European Journal of Operational Research 201 (2010) 45–54

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Lagrangian domain reductions for the single machine earliness–tardiness problem with release dates

Boris Detienne [a,*], Éric Pinson [b], David Rivreau [b]

[a] École Nationale Supérieure des Mines de Saint-Étienne, Centre Microélectronique de Provence Georges Charpak, 880 route de Mimet, 13541 Gardanne, France
[b] Institut de Mathématiques Appliquées, 3 place André-Leroy 49008 Angers, France

ARTICLE INFO

Article history:
Received 12 March 2008
Accepted 4 February 2009
Available online 14 February 2009

Keywords:
Scheduling
Just-in-time
Elimination rules
Lagrangian relaxation
Exact method

ABSTRACT

This paper presents new elimination rules for the single machine problem with general earliness and tardiness penalties subject to release dates. These rules, based on a Lagrangian decomposition, allow to drastically reduce the execution windows of the jobs. We measure the efficiency of these properties by integrating them in a branch-and-bound. Tests show that instances with up to 70 jobs without release dates, and up to 40 jobs with release dates, can be optimally solved within 1000 seconds.

1. Introduction

In the $1|r_j|\sum_j \alpha_j E_j + \beta_j T_j$ scheduling problem, a set J of n jobs has to be scheduled without preemption on a single processor, which is able to handle only one job at a time. Each job $j \in J$ is given an integer processing time p_j, a release date r_j, a due date d_j, and two positive penalties α_j and β_j. The *earliness* (resp. *tardiness*) of job j is defined as $E_j = \max(d_j - C_j, 0)$ (resp. $T_j = \max(C_j - d_j, 0)$), where C_j denotes the completion time of j, and a cost $\alpha_j E_j + \beta_j T_j$ is incurred if the completion time of j is different from its due date.

Reviews about this problem and variants can be found in Baker and Scudder (1990) or Kanet et al. (2000). Since its special case $1||\sum_j w_j T_j$ is NP-hard in the strong sense (Lenstra et al., 1977), this problem is clearly NP-hard. A few polynomial special cases exist, most of them involving a common due date. Hassin and Shani (2005) gather algorithms for solving some of these problems.

Kedad-Sidhoum et al. (2008) present a computational study of lower bounds for the general case, in which one can notice the excellent quality of time-indexed based bounds.

An interesting property of the problem has been exploited in many exact as well as heuristic approaches: solving the problem with respect to a fixed sequence of jobs (*timing problem*) can be done polynomially (see, e.g. Hendel and Sourd, 2007). Thus, an important part of the computational effort can be focused on searching for a good or optimal sequence.

Solving exactly $1|r_j|\sum_j \alpha_j E_j + \beta_j T_j$ appears to be difficult, even for small instances, and much work has been devoted to special cases or heuristic methods (Fry et al., 1987; Lee and Choi, 1995; Bülbül et al., 2007).

Exact approaches able to solve instances without release dates and with up to 30 jobs were proposed by Tanaka et al. (2003) and Sourd and Kedad-Sidhoum (2003). More recently, Sourd and Kedad-Sidhoum (2008) present a branch-and-bound algorithm based on a Lagrangian relaxation of resource constraints in the time-indexed formulation (Sousa, 1989) with new dominance rules that can solve most instances with 50 jobs. Yau et al. (2006) develop an hybrid Dynamic Programming – branch-and-bound method, relying on an assignment-based lower bound, allowing to solve instances with up to 50 jobs. Two very recent papers deal successfully with the presence of release dates: Sourd (in press) uses a Lagrangian relaxation of the number of occurrences in the time-indexed formulation and reinforcing valid inequalities to deal successfully with instances with up to 60 jobs. Tanaka and Fujikama (2008) develop a Successive Sublimation Dynamic Programming scheme to handle general single machine problems that can solve optimally all of the instances of the literature of $1|r_j|\sum_j \alpha_j E_j + \beta_j T_j$ with up to 200 jobs.

* Corresponding author. Tel.: +33 (0)4 42 61 66 76.
E-mail address: boris.detienne@emse.fr (B. Detienne).

0377-2217/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2009.02.005

This paper aims to present new and original elimination rules for this problem. It is organized as follows. In Section 2 we state a 0–1 time-indexed formulation of the problem, and we present a lower bound based on a Lagrangian decomposition. Next, we describe new elimination rules exploiting this decomposition (Section 3). Section 4 is devoted to a branch-and-bound exploiting these results and two Lagrangian upper bounds. Computational experiments are reported in the last section, followed by a conclusion.

2. Lower bound

2.1. Time-indexed formulation

Our elimination rules rely on the computation of a lower bound based on a Lagrangian decomposition over the following 0–1 time-indexed formulation. Let us introduce the notations used in the sequel:

- $T = \{0, \ldots, \tau\}$: scheduling horizon. We can set $\tau = \max_j \max(r_j, d_j) + \sum_j p_j$.
- $P = \sum_{j \in J} p_j$: sum of the processing times.
- $\forall j \in J, ect_j$ (resp. lct_j): earliest (resp. latest) possible completion time possible for job j.
- $\forall j \in J, D_j = \{ect_j, \ldots, lct_j\}$: completion time window for job j.
- $\forall j \in J, \forall t \in D_j, c_{jt} = \max(\alpha_j \cdot (d_j - t), \beta_j \cdot (t - d_j))$: cost incurred if job j completes at time t.
- $\forall j \in J, \forall t \in D_j, \forall \theta \in T, a_\theta^{jt} = \begin{cases} 1 & \text{if } \theta \in \{t - p_j, \ldots, t-1\} \\ 0 & \text{otherwise} \end{cases}$: indicates if job j is in processing at time θ if it completes at time t.

Our problem can be formulated as a time-indexed integer program, where a decision variable x_{jt} is equal to 1 if job j completes at time instant t, 0 otherwise; y_θ is equal to 1 if a task is processed at time instant θ

$$
\begin{aligned}
[ETTIS^D] \quad \min \quad & \sum_{j \in J} \sum_{t \in D_j} c_{jt} \cdot x_{jt} \\
\text{s.t.} \quad & \sum_{t \in D_j} x_{jt} = 1 \quad \forall j \in J, && (1) \\
& \sum_{j \in J} \sum_{t \in D_j} a_\theta^{jt} x_{jt} \leqslant y_\theta \quad \forall \theta \in T, && (2) \\
& \sum_{\theta \in T} y_\theta = P, && (3) \\
& x_{jt} \in \{0,1\} \quad \forall j \in J, \ \forall t \in D_j, && (4) \\
& y_\theta \in \{0,1\} \quad \forall \theta \in T. && (5)
\end{aligned}
$$

Constraint (1) ensures that at most one occurrence of each job is processed in any feasible schedule, while constraints (2) and (3) prevent from processing more than one job simultaneously. The use of the surrogate (3) allows the design of the elimination rule described in Proposition 5.

2.2. Lagrangian decomposition

Let $v = (v_0, \ldots, v_\tau) \geqslant 0$ denote Lagrangian multipliers associated with the $\tau + 1$ coupling constraints, we price out (2) to form the Lagrangian dual function:

$$
\begin{aligned}
L^D(v) = \min \quad & \sum_{j \in J} \sum_{t \in D_j} \left(c_{jt} + \sum_{\theta = t - p_j}^{t-1} v_\theta \right) \cdot x_{jt} - \sum_{\theta \in T} v_\theta \cdot y_\theta \\
\text{s.t.} \quad & (1) \wedge (3) \wedge (4) \wedge (5).
\end{aligned}
$$

It follows that the dual problem $[DP]$ consists in finding a vector of Lagrangian multipliers v^* maximizing $L^D(v)$:

$$[DP] = \max_{v \in \mathbb{R}^{+\tau}} L^D(v).$$

The resulting value provides us with a lower bound.

2.3. Solving the dual problem

Clearly, computing $L^D(v)$ for any vector of Lagrangian multipliers v leads to $n + 1$ independent subproblems. Indeed, one can write

$$L^D(v) = \sum_{j \in J} SP_x^{D_j}(v) - SP_y^D(v).$$

$\tilde{c}_{jt} = c_{jt} + \sum_{\theta = t - p_j}^{t-1} v_\theta$ noting the reduced cost of variable x_{jt}, the subproblem associated with each job j can be described by

$$
\begin{aligned}
SP_x^{Dj}(v) = \min \quad & \sum_{t \in D_j} \tilde{c}_{jt} \cdot x_{jt} \\
\text{s.t.} \quad & \sum_{t \in D_j} x_{jt} = 1, \\
& x_{jt} \in \{0,1\} \quad \forall t \in D_j.
\end{aligned}
$$