Compact MIP formulations for Minimizing Total Weighted Tardiness

Philippe Baptiste
Ecole Polytechnique, CNRS LIX, 91128 Palaiseau, France.
Tel: +33 6 87 02 24 23, Philippe.Baptiste@polytechnique.fr.

Sadykov Ruslan
CORE, Université Catholique de Louvain,
voie du Roman Pays, 34, 1348 Louvain-la-Neuve, Belgium
Tel: +32 10 47 94 21, fax: +32 10 47 43 01,
sadykov@core.ucl.ac.be

Abstract

We present new MIP formulations for the single machine total weighted tardiness problem. These formulations are based on a partition of the time horizon into intervals. This partition is done in such a way that the problem reduces to the problem of assigning jobs to intervals. This leads to a compact formulation that can be solved with standard integer programming techniques. We also introduce a relaxation of the problem, solved by coliumn generation, which gives tight lower bounds.

1 Introduction

A set of jobs $N = \{1, \ldots, n\}$ has to be processed on a single machine. Only one job can be processed at a time and preemptions are not allowed. All jobs are available for processing at time zero. Each job $j \in N$ has a processing time p_j, a due date d_j and a weight w_j. All data is integer. The tardiness $T_j(\pi)$ of job j in schedule π is defined as $T_j(\pi) = \max\{0, C_j(\pi) - d_j\}$. For given schedule π, job j is *on time* if $C_j(\pi) \leq d_j$, otherwise job j is *late*. The objective is to find a schedule π with minimum total weighted tardiness $\sum_{j \in N} w_j T_j(\pi)$. Using the standard scheduling notation, this problem is denoted as $1 \mid\mid \sum w_j T_j$. The problem has received a significant attention in the literature (e.g., dominance rules [3, 5, 1] and exact approach [2]).

2 Appropriate partition of the time horizon

This problem can be modelled with a large "time-indexed" MIP where, for each job j and each time point t, we have a binary variable $x_{jt} \in \{0, 1\}$ that indicates whether j is started at t or not. The formulation is the following.

$$\min \quad \sum_{t=0}^{P-1} \sum_{j \in N} w_j \cdot \max\{t + p_j - d_j, 0\} x_{jt}$$

$$s.t. \quad \sum_{t=0}^{P-p_j} x_{jt} = 1, \quad j \in N,$$

$$\sum_{j \in N} \sum_{s=\max\{0, t-p_j\}}^{t} x_{js} = 1, \quad 0 \leq t \leq P-1,$$

The main issue with this formulation is that the number of variables is pseudopolynomial, leading to huge MIPs that cannot be solved even for small sizes of instances. The main purpose of this section is to find a partition of the time horizon into a rather "small" set of intervals such that when we know which jobs are completed in the intervals we can immediately derive an optimal schedule. The major advantage of this technique is that the resulting MIP, in which time-indexed variables are replaced by interval-indexed variables, is small and can be solved efficiently.

First, note that there is an optimal schedule without idle times and hence the completion time of the last job equals $P = \sum_{j \in N} p_j$. Given a partition of $(0, P]$ into intervals $\{I^1, \ldots, I^m\}$, $\{\sigma^1, \ldots, \sigma^m\}$ is an *appropriate* set of permutations if and only if there is an optimal schedule S in which, for any two jobs i, j that are both completed in I^u, job i is sequenced before job j if and only if $\sigma^u(i) < \sigma^u(j)$.

Ex 1 Consider for instance $1 \mid\mid \sum w_j C_j$, a special case of $1 \mid\mid \sum w_j T_j$. In this case there is one single interval I^1 and the appropriate permutation σ^1 is the one corresponding to Smith order.

Ex 2 If $I^u = (u-1, u]$ $\forall u \in [1, \ldots, P]$ then any set of permutations is appropriate.

Ex 3 When all processing times are equal to p, the set of intervals $\{I^1, \ldots, I^m\}$ is exactly the set of intervals corresponding to consecutive due dates. It is then easy to see that for an interval $I^u = (d_i, d_j]$, the order corresponding to permutation σ^u can be described as follows. Put first jobs k with $d_k \leq d_i$ according to Smith rule, then put all other jobs in any order.

When processing times are arbitrary, Smith rule is not appropriate any more for late jobs. Consider an instance of the problem with 2 jobs, $p_1 = 4$, $d_1 = 9$, $w_1 = 2$, $p_2 = 10$, $d_2 = 5$, $w_2 = 3$. In the only optimal schedule S^* job 2 precedes job 1. In S^* both jobs are completed in interval $I^2 = [10, 14]$, but Smith order is $(1, 2)$. Our major result is that there are partitions (with a reasonably small number of intervals) for which we can compute appropriate permutations in polynomial time.

Proposition 1. *Let $I_u = [e_{u-1} + 1, e_u]$, $u = \{1, \ldots, m\}$ be any partition such that*

- *for all due date d_j there is an index u such that $d_j = e_u$,*
- *for any pair of jobs $(i, j) \in N$ with $p_i/w_i \leq p_j/w_j$ and any interval I^u, either $e_u \leq e_{u-1} + p_j$ or $e_{u-1} \geq d_i + \left\lceil \frac{w_j p_i}{w_i} \right\rceil - p_i$*

then an appropriate set of permutations can be computed in polynomial time.

3 MIP formulation

Consider a partition of the time horizon into intervals $\{I^1, \ldots, I^m\}$ and an appropriate set of permutations σ. Let $M = \{1, \ldots, m\}$. We denote A_j^u and B_j^u, $j \in N$, $u \in M$, the sets of jobs which come, respectively, **a**fter and **b**efore job j in permutation σ^u. We also denote τ_j^u the minimum tardiness of job j if it is assigned to interval u: $\tau_j^u = \max\{0, e_{u-1} + 1 - d_j\}$.

The assignment binary variable Z_j^u, $j \in N$, $u \in M$, takes the value 1 if job j is completed in interval u or earlier, otherwise $Z_{ju} = 0$. For each $j \in N$, we set $Z_j^0 = 0$ and $Z_j^m = 1$. The variable $\tilde{T}_j$, $j \in N$, denotes the difference between the tardiness T_j of job j and the value τ_j^u, where u is the interval to which job j is assigned. The variable C_j^u, $j \in N$, $u \in M$, denotes the completion time of job j if it is completed in interval I^u. We now give the formulation.

$$\min \quad \sum_{j \in N} w_j \cdot \left(\tilde{T}_j + \sum_{u \in M} \tau_j^u (Z_j^u - Z_j^{u-1}) \right) \tag{1}$$

$$s.t. \quad \sum_{j \in N} p_j Z_j^u \le e_u, \quad \forall u \in M, \tag{2}$$

$$Z_j^{u-1} \le Z_j^u, \quad \forall j \in N, \forall u \in M, \tag{3}$$

$$C_j^u \ge p_j Z_j^u + \sum_{i \in A_j^u} p_i Z_i^{u-1} + \sum_{i \in B_j^u} p_i Z_i^u, \quad \forall j \in N, \forall u \in M, \tag{4}$$

$$\tilde{T}_j \ge C_j^u - (e_{u-1} + 1) - (1 - Z_j^u + Z_j^{u-1}) \cdot (e_u - e_{u-1} - 1), \quad \forall j \in N, \forall u \in M, d_j \le e_{u-1}, \tag{5}$$

$$Z_j^u \in \{0, 1\}, \quad \forall j \in N, \forall u \in M, \tag{6}$$

$$C_j^u \ge 0, \quad \forall j \in N, \forall u \in M, \tag{7}$$

$$\tilde{T}_j \ge 0, \quad \forall j \in N. \tag{8}$$

The constraints (2) guarantee that schedule is feasible, i.e. the sum of the processing times of jobs assigned to the first u intervals is not more than e_u, the total length of these intervals. The inequalities (3) ensure that the values taken by Z are consistent with the meaning of these variables. The constraints (5) relate variables Z, $\tilde{T}$, and C.

Preliminary experiments show that the proposed formulation outperforms other MIP formulations appeared in the literature [4].

4 Dantzig-Wolfe reformulation

Let the set Ω^u, $u \in M$, includes all partial schedules starting not later than e_{u-1} where jobs are completed in interval I_u and processed according to permutation σ^u. Each set Ω^u, $u \in M \setminus \{m\}$, also includes empty partial schedules, i.e. the schedule containing no jobs. For each job $j \in N$, $a_{j\omega} = 1$ if partial schedule ω contains job j, otherwise $a_{j\omega} = 0$. Let c_ω, $\omega \in \Omega^u$, be the difference between e_{u-1} and the starting time of ω. Let h_ω, $\omega \in \Omega^u$, be the total cost of ω: $h_\omega = \sum_{j \in N} a_{j\omega} w_j T_j(\omega)$. The sum of processing times of all jobs included in ω we denote $\bar{p}_\omega$: $\bar{p}_\omega = \sum_{j \in N} a_{j\omega} p_j$.

The binary variable λ_ω equals 1 if the solution includes partial schedule ω, otherwise $\lambda_\omega = 0$. Using these variables, a relaxation of the problem can be formulated as the following master problem **(MP)**.

$$\min \quad \sum_{u \in M} \sum_{\omega \in \Omega^u} h_\omega \lambda_\omega \tag{9}$$

$$s.t. \quad \sum_{u \in M} \sum_{\omega \in \Omega^u} a_{j\omega} \lambda_\omega = 1, \quad \forall j \in N, \tag{10}$$

$$\sum_{\omega \in \Omega^{u+1}} c_\omega \lambda_\omega + \sum_{\omega \in \Omega^u} \bar{p}_\omega \lambda_\omega = e_u - e_{u-1} + \sum_{\omega \in \Omega^u} c_\omega \lambda_\omega, \quad 1 \leq u < m, \tag{11}$$

$$\sum_{\omega \in \Omega^u} \lambda_\omega = 1, \quad \forall u \in M \tag{12}$$

$$\lambda_\omega \geq 0, \quad \forall \omega \in \Omega^u, \forall u \in M. \tag{13}$$

The constraints (10) specify that each job is contained in exactly one partial schedule included in the solution. The constraints (11) guarantee a partial schedule starts at the time moment when the previous partial schedule is completed. Finally, the constraints (12) state that only one partial schedule should be selected for each interval. The formulation (MP) can be solved by iteratively adding columns with negative reduced cost. To find such columns, the *pricing problem* should be solved. This problem can be found in $O(nP)$ time by dynamic programming.

Preliminary experiments showed that the formulation (MP) can be solved in a reasonable time and gives tight lower bound for the problem. A comparison has been performed with a Lagrangean relaxation lower bound from [2].

References

[1] Arkturk M.S., M.B. Yildririm (1999). A new dominance rule for the total weighted tardiness problem. *Production Planning and Control*, **10**(2), 138–149.

[2] Babu P., L. Péridy, E. Pinson (2004). A branch and bound algorithm to minimize total weighted tardiness on a single processor. *Annals of Operations Research*, **129**, 33–46.

[3] Emmons H. (1969). One-machine sequencing to minimize certain functions of jobs tardiness. *Operations Research*, **17**, 701–715.

[4] Khowala K., A. Keha, J. Fowler (2005). A comparison of different formulations for the non-preemptive single machine total weighted tardiness scheduling problem. In: *The 2nd Multidisciplinary International conference on scheduling MISTA, New York, USA, July 2005.*

[5] Rinnooy Kan A., B. Lageweg, J.K. Lenstra (1975). Minimizing total cost in one-machine scheduling. *Operations Research*, **23**, 908–927.