy the following nice reachability property: any point in the reachable set can be reached using a piecewise constant control with no more than four switches. We show that even this form of finite reachability is no longer true for the case of fifth-order nilpotency.

Index Terms—Bang-bang control, bilinear control systems, differential inclusions, Lie algebra, Lie's product formula, optimal control, singular control, switched linear systems, Fuller's problem.

I. INTRODUCTION

Consider the *linear switched system*

$$\dot{\mathbf{x}}(t) = A_{\sigma(t)}\mathbf{x}(t) \quad (1)$$

where $\mathbf{x}(\cdot) \in \mathbb{R}^n$, $\sigma: \mathbb{R}_+ \rightarrow \{0, 1\}$ is a piecewise-constant function called the *switching function*, and $A_0, A_1 \in \mathbb{R}^{n \times n}$. Roughly speaking, (1) models a system that can "switch" between the two linear subsystems $\dot{\mathbf{x}}(t) = A_0\mathbf{x}(t)$ and $\dot{\mathbf{x}}(t) = A_1\mathbf{x}(t)$. Switched systems have numerous applications and are recently attracting considerable interest (see, e.g., [8] and [15]).

It is usually assumed that the switching function has a finite number of discontinuities on every bounded interval of time. Let Σ_k denote the set of switching functions with up to $k - 1$ discontinuities on $[0, T]$, and denote

$$\mathcal{F}(A_0, A_1, k, T) := \left\{ \exp(C_1 t_1) \exp(C_2 t_2) \dots \exp(C_k t_k) : \right. \\ \left. C_i \in \{A_0, A_1\}, t_i \geq 0, \sum_{j=1}^k t_j = T \right\}$$

that is, a product of up to k exponentials of the system's matrices with "total time" T . Then, any solution \mathbf{x} of (1) corresponding to some $\sigma \in \Sigma_k$ satisfies the following property. For any $T \geq 0$, $\mathbf{x}(T) = H\mathbf{x}(0)$ for some $H \in \mathcal{F}(A_0, A_1, k, T)$. Thus, the matrices in the set \mathcal{F} are in

Manuscript received August 28, 2006; revised December 31, 2006. Recommended by Associate Editor D. Angeli. This research was supported in part by the Israel Science Foundation. An abridged version of this paper was presented at the 45th IEEE Conference on Decision and Control (CDC'06).

The author is with the School of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel (e-mail: michaelm@eng.tau.ac.il).

Digital Object Identifier 10.1109/TAC.2007.899047