Timing problem for scheduling an airborne radar

Yann Hendel and Ruslan Sadykov
Ecole Polytechnique, LIX CNRS, F-91128 Palaiseau, France
Hendel,Sadykov@lix.polytechnique.fr

1 Introduction

In this abstract, we consider a generalization of the one-machine earliness-tardiness scheduling problem (OMETSP). In the OMETSP settings, the due dates are constants given in the input, whereas in this generalization, the due dates depend on the starting time of another job (called *reference job*). Our interest in this problem comes from the study of airborne radars: the airborne radar scheduling problem (ARSP) has been proposed by Winter and Baptiste [3] and we refer to their paper for the modelization of scheduling tasks on an airborne radar. Clearly, ARSP is NP-hard as a generalization of OMETSP [1]. In the literature, OMETSP is often divided into two subproblems: the *sequencing* problem consists of finding an execution order for the jobs, and the *timing* problem consists of obtaining an optimal schedule for a given execution order of the jobs. This latter problem is polynomial. However, it is important to get an efficient timing algorithm, since most metaheuristics in the scheduling field rely on neighborhoods based on permutation of jobs in a given sequence and hence, the timing algorithm must be called very often. The same approach can be applied to ARSP. In this abstract, we focus on the timing problem of ARSP.

We now provide a formal definition of the problem: we are given a sequence $(0, 1, \ldots, n)$ of jobs which have to be processed on a single machine. In the following, N denotes the set $\{1, \ldots, n\}$. Each job $j \in N$ has a reference job $b(j) < j$, a processing time $p_j > 0$, a due date d_j, an earliness penalty per unit time α_j and a tardiness penalty per unit time β_j. Moreover, preemption is not allowed. S_j denotes the starting time of job j. The difference between the starting times of jobs j and $b(j)$ should ideally be equal to d_j. If this difference is less (respectively more) than d_j, we say that the job is early (respectively tardy) and its earliness (respectively its tardiness) is denoted by $E_j = \max\{0, d_j - x\}$ (respectively $T_j = \max\{0, x - d_j\}$) with $x = S_j - S_{b(j)}$. Our aim is to find a schedule $S = \{S_j\}_{j \in N \cup \{0\}}$ such that $0 \leq S_{j-1} \leq S_j - p_j$, $j \in N$, which minimizes:

$$F = \sum_{j \in N} \alpha_j E_j + \beta_j T_j$$

In Section 2, we propose two linear programs to represent the problem, then in Section 3, we propose a combinatorial algorithm based on the dual formulation of the second linear program. In Section 4, we provide some computational results.

2 Primal Linear Program

First, we consider an equivalent problem where the processing times of the jobs can be reduced to zero (see Pan and Shi [2]): instead of considering the starting time of the jobs, we consider the total idle time before the starting of each job. However, the due dates should be modified:

$$d'_j = d_j - \sum_{i=b(j)}^{j-1} p_j,$$

and any schedule S such that $0 \leq S_{j-1} \leq S_j$, $j \in N$, becomes feasible.

We now provide a first Linear Programming formulation of the problem: M_j represents the idle time between jobs $j-1$ and j. Observe that this idle time impacts the cost of all the jobs $k \geq j$ such that $b(k) < j$.

$$\text{(PR1)} \quad \begin{aligned} \min \quad & \sum_{j \in N} \alpha_j E_j + \beta_j T_j && (1) \\ s.t. \quad & E_j + \sum_{i=b(j)+1}^{j} M_i \geq d'_j, \quad j \in N, && (2) \\ & T_j - \sum_{i=b(j)+1}^{j} M_i \geq -d'_j, \quad j \in N, && (3) \\ & E_j \geq 0, \quad T_j \geq 0, \quad M_j \geq 0 \quad j \in N. && (4) \end{aligned}$$

Equations (2) and (3) represent respectively the earliness and tardiness of the jobs. Observe that the order between the jobs is induces by the non-negativity of variables M.

The formulation (PR1) can be solved by any LP algorithm (for example, by the simplex algorithm). Our goal is to find a faster combinatorial algorithm to solve (PR1). We first propose an equivalent LP formulation:

$$\text{(PR2)} \quad \begin{aligned} \min \quad & \sum_{j \in N} (\alpha_j + \beta_j) E_j + \sum_{j \in N} \Big(\sum_{\substack{i \in [j,n]: \\ b(i) < j}} \beta_i \Big) M_j - \sum_{j \in N} d'_j \beta_j && (5) \\ s.t. \quad & E_j + \sum_{i=b(j)+1}^{j} M_i \geq d'_j, \quad j \in N, && (6) \\ & E_j \geq 0, \quad M_j \geq 0, \quad j \in N. && (7) \end{aligned}$$

Theorem 1 *(PR2) is equivalent to (PR1).*

Sketch of proof. For each job j, the objective function is to minimize $(\alpha_j + \beta_j)E_j + \beta_j(\sum_{i=b(j)+1}^{j} M_i - d'j)$. If $d'_j \geq \sum_{i=b(j)+1}^{j} M_i$, then $E_j = 0$ and the objective function for job j is $\beta_j(\sum_{i=b(j)+1}^{j} M_i - d'_j)$ which is the tardiness cost. Else, $E_j = d'_j - \sum_{i=b(j)+1}^{j} M_i$ and hence the objective function becomes $(\alpha_j + \beta_j)(d'_j - \sum_{i=b(j)+1}^{j} M_i) + \beta_j(\sum_{i=b(j)+1}^{j} M_i - d'j) = \alpha_j(d'_j - \sum_{i=b(j)+1}^{j} M_i)$ which is the earliness cost. □

3 Dual Linear Program

We define $\bar{\beta}_j = \sum_{i \in [j,n]:b(i)<j} \beta_i$. We now look at the dual linear program of (PR2):

$$\text{(DU2)} \qquad \begin{aligned} \max \quad & \sum_{j \in N} d'_j X_j - \sum_{j \in N} d'_j \beta_j && (8) \\ s.t \quad & 0 \leq X_j \leq \alpha_j + \beta_j, \quad j \in N, && (9) \\ & \sum_{\substack{j \leq i \leq n: \\ b(i)<j}} X_i \leq \bar{\beta}_j, \quad j \in N. && (10) \end{aligned}$$

Theorem 2 *(DU2) can be solved using a minimum cost flow algorithm*

Sketch of proof. The matrix of coefficients of the constraints (10) is interval. Therefore, in the same manner as it has been done in [4], (DU2) can be formulated as a minimum cost flow problem (see Figure 1). In the graph, vertex j represents job j. For each $j \in N$, there is edge $(j, j-1)$ with capacity $\bar{\beta}_j$ and cost 0, and edge $(j, b(j))$ with capacity $\alpha_j + \beta_j$ and cost $-d'_j$. The deficit of vertex j is $\bar{\beta}_{j+1} - \bar{\beta}_j = \sum_{i \in N:b(i)=j} \beta_i - \beta_j$. □

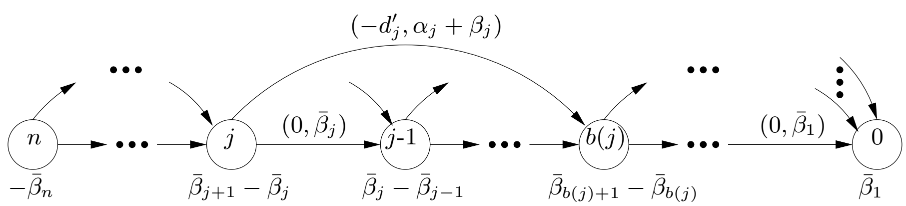

Figure 1: Minimum cost flow graph

4 Computational results

We have performed a numerical study where we compared the solution times of the Linear Programming and Minimum Cost Flow algorithms for our timing problem. In the experiment, the formulation (PR2) is solved using CPLEX 10.1 (see [5]), and for the minimum cost flow problem, the MCFZIB solver [6] was used. The solution times are shown in Table 1. As you can notice, the difference can be clearly seen from $n = 500$.

n	LP	MCF
250	0.02s	<0.01s
500	0.10s	0.01s
1000	0.44s	0.02s
2500	4.62s	0.05s

Table 1: Numerical study of the Linear Programming (LP) and Minimum Cost Flow (MCF) solvers

5 Conclusion

In this abstract, we have proposed an efficient combinatorial timing algorithm for the airborne radars scheduling problem. Subsequent studies will focus on designing metaheuristics for the general problem.

References

[1] M.R. Garey, R.E. Tarjan, G.T. Wilfong (1988), One-processor scheduling with symmetric earliness and tardiness penalties, *Mathematics of Operations Research* 13:330–348.

[2] Y. Pan, L. Shi (2005), Dual constrained single machine sequencing to minimize total weighted completion time, *IEEE Transactions on Automation Science and Engineering* 2(4):347–357.

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

[4] K. I. Bouzina, H. Emmons (1996), Interval scheduling on identical machines, *J Global Optim* 9: 379-393.

[5] ILOG Cplex 10.1, http://www.ilog.com/products/cplex/

[6] http://www.di.unipi.it/di/groups/optimize/Software/MCF.html