Military Aircraft Mission Planning - Efficient Model-Based Metaheuristic Approaches

Nils-Hassan Quttineh Torbjörn Larsson

November 25, 2014

Abstract

We consider a military mission planning problem where a given fleet of aircraft should attack a number of ground targets. At each attack, two aircraft need to be synchronized in both space and time. Further, there are multiple attack options against each targets, with different target effects. The objective is to maximize the outcome of the entire attack, while also minimizing the mission timespan. Real-life mission planning instances involve only a few targets and a few aircraft, but are still computationally challenging.

We present metaheuristic solution methods for this problem, based on an earlier presented model. The problem includes three types of decisions: attack directions, task assignments and scheduling, and the solution methods exploit this structure in a two-stage approach. In an outer stage, a heuristic search is performed with respect to attack directions, while in an inner stage the other two decisions are optimized, given the outer stage decisions. The proposed metaheuristics are capable of producing high-quality solutions and are fast enough to be incorporated in a decision support tool.

Keywords: Military operations research, Generalized vehicle routing, Mixed integer programming, Matheuristics, Decision support.

1 Problem setting

Military mission planning is a complex task, and it is still to a large extent performed manually by experienced personnel. This is time-consuming and the quality of the plans depend highly on the expertice and skills of the planners. Most military missions of course involve some kind of vehicles, and there is a vast literature on operations research approaches to various military vehicle assignment and path planning problems, in recent years especially involving unmanned aerial vehicles (UAV). Some examples from this literature, adjacent to our work, are [2, 6, 8, 9].

We here address a mission planning problem, introduced in Quttineh *et al.* [5], where a fleet of aircraft should attack a number of ground targets, subject to certain restrictions, such as precedence relationship between the target attacks. The planning of such a mission is typically made close to when the mission actually takes place (say, within 24 hours); one important reason for this is that the planning can then be based on the most recent information. The time needed for the chain of planning is of the order of several hours. The work presented in [5] and in this paper has been performed in collaboration with an industrial partner.

The aircraft fleet is typically deployed from a base situated far away from the targets. The geographic area of interest, where the attacks shall take place, is referred to as the target scene, see Figure 1, and it is defined by a line of entrance and a line of exit for the aircraft. These enter the scene at the entry line, and when the mission has been carried out they leave at the exit line and turn back to a base. In addition to the ground targets there are enemy defense positions, like surface-to-air missiles (SAMs), and also protected objects, like hospitals and schools, not to be touched by the attacks. All objects are assumed to be stationary with known positions.

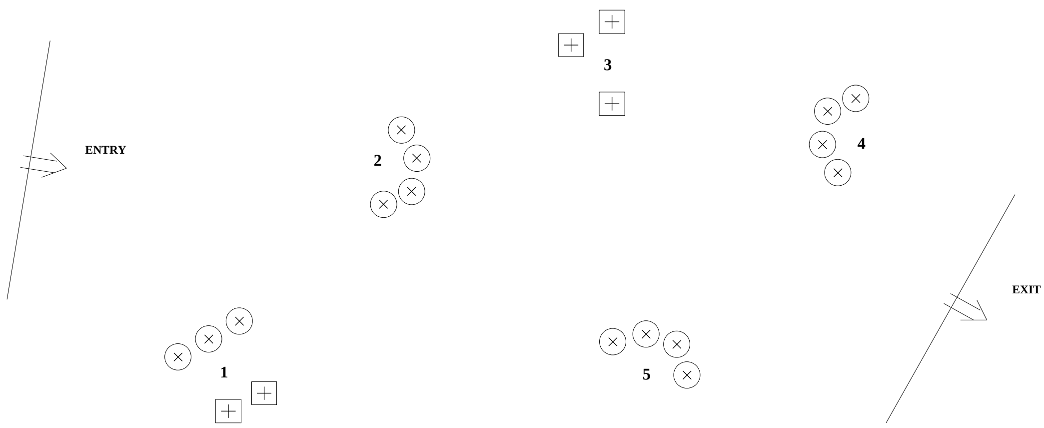

Figure 1: An example of a target scene, including five targets and nearby SAMs ($\times$) and protected objects ($+$). The entry line is to the left and the exit line to the right.

The goal of the mission is to gain maximal expected effect against all targets within a short timespan. The mission time is from the first aircraft passing the line of entry until the last aircraft passes the exit line. We assume air supremacy and that the threat from defense positions is taken into account by restricting the aircraft not to fly through defended airspace (or by allowing them to fly only in airspace with an acceptably low level of risk). Weapons may pass through defended airspace, but at the risk of being shot down, that is, with a lower expected effect on the target.

The diameter of a target scene is around 100 km, while distances between targets are at most a few kilometers, and the timespan of the attacks is around a quarter of an hour. A large attack would involve 6–8 targets and 4–6 aircraft, and would require several hours, at least, of manual planning to find an acceptable attack plan.

The expected effect of an attack depends on the kind of weapon being used, which is decided in advance, but also on the direction of the impact and the weapon's kinetic energy, which in turn depends on the velocity and altitude of the aircraft at the time of the launch. Each aircraft has an armament capacity which limits the number of attacks it can perform. In addition to the armament, aircraft can also carry illumination laser pods, used for guiding weapons. Once the mission has been planned, it is known how each aircraft shall be equipped to be able to fulfil its tasks.

Each target needs to be attacked exactly once and the attack requires two aircraft to rendez-vous: one that launches the weapon and one that illuminates the target with a laser beam from the launch of the weapon until its impact. At the time of the launch, the flight direction of the attacking aircraft is towards the target, while the illuminating aircraft usually passes by the target relatively far away. We will refer to attack and illumination alternatives as positions.

Any attack is associated with a risk of collateral damage, and for each tentative attack position one can estimate an area of risk for such damage. To prevent any collateral damage, an attack positions is allowed only if no protected object is located within this area. Hence, for each target one can derive a set of feasible attack positions, and each of these is associated with a set of compatible illumination positions.

To summarize, the problem is to define an attack sequence for the given fleet of aircraft so that each target is attacked and illuminated exactly once, in a synchronized manner using compatible attack and illumination positions. Further, the sequence must respect given precedence restrictions, if any. The goal is to maximize the overall expected effect of the attacks on the targets while also minimizing the total timespan of the mission. This problem belongs to the class of vehicle routing problems, where attack and illumination positions are described by nodes, and paths between such positions are described by arcs.

The problem involves three types of decisions. First, the choice of attack direction against each target. Second, which two aircraft shall deliver the weapon and illuminate each target. Third, the schedule for each aircraft's assigned tasks in the mission. Note that these decisions are not independent nor ordered in an hierarchy. We exploit this structure in tailored metaheuristics, based on a partitioning of the decisions. In an outer stage, a heuristic search is made with respect to attack directions, while in an inner stage the other two decisions are optimized, given the outer stage decisions.

The paper is organized in the following way. In Section 2 we introduce notations and present a network-based mathematical model for this mission planning problem. In Section 3 we present three metaheuristics for the problem. In Section 4 we present numerical results, and finally conclusions and future work are discussed in Section 5.

2 Mathematical description

For a specific type of aircraft and a target requiring a specific type of weapon, one can derive a possible attack space against the target, represented by inner and outer radii of attack. This space is divided into a number of sectors, in which we introduce discrete attack positions, and for each and one of these we create compatible illumination positions. We have chosen to use six sectors with at most three possible attack positions in each. Since only attack positions that are feasible with respect to collateral damage are included, the number of such positions might vary between sectors. For any attack position, an expected effect on the target can be calculated. We introduce two illumination alternatives per sector, which are both compatible with all its attack positions. The two illumination alternatives utilize the same flight path but differ in flight direction, which is roughly clockwise or counter-clockwise with respect to the target. See Figure 2 for an illustration of the discretization used. It is of course possible to generalize this discretization by including attack options on discrete altitude layers.

Attack and illumination alternatives are represented by nodes. By further introducing dummy nodes associated with the crossings of the entry and exit lines of the target scene, and modelling possible aircraft movements by arcs, the mission planning problem can partly be represented by a network. Since each target shall be attacked and illuminated exactly once and an aircraft can not both attack and illuminate a target, this network only contains arcs between nodes corresponding to different targets, or from or to the dummy nodes. Further, only arcs respecting the precedence constraints are included in the network.

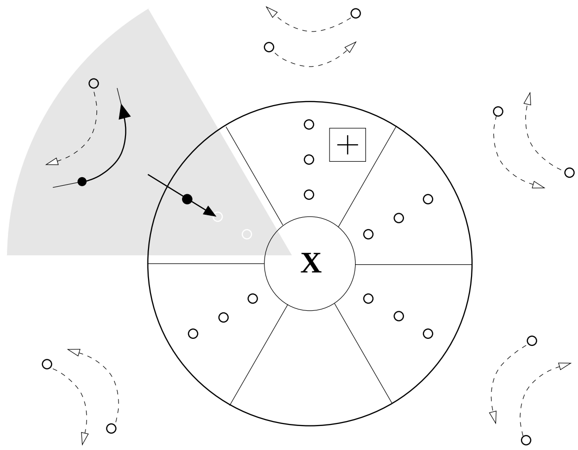

Figure 2: The feasible attack space defined by inner and outer radii, and divided into six sectors with a coarse discretization of three attack positions and two illumination alternatives in each. The latter are compatible with all the attack positions in the same sector. A protected building is present in the north sector, and due to the risk of collateral damage no feasible attack positions are available in the south sector. A pair of compatible attack and illumination positions is marked, where the arrows indicate the flight directions.

The arcs of the network have two attributes: expected effect and travel time. The effect attribute is present only for an arc that is leaving an attack node, and it then equals the resulting expected effect against the target.

For calculating the travel time attribute, we must between all positions described by nodes find flight paths that are minimal with respect to time. These paths must comply with restrictions on the aircraft dynamics, in particular restrictions on velocity and turning radius. The paths also need to be safe, meaning that the aircraft cannot pass through defended airspace.

The problem of finding an optimal flight path from a given starting point to a given destination, while avoiding defended airspace, is referred to as the aircraft routing problem. We refer to [11] and [1] for examples of algorithms that can be used to solve this problem, and to [7] for a closely related routing problem which gives rise to a shortest-path problem with side constraints.

In our numerical experiments we use a flight path generator which is provided by our industrial partner; it is based on a discretization of the airspace and a calculation of a shortest path. The length of the path is then converted into a time required to traverse it. In our network representation, each node is associated with both a location and a flight direction. Therefore, travel times will in general be asymmetric.

2.1 Notation

Given is a set R of aircraft and a set M of targets to be attacked. Each feasible attack space is divided into six sectors, and we let G denote the set of all sectors for all targets while G_m is the subset of sectors that belong to target $m \in M$.

Each target $m \in M$ is associated with a set of attack nodes, N_m^A, and illumination nodes, N_m^I. We also define the set $N_m = N_m^A \cup N_m^I$, that is, the set of all nodes associated with a target. Let N denote all the nodes in the network, including a dummy origin, o, and a dummy destination, d.

Let A be the set of all arcs in the network, let A_g be the subset of arcs with heads at attack nodes in sector $g \in G$, and let I_g be the subset of arcs with heads at illumination nodes in sector $g \in G$. We also define the set of all arcs between two targets m and n, $A_{mn} = \{(i,j) \in A : i \in N_n, j \in N_m\}$.

The effect attribute is denoted c_{ij}^r and defined for all $(i,j) \in A$ and $r \in R$. For arcs leaving attack nodes, its value is the expected effect of the attack, and otherwise the value is zero. Further, let T_{ij}^r denote the time needed for aircraft $r \in R$ to traverse arc $(i,j) \in A$. We also introduce T_{max}, either as a pessimistic bound on the total mission time or as a given upper time limit for the duration of the mission.

Let q_m denote the number of weapons to be used towards target $m \in M$. Each aircraft $r \in R$ is limited to carry at most Γ weapons. To capture precedence relations, let P denote the set of ordered pairs (m,n) of targets such that target m cannot be attacked before target n. If no precedence is given, this set is empty. Further, let K be the number of nonempty subsets of targets, and let S_k denote each such subset.

We introduce two types of binary decision variables: z_g and x_{ij}^r. Variable z_g equals one exactly when a sector $g \in G$ is active, that is, if an attack is performed from the sector. Variable x_{ij}^r equals one exactly when aircraft $r \in R$ traverses arc $(i,j) \in A$. To obtain a model with a stronger LP relaxation, we introduce an auxiliary binary variable, u_{mn}^r, that equals one exactly when aircraft r travels directly from target m to target n. The variables u_{mn}^r are defined on the set of ordered pairs $U = (M \times M) \setminus P$.

Further, we introduce the continuous time variables, t_i, t_m^A and t_m^I. Variable t_i is the time at which node $i \in N$ is visited, by some aircraft, and it equals zero if the node is not visited by any aircraft. The starting time for all aircraft is $t_o = 0$ while the time of the last aircraft to exit the target scene is t_d. Variables t_m^A and t_m^I are the times of the attack and illumination, respectively, of each target $m \in M$.

2.2 The mathematical model

The goal is to maximize the total expected effect against the targets, while also minimizing the mission timespan. We choose to optimize a weighted combination of these two objectives, using a parameter $\mu > 0$. This yields a solution that is Pareto optimal with respect to the two objectives. The mathematical model for the Military Aircraft Mission Planning Problem (MAMPP) is given below.

$$\max \sum_{r \in R} \sum_{(i,j) \in A} c_{ij}^r x_{ij}^r - \mu t_d$$

$$\begin{array}{llll}
s.t. \quad \displaystyle\sum_{(o,j) \in A} x_{oj}^r & = 1, & r \in R & (1) \\
\displaystyle\sum_{(i,d) \in A} x_{id}^r & = 1, & r \in R & (2) \\
\displaystyle\sum_{(i,k) \in A} x_{ik}^r - \sum_{(k,j) \in A} x_{kj}^r & = 0, & r \in R,\ k \in N \setminus \{o, d\} & (3) \\
\displaystyle\sum_{r \in R} \sum_{g \in G_m} \sum_{(i,j) \in A_g} x_{ij}^r & = 1, & m \in M & (4) \\
\displaystyle\sum_{r \in R} \sum_{g \in G_m} \sum_{(i,j) \in I_g} x_{ij}^r & = 1, & m \in M & (5) \\
\displaystyle\sum_{r \in R} \sum_{(i,j) \in A_g} x_{ij}^r - \sum_{r \in R} \sum_{(i,j) \in I_g} x_{ij}^r & = 0, & g \in G & (6) \\
\displaystyle\sum_{g \in G_m} z_g & = 1, & m \in M & (7) \\
\displaystyle\sum_{(i,j) \in A_g \cup I_g} x_{ij}^r & \leq z_g, & r \in R,\ g \in G & (8) \\
\displaystyle\sum_{m \in M} \sum_{g \in G_m} \sum_{(i,j) \in A_g} q_m x_{ij}^r & \leq \Gamma, & r \in R & (9) \\
\displaystyle\sum_{(i,j) \in A_{mn}} x_{ij}^r & = u_{mn}^r, & r \in R,\ (m,n) \in U & (10) \\
\displaystyle\sum_{m \in S_k} \sum_{n \in S_k} u_{mn}^r & \leq |S_k| - 1, & r \in R,\ k = 1, \ldots, K & (11) \\
\displaystyle t_i + \sum_{r \in R} T_{ij}^r x_{ij}^r - T_{max} \left(1 - \sum_{r \in R} x_{ij}^r \right) & \leq t_j, & (i,j) \in A & (12) \\
\displaystyle t_i - T_{max} \sum_{r \in R} \sum_{(i,j) \in A} x_{ij}^r & \leq 0, & i \in N & (13) \\
\displaystyle\sum_{i \in N_m^A} t_i & = t_m^A, & m \in M & (14)
\end{array}$$

$$
\begin{aligned}
\sum_{i \in N_m^I} t_i &= t_m^I, && m \in M && (15)\\
t_m^A &= t_m^I, && m \in M && (16)\\
t_m^A &\geq t_n^A, && (m,n) \in P && (17)\\
x_{ij}^r &\in \{0,1\}, && r \in R,\ (i,j) \in A && (18)\\
z_g &\in \{0,1\}, && g \in G && (19)\\
u_{mn}^r &\in \{0,1\}, && (m,n) \in U && (20)\\
t_m^A,\ t_m^I &\geq 0, && m \in M && (21)\\
t_i &\geq 0, && i \in N && (22)
\end{aligned}
$$

Constraints (1) – (3) ensure that each aircraft enters and leaves the target scene via the dummy nodes, constraint (4) ensures that each target $m \in M$ is attacked exactly once, while (5) does the same for the illumination. Constraint (6) ensures that the attack and the illumination against each target are performed in the same sector.

Constraint (7) states that exactly one sector should be active for each target, and constraint (8) states that each aircraft can visit each target at most once, either for attacking or for illuminating the target, and then only in the active sector. Constraint (9) limits each aircraft to utilize at most Γ weapons.

The variables u_{mn}^r and x_{ij}^r are coupled by constraint (10), and (11) is the standard subtour eliminating constraint. Note that the latter constraint is redundant, but by including all such constraints a significantly stronger LP relaxation is obtained. (Even for a scenario with ten targets, there are only at most $2^{10} - 1 = 1023$ such constraints, which is a relatively small number compared to the overall number of constraints.)

Time is propagated by constraint (12), making sure that if an aircraft r traverses arc (i, j), node j is visited no earlier than the time of the visit to node i plus the time needed to traverse the arc. Note that constraint (12) also eliminates subtours. Note also that the construction of this constraint relies on that each arc can be traversed by at most one aircraft, which follows from (3) and (4). For $j = d$, constraint (12) defines the total mission time, t_d, since all aircraft end up at the dummy destination.

Constraint (13) enforces that $t_i = 0$ holds if node i is not visited by any aircraft, and constraints (14) and (15) assign the correct times of attack and illumination, respectively, for each target m. Constraint (16) states that these times must coincide, and constraint (17) imposes the precedence order. Note that although all arcs in the network respect the given precedence order, constraint (17) is not redundant, since if these constraints are removed, precedence relations between non-consecutive targets can be violated. Finally, constraints (18) – (22) are definitional constraints.

The variables z_g and constraints (7) and (8) are actually not needed in the model, since the attack constraints (4) – (6), the time propagation constraint (12), and the time synchronization constraint (16) imply that there are values of z_g such that (7) and (8) are satisfied. However, the variables z_g and constraints (7) and (8) are included since they are instrumental for the design of the metaheuristics to be presented.

The MAMPP and its mathematical model was introduced in Quttineh *et al.* [5], where a thorough description of the problem and model can be found. For the scenario described in Figure 1, the resulting network model contains 110 nodes and 9771 arcs. With four aircraft, the model of MAMPP contains roughly 39,000 binary variables and 130 continuous variables, and roughly 11,000 constraints.

The model of MAMPP belongs to the class of vehicle routing problem, but has the following non-standard characteristics.

i) Since exactly one attack node and exactly one illumination node shall be visited for each target, it is a generalized vehicle routing problem.

ii) Since the attack and illumination nodes for a target need to be compatible, the visits to nodes are coupled by side constraints.

iii) The times of arrival to the compatible attack and illumination nodes for a target are required to be synchronized.

iv) The order in time of the visits to the pairs of attack and illumination nodes for the targets are restricted by precedence relations.

Numerical results in [5] show that standard solver software can manage to solve small-sized problem instances, but even a mid-sized scenario with four targets and two aircraft, without any precedence on the targets, is not solvable to proven optimality within a reasonable time frame (more than three hours). This implies that it is not reasonable to base a decision support system on standard solver software, since the solver in such a system needs to be fast enough to allow interactive use.

Below, we explore the possibility to find high-quality solutions by embedding a standard solver software into tailored metaheuristic searches, and thereby boosting its performance.

2.3 Example solution

We here illustrate a solution for the scenario shown in Figure 1, when four aircraft are used. All numerical data used in the scenario were provided by our industrial partner. For this problem instance, which is no. 16 in Table 1, the aircraft have dedicated roles (only attacks or only illuminations), an aircraft can attack at most three targets, and the prescribed attack order is 1–2–3–4–5.

The model was implemented using AMPL, see Fourer *et al.* [3], and solved with the MIP solver CPLEX [4]. After preprocessing, the problem consists of 5,905 rows, 4,507 columns and 4,390 binary variables. An optimal solution is shown in Figure 3. The total mission time is $t_d = 532$ seconds. The routes for the attackers are shown as dashed lines, while the routes for the illuminating aircraft are shown as solid lines. The maximal effect is achieved against all targets except the third one, to which a slightly reduced effect is achieved.

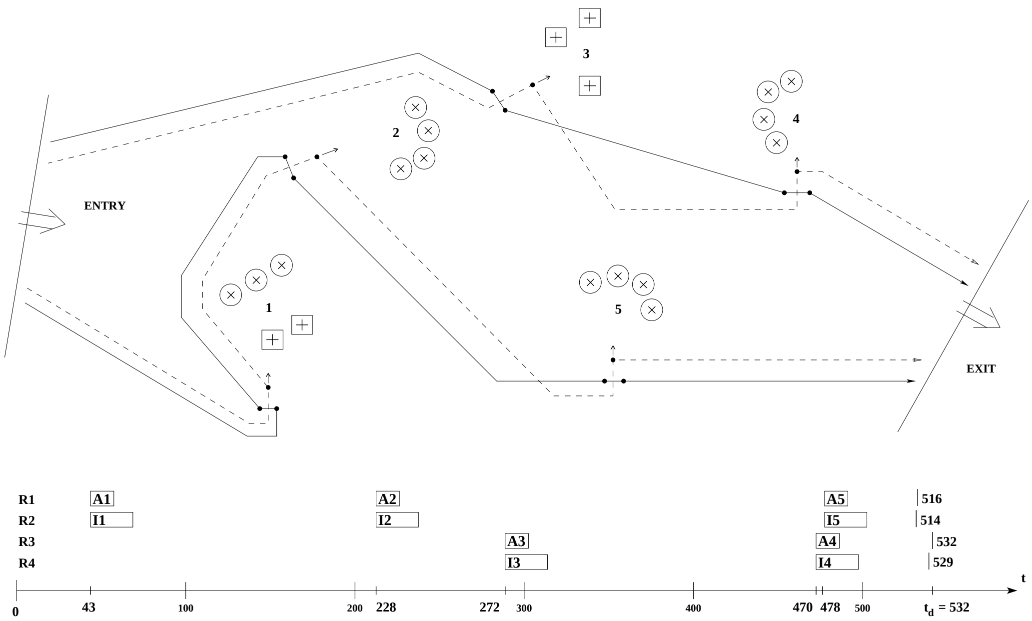

Figure 3: Optimal solution to problem instance no. 16 in Table 1. This instance has target positions as shown in Figure 1, and with prescribed attack order 1–2–3–4–5. Shown are aircraft routes, chosen attack and illumination nodes against each target, and the time schedule for the mission.

3 Partitioning-based metaheuristics

The model of MAMPP presented above contains a large number of binary routing variables. Also, the time-propagating constraint (12) are of big-M type, since T_{max} is usually quite large, which makes the linear programming relaxation weak. For these two reasons, the model is computationally challenging.

In this section we present three tailored metaheuristic approaches for the MAMPP, based on a common framework. The first of these metaheuristics is a simulated annealing procedure, while the other two are tabu searches. Readers unfamiliar to metaheuristics are referred to, for example, the book by Talbi [10].

As mentioned above, we identify three types of decisions in this problem: the choice of attack direction, that is, active sector, against each target, the assignment of aircraft against each target for attack and illumination, and the schedule of tasks for each aircraft. Note that the values of all time variables in the model of MAMPP are direct consequences of these decisions.

The common framework for the heuristics is the partitioning of the three types of decisions into two stages. In an outer stage, the choice of active sectors for all targets is optimized by a metaheuristic search. In an inner stage, all other decisions are optimized, given every tentative choice of active sectors proposed by the outer stage heuristic. The reason for this solution strategy is the observation that if we select active sectors for all targets, that is, forcing all other sectors to be inactive, most of the binary routing variables will be eliminated. This reduces the problem size significantly, so that the remaining, restricted, problem becomes solvable within a reasonable time frame (optimality typically proven within one minute).

More specifically, we introduce the index set $FIX \subset G$ that contains exactly one sector from each target. Letting $z_g = 1$, $g \in FIX$, and $z_g = 0$ otherwise, constraint (7) is clearly fulfilled, and constraint (8) will force a majority of all routing variables to become zero. For example, in a problem instance with five targets and two aircraft, the number of binary variables after AMPL and CPLEX preprocessing is reduced by 94%, from 15,952 to 925. A similar reduction holds for the number of rows.

For a given index set FIX, the inner stage problem is solved using CPLEX. The computation time required for this problem is not insignificant, and it is hence reasonable to prevent evaluating the same choice of FIX twice. To this end, we maintain for all three metaheuristics a list of already examined sets FIX. To initialize the index set FIX in the metaheuristic search we solve the linear programming relaxation of the MAMPP model and choose at random for each target a sector with $z_g > 0$ in the optimal solution.

3.1 Simulated annealing

We have used a straightforward Simulated Annealing (SA) procedure. The neighbourhood of a given index set FIX is the family of all sets where exactly one target's active sector has been changed. This neighbourhood contains $|G| - |M|$ neighbours. A pseudo-code description is given in Algorithm 1.

Algorithm 1 Simulated Annealing

Initialization: Generate set FIX as described. Temperature $T = 2$. Cooling factor $COOL = 0.97$.

1: **for** $k = 1 \ldots MAXITER$ **do**
2: Create a tentative FIX set by a random change of sector for a random target.
3: If the tentative FIX has not already been examined, solve the inner stage problem.
4: Accept or reject the tentative FIX according to the standard SA rule.
5: Record the solution if it is the best one ever found.
6: Update temperature: $T = T \cdot COOL$.
7: **end for**

3.2 Basic tabu search

The two tabu searches are based on the same type of change of the current set FIX, although the neighbourhoods are of different sizes. The change of this set corresponds to replacing the active sector for a specific target to one of its adjacent sectors, while maintaining active sectors for all other targets, as illustrated in Figure 4.

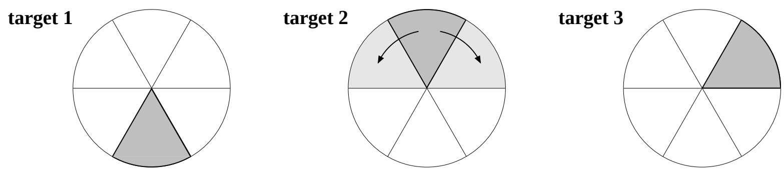

Figure 4: Possible changes if the active sector of target 2 is replaced by an adjacent sector, while the active sectors for the other two targets are maintained.

In the Basic Tabu Search (BTS), we choose a target at random and solve the inner stage problem for the two sectors that are adjacent to the current one. This neighbourhood thus consists of only two neighbours. The set FIX is updated to the best of these two alternatives. To avoid making a modification for the same target twice in a row, the target is placed on a tabu list, that is of length one only. (Recall that the number of targets is small.) An outline of this metaheuristic is found in Algorithm 2.

Algorithm 2 Basic Tabu Search

Initialization: Generate set FIX as described.

for $k = 1 \ldots MAXITER$ **do**
  Choose a target at random which is not on the tabu list. Keep the sectors of all other targets fixed.
  Consider the tentative FIX sets that correspond to the two adjacent sectors for the chosen target.
  If both sets have already been examined, a new target is chosen at random.
  Solve the inner stage problem for each of the tentative FIX sets, unless already examined.
  Choose the best of the two evaluated tentative FIX sets.
  Record the solution if it is the best one ever found.
  Update the tabu list.
end for

3.3 Expanded tabu search

The Expanded Tabu Search (ETS) differs from BTS in the respect that the evaluation of adjacent sectors that is in BTS performed for a specific target is now instead performed for all the targets before any change of FIX is made. However, if an improvement is found, compared to the current FIX set, the evaluation of all the remaining targets is aborted and the current tentative FIX set is chosen. Otherwise, if no improvement is found, the best of all the evaluted tentative FIX sets is chosen.

To avoid any bias, the targets are evaluated in a random order. This can of course have an effect on the update of FIX only when an improvement is found. The neighbourhood in ETS, which is clearly an expansion of the one used in BTS, consists of $2 \cdot |M|$ neighbours. There is no conventional tabu list, but only the list of already examined sets FIX. This metaheuristic is outlined in Algorithm 3.

4 Numerical results

In our numerical experiments we consider a homogeneous fleet of aircraft. As is always the case with vehicle routing problems with identical resources, symmetry is an issue. To partially resolve this issue we add to the model of MAMPP constraints stating that the first target is attacked by a specific aircraft and illuminated by another specific aircraft, which does not cause any loss of generality.

Algorithm 3 Expanded Tabu Search

Initialization: Generate set FIX as described.

for $k = 1 \dots MAXITER$ **do**
  Generate a random permutation of the targets, $TARGETS$.
  for $m \in TARGETS$ **do**
    Consider the tentative FIX sets that correspond to the two adjacent sectors for the chosen target.
    If both sets have already been examined, continue with next target.
    Solve the inner stage problem for each of the tentative FIX sets, unless already examined.
    If an inner stage problem yields an improvement, the evaluation of all the remaining targets is aborted and the current tentative FIX set is chosen.
  end for
  If no improvement was found, choose the best of all the evaluated tentative FIX sets.
  Record the solution if it is the best one ever found.
end for

The aircraft fleet can be required to be utilized in different ways. A given aircraft, r_1, can be dedicated to operate pairwise with another aircraft, r_2, throughout the mission, which is easily modelled as $u_{mn}^{r_1} = u_{mn}^{r_2}$, for all $(m, n) \in U$. An aircraft r can also be given a dedicated role throughout the mission, that is, to perform attacks *or* illuminations only. This is modelled by $x_{ij}^r = 0$ for all $(i, j) \in I_g,\ g \in G$, and $x_{ij}^r = 0$ for all $(i, j) \in A_g,\ g \in G$, respectively. Both requirements, working pairwise and dedicated roles, can be imposed simultaneously.

The model of MAMPP, with the abovementioned extensions, was implemented in AMPL [3] and solved with CPLEX [4], version 12.3 for 64 bit environment. The tests were performed on a HP DL160 server with two 6-core Intel Xeon CPUs and 72 GB of RAM memory, running Linux.

We consider a benchmark of varying problem scenarios, ranging from four targets and two aircraft up to six targets and eight aircraft, which are solved under various assumptions. Throughout all scenarios, we use six sectors with three attack positions and two illumination positions in each. Further, $q_m = 1$ holds for all targets. For all problem instances, the aircraft fleet is assigned to work in pairs.

To describe precedence relations for targets, we introduce the notation `{1234}` for no precedences at all among four targets, while for example `{12|34}` means that targets 1 and 2 must be attacked before targets 3 and 4. Similarly, `{1|2|3|4}` means that a totally ordered attack sequence is set in advance.

Table 1: Results for the benchmark instances. Columns $|M|$ and $|R|$ state the number of targets and aircraft, respectively. *Sequence* defines precedence relations and Γ is the maximal number of attacks per aircraft. The column *DR* states whether dedicated roles are used (x) or not (-). The best found solutions from CPLEX within the given time limits are given in columns *15 min* and *3 hours*, respectively, where an * indicates proven optimality. Columns *SA, BTS* and *ETS* give the median objective values reached within 15 minutes, taken over seven runs. The last column compares ETS to CPLEX 3 hours, over all runs.

	PROBLEM					CPLEX		HEURISTICS			ETS
No.	$\|M\|$	$\|R\|$	Sequence	Γ	DR	15 min	3 hours	SA	BTS	ETS	+ = −
1 (5)	4	2	{1234}	3	-	1.176	+0.187	+0.187	+0.187	+0.187	- 7 -
2 (6)				2	-	0.915	± 0 *	± 0	± 0	± 0	- 7 -
3 (7)			{12|34}	3	-	1.363	± 0 *	± 0	± 0	± 0	- 7 -
4 (8)				2	-	0.670	+0.245 *	+0.245	+0.245	+0.245	- 7 -
5 (9)	4	4	{1234}	3	-	8.411	± 0 *	± 0	± 0	± 0	- 7 -
6 (10)			{12|34}	3	-	8.411	± 0 *	± 0	± 0	± 0	- 7 -
7 (11)			{1|2|3|4}	3	-	7.620	± 0 *	± 0	± 0	± 0	- 7 -
8 (15)	5	2	{12345}	3	-	4.363	+0.751	+0.215	-0.492	+0.751	- 7 -
9 (16)			{125|34}	3	-	1.357	± 0	± 0	± 0	± 0	- 7 -
10 (17)			{1|2|3|4|5}	3	-	2.254	± 0 *	± 0	± 0	± 0	- 7 -
11 (–)	5	4	{12345}	2	-	12.419	± 0	**+0.213**	-0.274	**+0.237**	7 - -
12 (18)				3	x	12.666	± 0	± 0	± 0	± 0	- 7 -
13 (–)			{125|34}	2	-	10.974	+1.682	+1.682	+1.682	+1.682	- 7 -
14 (19)				3	x	12.666	± 0 *	± 0	± 0	± 0	- 7 -
15 (20)			{1|2|3|4|5}	2	-	11.837	± 0 *	± 0	± 0	± 0	- 7 -
16 (21)				3	x	11.838	± 0 *	± 0	± 0	± 0	- 7 -
17 (–)	5	6	{12345}	2	-	16.390	± 0	-0.050	-0.050	± 0	- 7 -
18 (22)				2	x	16.390	± 0 *	± 0	± 0	± 0	- 7 -
19 (23)			{125|34}	2	-	16.390	± 0 *	± 0	± 0	± 0	- 7 -
20 (24)				2	x	16.390	± 0 *	± 0	± 0	± 0	- 7 -
21 (25)			{1|2|3|4|5}	2	-	16.150	± 0 *	± 0	-0.351	± 0	- 7 -
22 (–)	6	2	{2|3|1|4|5|6}	3	-	3.031	± 0	± 0	± 0	± 0	- 7 -
23 (–)	6	4	{123456}	3	-	10.918	± 0	-1.706	-0.277	± 0	- 7 -
24 (–)			{123|456}	3	-	10.918	± 0	-1.489	-1.351	± 0	- 6 1
25 (–)			{12|34|56}	3	-	8.392	+1.037	+0.615	+0.485	+1.037	- 5 2
26 (–)			{2|3|1|4|5|6}	3	-	10.918	± 0	-0.277	-0.789	± 0	- 5 2
27 (26)				3	x	9.577	± 0 *	± 0	-0.111	± 0	- 7 -
28 (–)	6	6	{123456}	3	-	13.603	+0.080	-2.676	± 0	**+0.186**	7 - -
29 (–)			{123|456}	3	-	13.785	± 0	-2.730	-0.331	± 0	- 5 2
30 (–)			{12|34|56}	3	-	13.622	+0.002	-0.769	-0.098	+0.002	- 4 3
31 (–)			{2|3|1|4|5|6}	3	-	13.785	± 0	-0.400	-0.129	± 0	- 6 1
32 (27)				3	x	13.785	± 0	± 0	-0.100	± 0	- 7 -
33 (–)	6	8	{2|3|1|4|5|6}	3	-	15.756	± 0	± 0	-0.225	± 0	- 7 -
34 (28)				3	x	15.685	± 0 *	± 0	-0.150	± 0	- 7 -

Problem characteristics and results are shown in Table 1, for the choice $\mu = 0.05$ in the objective (which means high priority to expected effects on targets). The problem numbers within brackets are the corresponding problem instances in [5]. We report the best found objective values found by CPLEX after 15 minutes and three hours, respectively, which are used as reference values when evaluating our metaheuristics. Proven optimality for an instance is indicated by an * in the table.

The column for the three hour runs of CPLEX gives the resulting objective values, compared to those obtained by CPLEX in 15 minutes. (For example, for problem instance no. 1, the value +0.187 means that the objective value $1.176 + 0.187$ is obtained.) We set a time limit of 15 minutes for each metaheuristic and run each of them seven times, since all of them contain random components. The values given in the columns for the three metaheuristics are the median objective values, again compared to the objective value obtained by CPLEX in 15 minutes. The time limit of 15 minutes is motivated by what we consider to be a reasonable waiting time when using a solver interactively in a decision support tool for MAMPP.

Our analysis of the results shows that ETS is the superior metaheuristic, and hence we give some additional results for ETS in the last column. The figures in the triplet state the number of runs of ETS that produce an objective value that is better than, equal to, and worse than, respectively, the objective value found by CPLEX in three hours. (Again, for problem instance no. 1, the triplet "- 7 -" means that all seven runs of ETS gave the same objective value as CPLEX.) Table 2 contains the corresponding figures for all three metaheuristics, but here aggregated over all instances and compared also to the results obtained by CPLEX in 15 minutes.

Table 2: A comparison of all three metaheuristics against CPLEX, for all problem instances: number of objective values that are better, equal, and worse.

$+ = -$	SA			BTS			ETS		
15 minutes	38	151	56	32	121	92	53	183	9
3 hours	6	275	64	5	138	102	14	220	11

The overall conclusion of the numerical experiments is that the performance of all metaheuristics is fully satisfactory, and that the expanded tabu search is clearly superior to the other two metaheuristics. We note that each of the metaheuristics sometimes produces a solution that is better than what CPLEX does in 15 minutes, while it also sometimes produces a worse solution. Typically, though, each of the metaheuristics produces solutions that are equally good or better. In particular, ETS almost always finds solutions that are at least as good as CPLEX does, and, moreover, for every instance the median result is at least as good. When compared to a three-hour run of CPLEX, ETS in only 15 minutes finds equally good or better solutions in 96.0% of all runs on all instances.

5 Concluding remarks

We revisited the military mission planning problem introduced in [5], which involves the routing of aircraft that shall perform attacks against several ground targets. The mathematical model is recognized as a generalized vehicle routing problem with side constraints. This problem involves three types of decisions: choice of attack directions, assignment of aircraft against targets, and scheduling of each aircraft's assigned tasks. It is computationally challenging even for a few targets and aircraft.

Rather than developing a combinatorial construction and improvement heuristic, which would be quite natural for a vehicle routing type problem, we explored the option to exploit a standard solver as a key component in tailored metaheuristics. An analysis of the structure of the decisions revealed an advantageous partitioning of the problem into two stages; these involve selection of target attack directions, and aircraft assignment and scheduling, respectively. This partitioning is the basis for the three proposed metaheuristics for the problem.

They are all able to find solutions in 15 minutes that are equally good, and sometimes even better, than CPLEX finds within three hours. A deeper analysis of the progress of the metaheuristics shows that they all frequently find good or near-optimal solutions within a few minutes. Out of the three methods, the expanded tabu search is by far the most promising. The ability to find high-quality solutions in a short time is significant since the ultimate goal of this research is to enable the development of a decision support system that can be used in an interactive manner.

An alternative to our metaheuristics would be to apply CPLEX to the full model of MAMPP and branch primarily on the variables z_g, by invoking this option in CPLEX. We have indeed tried this strategy, with disappointing outcome. It is of course possible to design metaheuristics by exploiting other partitionings of the decisions than the one used in this paper. Preliminary assessements of other partitionings indicate however that these are not as efficient, since they give rise to larger and more computationally demanding inner stage problems. Nevertheless, the other partitionings can perhaps be useful in a variable neighbourhood search context, and this is an area for future research.

Since the proposed metaheuristics retain feasibility, they can be terminated at any time with feasible solutions, and they further allow us to gather a population of feasible solutions. The latter is a valuable feature in a decision support tool, since it might be of interest to generate a Pareto-front that describes the trade-off between expected effect and mission time, and also because the mission planning situation can include considerations that can not be taken into account in the model.

Our ongoing work includes a study of a column generation approach where a column represents a route for an individual aircraft.

6 Acknowledgements

The authors thank the editor of the journal and the anonymous reviewers for their helpful comments and suggestions.

The final publication is available at http://link.springer.com/article/10.1007/s11590-014-0831-x

References

[1] Bartholomew-Biggs, M.C., Parkhurst, S.C., Wilson, S.P.: Using DIRECT to Solve an Aircraft Routing Problem. Computational Optimization and Applications 21, 311–323 (2002)

[2] Eun, Y., Bang, H.: Cooperative Task Assignment/Path Planning of Multiple Unmanned Aerial Vehicles Using Generic Algorithms. Journal of Aircraft 46, 338–343 (2009)

[3] Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL – A modeling language for mathematical programming. Duxbury Press (2003)

[4] ILOG CPLEX: High-performance software for mathematical programming and optimization. ILOG, Inc. (2012)

[5] Quttineh, N-H., Larsson, T., Lundberg, K., Holmberg, K.: Military aircraft mission planning: a generalized vehicle routing model with synchronization and precedence. EURO Journal of Transportation Logistics 2, 109–127 (2013)

[6] Ralph, J.F., Jones, D.M.: Automatic Task Assignment for Mixed Aircraft Formations. In: Henry, D.J. (ed.) Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications VI, Proceedings of SPIE, Vol. 7307, 73070N:1–9 (2009)

[7] Royset, J.O., Carlyle, W.M., Wood, R.K.: Routing military aircraft with a constrained shortest-path algorithm. Military Operations Research 3, 31–52 (2009)

[8] Schumacher, C., Chandler, P.R., Pachter, M., Pachter, L.S.: Optimization of air vehicles operations using mixed-integer linear programming. Journal of Operational Research Society 58, 516–527 (2007)

[9] Shetty, V.K., Sudit, M., Nagi, R.: Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. Computers & Operations Research 35, 1813–1828 (2008)

[10] Talbi, E.-G.: Metaheuristics: from design to implementation. Wiley, Hoboken, NJ (2009)

[11] Zabarankin, M., Uryasev, S., Murphey, R.: Aircraft Routing under the Risk of Detection. Naval Research Logistics 53, 728–747 (2006)