Computing Lower Bounds for the Schedule of a Multifunction Radar

Emilie Winter
Thales Systèmes Aéroportés, 2, av Gay-Lussac, 78851 Elancourt, France, emilie.winter@fr.thalesgroup.com

Ruslan Sadykov
Lix, Cnrs, École Polytechnique, 91128 Palaiseau, France, sadykov@lix.polytechnique.fr

Among several other tasks, the radar of a fighter has to search, track and identify potential targets. The waveforms used by the radar for each of these tasks are most often incompatible and hence, cannot be processed simultaneously. Moreover, these tasks are repeated several times in a cyclic fashion. Altogether, this defines a complex scheduling problem that impacts a lot on the quality of the radar's output. In [10], a formal framework has been defined for this real time scheduling problem and we have introduced several techniques to compute efficient schedules for the radar. In order to evalutate the quality of these schedules, we present in this paper, two algorithms that compute high lower bounds for the problem. The first one is based upon a column generation scheme, the other one consists in a relaxation of the time-indexed MIP formulation of the problem.

Keywords: Radar Scheduling, Column generation, Linear Programming.

1 Introduction

A radar is a system using radiowaves to detect the presence of objects in a given volume of space. It can also compute the range (distance) as well as the relative radial velocity of these objects. Airborne radars consist of a transmitter, a single antenna and a receiver. The transmitter generates radiowaves which are sent out in a narrow beam by the antenna in a specific direction. Objects located in the beam intercept this signal and scatter the energy in all directions. A portion of this energy is scattered back to the receiver of the radar listening to all potential echoes. See [8] and [9] for a detailed description of airborne radars.

Nowadays, most of the airborne radars have a mechanically steered antenna. With such antennas, the beam is perpendicular to the antenna which is pivoting so as to direct the beam. This paper is focused on recent radars with an Electronically Steered Antenna (ESA). An ESA is a planar array antenna made of many individual radiating elements. Unlike a mechanically steered antenna, an ESA lies in a fix position on the aircraft. The phase of the radiowaves is controlled electronically so that the radar beam lights up the desired direction.

One of the key advantages of ESA is that the beam is extremely agile. As it is not subjected to the mechanical inertia of the antenna, it can be moved instantaneously from one part of the space to another, even outside the search domain (Figure 1). Moreover, we consider that the radar can switch instantaneously to an appropriate waveform.

The following tasks have to be achieved by an airborne radar.

- **Research.** The radar is sweeping across a domain to detect the potential presence of targets inside.
- **Tracking.** The radar is closely monitoring the behavior of targets (initially detected at the Research stage).

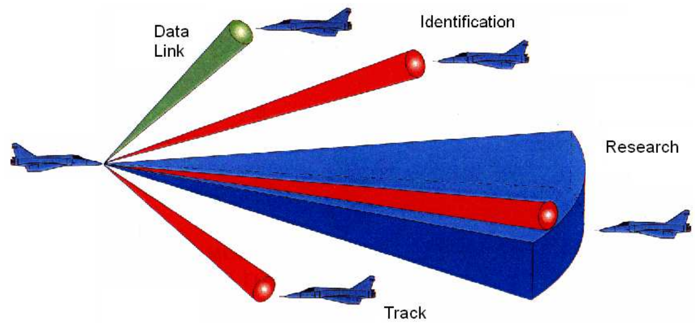

Figure 1: Functions of ESA Radars

- **Data Link.** The radar is used as a communication tool with other platforms.
- **Calibration.** The radar is performing cyclic calibration to ensure a high reliability level.

Research, Tracking, Data Link and Calibration require incompatible waveforms. So all corresponding tasks have to be *scheduled to ensure they do not overlap in time*. Moreover, Research, Tracking, Data Link and Calibration tasks have to be repeated in a more or less regular fashion. The execution of any single radar operation is called a *dwell*. Depending on the task, the periodicity constraint between dwells of a same type can be extremely important or not. For instance, as it is absolutely forbidden to loose a tracked target, tracking dwells have to be repeated with high regularity. Likewise, data link dwells are played at regular intervals, but there, the regularity constraint is much higher. Finally, to ensure a surveillance of great quality, in any circumstances, the radar has to play at least a minimum amount of research dwells.

It results from the use of only one device for all radar functions certain limitations. For instance, as different dwells cannot be played at the same time, while the radar is used in tracking mode, the pilot has no knowledge of the evolution of the tactical situation, that is new targets coming and so on. Also, the more targets to track, the less time for the search. Thus, the problem is how to use the radar time ressources to keep at best the whole situation awareness. We are interested with the scheduling of the dwells to make the radar more efficient while meeting the constraints unformally described above.

Barbaresco [1] describes a strongly related framework: Tasks are scheduled on a frame duration and are executed while the next schedule is computed. To schedule the tasks, Barbaresco associates deadlines to tasks and uses a heuristic called EDF (Earliest Deadline First). If one of the most important tasks is completed after its deadline then some tasks are removed of the system to reduce the load and the scheduling procedure is run again. The empirical study led in [1] shows that simple scheduling heuristics often improve the behavior of the radar.

The scheduling of the radar tasks can also be seen as a timing problem in which the tasks are scheduled to minimize an Earliness-Tardiness criterion [6].

Another scheduling problem for radars is the problem of interleaving tasks corresponding to receiving and sending data [7]. We study a situation where we do not have interleaving and both the sending and the receiving tasks are modeled as a unique task.

We have proposed in [10] a formal model based on cost functions that describes the problem. This model allows one to associate a cost, *i.e.* a score, to the schedules and it has been used to compare several scheduling heuristics. It has also been used to compute lower bounds to estimate the quality of our solutions. This paper focuses on the search of another lower bound by column

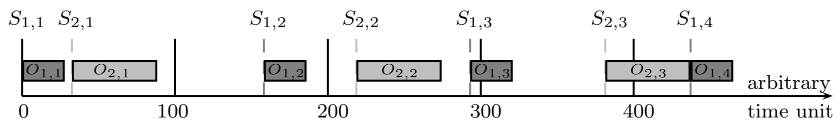

Figure 2: Scheduling Constraints

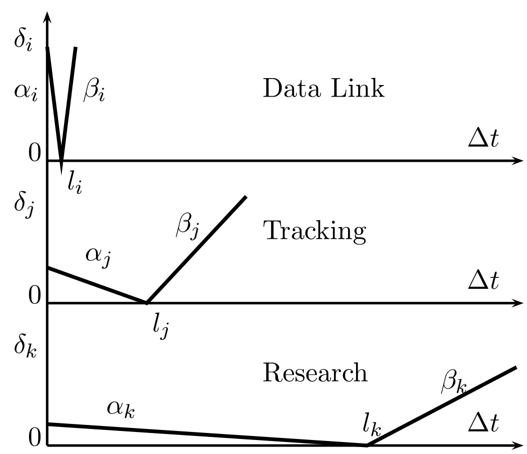

Figure 3: Typical Cost Functions for different jobs

generation. First, we re-introduce the model used in [10], adding some new notations. Then, we present an ILP formulation for the problem and we explain how to solve it by a column generation method. Further, we recall how we have computed time-indexed lower bounds in [10] and finally, we compare our experimental results to the column generation method.

2 Notations

We define a "*job*" as a single radar function that must be repeated in a more or less strict manner. A job is either the track of a given target or a data link or the "research". As several dwells are required to "execute" most jobs, a job consists of a set of non-preemptive "operations" which must obey a kind of periodicity rule. As shown on Figure 2, jobs are not necessarily executed with strict periodicity. The temporal gap between the two first operations of job 1 is greater than the gap between the operations 2 and 3. This schedule is feasible if performance constraints are ensured. The radar is seen as a single machine on which jobs have to be processed.

To model the problem, we associate cost functions δ_i to the distance between starting times of consecutive operations of the same job i. They are V-Shaped, *i.e.* $\delta_i(x) = \max(\alpha_i(l_i - x), \beta_i(x - l_i))$ where $l_i, \alpha_i \geq 0$ and $\beta_i \geq 0$ are respectively the ideal distance between two starting times and the penalty weights associated to a smaller (resp. larger) inter-distance. It derives from the penalty weights that some operations have greater priorities than others at a one time.

For each kind of job i, we have defined, in collaboration with radar engineers, cost functions features that fit real life scenario. An example of what cost functions can be is described Figure 3.

We are now ready to formally define the problem. Given n jobs 1, 2,..., n. Each job i is made of $n(i)$ consecutive and identical operations $O_{i1}, ..., O_{in(i)}$ with integer processing time p_i. For each job i, we have a penalty function δ_i associated to the distance between starting times of consecutive operations. We are also given an integer $H \geq \sum_i n(i)p_i$ that represents the horizon of the schedule (*i.e.*, all operations have to be processed between 0 and H).

A set of starting times S_{ij} defines a feasible schedule if and only if (1) operations start after or at 0 and are not completed later than H and (2) operations do not overlap, *i.e.*, for any pair of operations $O_{ij}, O_{i'j'}$, $S_{ij}+p_i \leq S_{i'j'}$ or $S_{i'j'}+p_{i'} \leq S_{ij}$. The cost associated to the schedule is exactly the sum over all jobs i of $\sum_{u=1}^{n(i)-1} \delta_i(S_{iu+1} - S_{iu})$. The scheduling heuristics in [10] aim to compute the optimal schedule for the radar, that is the schedule that minimizes $\sum_{u=1}^{n(i)-1} \delta_i(S_{iu+1} - S_{iu})$.

Note that, in practice, the problem is solved continuously and the schedule followed in the past interacts with the schedule under construction. Stated another way, many jobs have been started a long time ago. This feature does not change the combinatorial structure of the problem and to keep things simple, we omit all details about it.

In order to compute lower bounds by column generation, we use some additional notations :

For each job i, we enumerate all the possible schedules. Sch_{ik} is the k^{th} possible schedule for job i, that is for i and k given, the starting times of all the operations of job i are known. The number of possible schedules for job i is denoted s_i and $s_i = \frac{Hr!}{n(i)!(Hr-n(i))!}$, where $Hr = H - p_i + 1$. The cost to play Sch_{ik} is $cost_{ik}$. $cost_{ik}$ can be computed thanks to the cost functions δ_i defined above. We also define a binary variable X_{ik} that equals 1 if the schedule Sch_{ik} is followed, 0 otherwise. Finally, we define the following function: $\delta(i,k,t)$ equals 1 if S_{ik} is active at time t, 0 otherwise.

3 ILP Formulation

The scheduling problem can be written as an integer linear program : ILP minimizes the summation of the costs of all the jobs in the sequence, making sure jobs do not overlap in time (1^{st} set of constraints) and that one and only one schedule is performed for each job i (2^{nd} set of constraints).

$$\text{(ILP)} : \min \sum_{i=1}^{n} \sum_{k=1}^{s_i} cost_{ik} X_{ik}$$
$$s.t. \begin{cases} \forall t, \sum_{i=1}^{n} \sum_{k=1}^{s_i} \delta(i,k,t) \cdot X_{ik} \leq 1 \\ \forall i, \sum_{k=1}^{s_i} X_{ik} = 1 \\ \forall i, \forall k, X_{ik} \in \{0,1\} \end{cases}$$

To compute lower bounds by column generation, we consider the relaxation $(ILPR)$ of (ILP). For convenience matters, equation (2) has been changed in an inequality *greater or equal than*, which is equivalent, as it is a minimization problem.

$$\text{(ILPR)} : \min \sum_{i=1}^{n} \sum_{k=1}^{s_i} cost_{ik} X_{ik}$$
$$s.t. \begin{cases} \forall t, -\sum_{i=1}^{n} \sum_{k=1}^{s_i} \delta(i,k,t) \cdot X_{ik} \geq -1 \\ \forall i, \sum_{k=1}^{s_i} X_{ik} \geq 1 \\ \forall i, \forall k, 0 \leq X_{ik} \leq 1 \end{cases}$$

To this point, we have stated the problem as a linear program with a great number of variables, $\sum_i^n s_i$. As the problem has many variables (columns) but relatively few constraints ($H+n$), column generation may be beneficial. The idea is that the linear program is too large to consider all the variables explicitly. Since most of the variables will be zero in the optimal solution, only a subset of variables needs to be considered when solving the problem. Column generation allows to generate only the variables that are required to decrease the objective function. *i.e.* to find variables with negative reduced cost. (see [4] for an introduction to column generation).

In order to characterize the optimal solution, we write the dual problem which is used to identify a new variable. This problem is known as the Pricing Problem and it is developed in the next section.

4 The Pricing Problem

Let Z_t and Y_i be respectively the dual constraints associated to the first and the second set of constraints of ILPR. The dual of ILPR is :

$$(DUAL)\ :\qquad \max \sum_{i=1}^{n} Y_i - \sum_{t=0}^{H-1} Z_t$$

$$s.t. \begin{cases} Y_i - \sum_{t=0}^{H-1} \delta(i,k,t)\cdot Z_t \le cost_{ik} & 1 \le i \le n,\ 1 \le k \le ns_i \\ Z_t \ge 0 & 0 \le t \le H-1 \\ Y_i \ge 0 & 1 \le i \le n \end{cases}$$

In order to be practical, we must be able to solve the pricing problem. We obtain dual prices for each of the constraints of the relaxed problem ILPR. Given that set of dual values, if a constraint of DUAL is violated, then a variable with negative reduced cost has been identified in ILPR. This variable is added to ILPR. ILPR is solved again to generate a new set of dual values. The process is repeated until no negative reduced cost variables are identified. When all the constraints of DUAL are satisfied, the solution of ILPR is optimal.

Thus, we have some given (fixed) values for Y_i and Z_t and we are looking for a constraint

$$Y_i - \sum_{t=0}^{H-1} \delta(i,k,t)\cdot Z_t \le cost_{ik}$$

that could be violated. Stated in scheduling terms, this means that, for all i, we are looking for a schedule of job i (*i.e.*, an index $k \le s_i$) that minimizes

$$cost_{ik} + \sum_{t=0}^{H-1} \delta(i,k,t)\cdot Z_t. \tag{1}$$

We build such a schedule by dynamic programing. Let j be the number of operations of job i that still have to be scheduled. Let $P_i(j,t)$ denote the minimal cost of the schedule of the last $n(i)-j+1$ operations of job i assuming that the last $(n(i)-j+1)^{th}$ operation is completed at time t.

It is obvious that the first operation of job i to be scheduled has the minimum value of $\min_{0\le t\le H} P_i(1,t)$. If the j^{th} operation is assumed to complete at time t, then operation $j+1$ must complete at some time t' that is more or equal to $t+p_i$. Let us also define $Z(i,t) = \sum_{\theta=t}^{t+p_i-1} Z_\theta$ and $F_i(t'-t)$ the cost introduced by the gap between two consecutive operations of job i. Now, we have the following recursion that defines the minimum cost of the schedule such that the completion time of operation j

$$\begin{aligned} P_i(n(i),t) &= Z(i,t), \\ P_i(j,t) &= Z(i,t) + \min_{t+p_i \le t' \le H} \left\{ F_i(t'-t) + P_i(j+1,t') \right\}, \quad \forall j < n(i). \end{aligned}$$

Using the recursion above, this value can be found in $O(n(i) \cdot H^2)$ time, which could be improved. But there is a way to solve the pricing problem faster using the structure of the objective function F_i.

Consider again job i. Note that values $P_i(n(i),t)$ for all the time moments t can be computed in $O(H)$ time. Suppose now that we know the values $P_i(j+1,t)$, $j < n(i)$, $t \le H$. Then, we compute the following values for all time moments t.

$$\begin{aligned} a_i(j,t) &= \operatorname*{argmin}_{0 \le t' \le t} \left\{ \alpha_i(t-t') + P_i(j+1,t') \right\}, \\ b_i(j,t) &= \operatorname*{argmin}_{t \le t' \le H} \left\{ \beta_i(t'-t) + P_i(j+1,t') \right\}. \end{aligned}$$

This can be done in $O(H)$ time using the following recursion, $a_i(j,0) = 0$, $b_i(j,H) = H$, and

$$\begin{aligned} a_i(j,t) &= \begin{cases} t, & P_i(j+1,t) < P(j+1,a_i(j,t-1)) + \alpha_i \cdot (t - a_i(j,t-1)), \\ a_i(j,t-1), & \text{otherwise,} \end{cases} \\ b_i(j,t) &= \begin{cases} t, & P_i(j+1,t) < P(j+1,b_i(j,t+1)) + \beta_i \cdot (b_i(j,t+1) - t), \\ b_i(j,t+1), & \text{otherwise.} \end{cases} \end{aligned}$$

Let d'_i equal $\max\{d_i, p_i\}$. Using the values $a_i(j,t)$ and $b_i(j,t)$, the main recursion can be rewritten in the following way.

$$P_i(j,t) = Z(i,t) + \min\left\{ A, F_i(b_i(j,t+d'_i) - t) + P_i(j+1, b_i(j,t+d'_i)) \right\}, \tag{2}$$

$$\text{where} \quad A = \begin{cases} +\infty, & p_i \ge d_i, \\ F_i(a_i(j,t+d_i) - t) + P_i(j+1, a_i(j,t+d_i)), & p_i < d_i \text{ and } a_i(j,t+d_i) \ge t + p_i, \\ \min\limits_{t+p_i \le t' < t+d_i} \left\{ F_i(t'-t) + P_i(j+1,t') \right\}, & p_i < d_i \text{ and } a_i(j,t+d_i) < t + p_i. \end{cases}$$

Then, the theoretical complexity of the pricing problem for job i is reduced to $O\Big(n(i) \cdot H \cdot \max\{1, d_i - p_i\}\Big)$. Moreover, in practice, the resolution of the pricing problem for all the jobs is computable in $O(nH)$ time if one uses the updated recursion (2). Experiments have shown that the time needed to solve the pricing problem is negligible in comparison with the overall running time of the column generation algorithm.

For each job i, we check whether the minimum value of (1) is less than Y_i. If it is, we add to ILPR the column which corresponds to the schedule which minimises (1). If $\min_{k \le s_i}\{(1)\} \ge Y_i$ for all i, an optimal solution of ILPR is found. The value of this solution gives us a lower bound on the optimal solution of the radar problem.

5 Computing Lower Bounds by relaxing a Time-Indexed MIP

Let X_{ijt} be a binary variable which equals 1 if O_{ij} is played at time t, 0 otherwise. The optimal total minimum cost of the schedule can be computed by the Mixed Integer Linear Program below.

The first two constraints are due to the cost functions, The following one links the starting times to the X_{ijt} variables. The three next ones ensure respectively that each operation is scheduled at some time point, that precedence constraints between operations are met and finally that over all operations, only one can start at a time.

$$\min \sum_{i=1}^{n} \sum_{j=2}^{n(i)} W_{ij}$$
$$\begin{cases} W_{ij} \geq \alpha_i(l_i - S_{ij} + S_{ij-1}) & 1 \leq i \leq n, 2 \leq j \leq n(i) \\ W_{ij} \geq \beta_i(S_{ij} - S_{ij-1} - l_i) & 1 \leq i \leq n, 2 \leq j \leq n(i) \\ S_{ij} = \sum_{t=0}^{H-1} tX_{ijt} & 1 \leq i \leq n, 1 \leq j \leq n(i) \\ \sum_{t=0}^{H} X_{ijt} = 1 & 1 \leq i \leq n, 1 \leq j \leq n(i) \\ S_{ij} + p_i \leq S_{ij+1} & 1 \leq i \leq n, 1 \leq j \leq n(i) \\ \sum_{i=1}^{n} \sum_{j=2}^{n(i)} \sum_{t'=t-p_i+1}^{t} X_{ijt'} \leq 1 & 1 \leq i \leq n, 1 \leq j \leq n(i), 0 \leq t \leq H-1 \\ 0 \leq S_{ij} \leq H - p_i & 1 \leq i \leq n, 1 \leq j \leq n(i) \\ 0 \leq W_{ij} & 1 \leq i \leq n, 2 \leq j \leq n(i) \\ X_{ijt} \in \{0,1\},\ integer & 1 \leq i \leq n,\ 1 \leq j \leq n_i, 0 \leq t \leq H-1 \\ S_{i0} < 0,\ \text{known} & 1 \leq i \leq n \end{cases}$$

The program gives the optimal solution but runs with prohibitive computation time Still, this program allows us to compute lower bounds by relaxing the integrity of the binary variable X_{ijt}, that is making the variables X_{ijt} real.

6 Experimental Results

Together with Thales engineering department, we have generated 25 instances that represent real life situations. The total number of operations varies from 44 to 96, the load of the radar varies from 45% to 100 % and the slopes of the cost functions α and β have been chosen to model different kind of situations. The local search and time-indexed experiments have run on DELL Latitude D600 laptop running Linux with the GNU Linear Programming Kit [2] as a LP solver while the column generation method has been run with Cplex [3]. For the local search, the run time has been limited to 1.5 second.

In any case, the column generation method has provided an upper lower bound than the time-indexed method. Let us consider the twelve instances up to 83 operations. We have obtained the same results with the local search and the column generation (respectively time-indexed) lower bound on 9 instances out of 12 (respectively, 2 out of 12). Thus, for those instances we know we have reached the optimal solution both with the scheduling heuristic and the lower bound. By looking at the thirteen instances with more than 91 operations, we have noticed that the local search doesn't run long enough to obtain as good results.

For a given instance, we denote by TI the value of the time-indexed lower bound and CG the column generation one. We have computed (TI/CG)*100 and (CG/LS)*100 for all the instances. The mean of the results in percentage appear in the comparative statement below.

Number of operations per instance	Number of instances	Mean of (TI/CG)*100 in %	Mean of (CG/LS)*100 in %
≤ 83	12	0.960	0.795
≥ 91	13	0.858	0.206

7 Conclusion

In this paper, we have proposed two methods to compute lower bounds for the radar scheduling problem. These approaches have been tested on a set of instances close from real life scenariosẆe have shown experimentally that the column generation algorithm provides stronger lower bounds than the linear programming relaxation of the time-indexed formulation of the problem. The lower bounds obtained have allowed us to evaluate the quality of the local search algorithm presented in [10] to schedule the tasks of the radar. It has been shown that it performs very well when the convergence time choosen for the algorithm fits the number of operations of the test instances. Thus, for a half of these instances, the optimality of the solutions found has been proved. However, for the test instances with more than 90 operations, the gap between lower and upper bounds known is still high. In the future, we hope to reduce this gap both by improving the local search algorithm and proposing methods to tighten lower bounds.

References

[1] F. Barbaresco (2003), Approche Cognitive de la Gestion Radar, *Proceedings of the conference Cogis 2003.*

[2] GNU Linear Programming Kit home page. http://www.gnu.org/software/glpk /glpk.html

[3] Cplex home page. http://www.ilog.com/products/cplex/

[4] G.B. Dantzig, P. Wolfe (1960), Decomposition principle for linear programs, *Oper. Res.* **8**, 101 – 111.

[5] M. Elshafei, H. D. Sherali, J. C. Smith (2003), Radar pulse interleaving for multi-target tracking, *Naval Research Logistics* **51**(1), 72 – 94.

[6] Y. Hendel (2005), Contributions à l'ordonnancement juste à temps, rapport de thèse, université Pierre et Marie Curie, laboratoire Lip6/SysDef.

[7] A. J. Orman, C. N. Potts (1997), On the Complexity of Coupled-task Scheduling, *Discrete Applied Mathematics* **72**(1-2), 141 – 154.

[8] M. I. Skolnik (1979), Introduction to Radar Systems, 2^{nd} edition, *McGraw-Hill Kogakusha.*

[9] G. W. Stimson (1998), *Introduction to Airborne Radar*, 2^{nd} Edition. SciTech Publishing, Inc.

[10] E. Winter, Ph. Baptiste (2007), On Scheduling a Multifunction Radar , accepted for publication in *Aerospace Science and Technologies.*