European Journal of Operational Research 202 (2009) 646–653

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A new dynamic programming formulation for scheduling independent tasks with common due date on parallel machines

Nguyen Huynh Tuong, Ameur Soukhal *, Jean-Charles Billaut

Laboratoire d'Informatique, Université François Rabelais, Tours 64 Avenue Jean Portalis – 37200 Tours, France

ARTICLE INFO

Article history:
Received 29 July 2008
Accepted 22 June 2009
Available online 7 July 2009

Keywords:
Scheduling
Single machine
Parallel machines
Dynamic programming

ABSTRACT

This paper deals with a scheduling problem of independent tasks with common due date where the objective is to minimize the total weighted tardiness. The problem is known to be ordinary NP-hard in the case of a single machine and a dynamic programming algorithm was presented in the seminal work of Lawler and Moore [E.L. Lawler, J.M. Moore, A functional equation and its application to resource allocation and sequencing problems, Management Science 16 (1969) 77–84]. In this paper, this algorithm is described and discussed. Then, a new dynamic programming algorithm is proposed for solving the single machine case. These methods are extended for solving the identical and uniform parallel-machine scheduling problems.

1. Introduction

Scheduling problems with the objective of minimizing the total weighted tardiness have been paid attention since more than 40 years. From a practical point of view, this objective function aims to reduce the costs caused by contract penalties, loss of reputation or client dissatisfaction. These scheduling problems can also be considered as special cases of Just-In-Time scheduling problems, when the storage cost and deterioration of goods are negligible, i.e. when there is no earliness penalties (see [2,5,6] for more details on Just-In-Time scheduling problems).

In this paper, we study a problem of scheduling a set $\mathcal{J} = \{J_1, \ldots, J_n\}$ of n jobs without preemption on a set $\mathcal{M} = \{M_1, \ldots, M_m\}$ of m ($m \geqslant 1$) parallel machines. Each job J_i is characterized by a processing time p_i, a positive integer tardiness penalty w_i, and a common due date $d_i = d,\ d \geqslant 0$. Each job is available for processing at time zero. We denote by S_i the starting time of J_i and by C_i its completion time.

As defined in [7], a job J_i can be in one of the three following states:

1. early job if $C_i < d$,
2. fully-tardy job if $S_i > d$,
3. straddling job if $S_i < d$ and $C_i \geqslant d$.

The tardiness of J_i, denoted by T_i is defined by $T_i = \max(0, C_i - d)$. So, if J_i is completed before the due date ($C_i < d$) there is no penalty. Otherwise, there is a job-dependent tardiness penalty equal to $w_i \times T_i$. The objective is to minimize the sum of tardiness penalties $\sum_{i=1}^{n} w_i T_i$. According to the standard scheduling notation, the considered problem is traditionally denoted by $Pm|d_i = d|\sum w_i T_i,\ m \geqslant 1$. This problem is NP-hard since the problem $1|d_i = d|\sum w_i T_i$ is NP-hard [17].

The general case on a single machine $1\|\sum w_i T_i$ is strongly NP-hard [11] and is considered as a classical scheduling problem [1,15]. Several dynamic programs (DP) and approximation algorithms have been proposed for some extensions of this problem. In the case of agreeable weights (i.e. $p_j < p_i \Rightarrow w_j \geqslant w_i$), Lawler gives an optimal pseudopolynomial algorithm [11]. The problem with unit weights $1\|\sum T_i$ has been proved ordinary NP-hard [4] and accepts a fully polynomial-time approximation scheme (FPTAS) [12]. For the general case $1\|\sum w_i T_i$, Cheng et al. [3] show that a $(n-1)$-approximation algorithm exists. With a fixed number of distinct due dates, Kolliopoulos and Steiner [8] propose a pseudopolynomial dynamic programming algorithm. Moreover, they explain that if the job weights are bounded by a polynomial

* Corresponding author.
E-mail addresses: nguyen.huynh@etu.univ-tours.fr (N. Huynh Tuong), ameur.soukhal@univ-tours.fr (A. Soukhal), jean-charles.billaut@univ-tours.fr (J.-C. Billaut).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2009.06.033

function in n, the problem accepts an FPTAS. For the common due date case $1|d_i = d|\sum w_iT_i$, Lawler and Moore [13] propose an $O(n^2d)$ algorithm. Kolliopoulos and Steiner [9] determine an FPTAS in $O(n^3/\varepsilon)$ for problem $1|d_i = d|\sum w_i(T_i + d)$, where $\sum w_id$ is a supplementary constant. It is easy to show that an optimal sequence for $1|d_i = d|\sum w_i(T_i + d)$ is also optimal in the case of minimizing $1|d_i = d|\sum w_iT_i$, but an approximation algorithm for the first one cannot guarantee the same approximation scheme for the second. However, without any supplementary constant, Kellerer and Strusevich [7] determine an FPTAS in $O(n^6 \log W/\varepsilon^3)$ where $W = \sum_{i=1}^{n} w_i$.

In the case of identical parallel machines, problem $Pm|d_i = d|\sum w_iT_i$, Kovalyov and Werner [10] show that there is no possible polynomial-time approximation algorithm unless $P = NP$. Moreover, they propose two approximation algorithms with guaranteed performances and show that the value of the solution given by one of the two approximation algorithms, denoted by X^0, satisfies the following inequality $(X^0 - X^*)/(X^* + d) \leqslant \varepsilon$ (X^* being equal to the optimal solution value).

As far as we know – except in [13] – there is no result with uniform parallel machines in the literature. Our contribution is about a dynamic programming algorithm that solves optimally the $Qm|d_i = d|\sum w_iT_i$ problem.

The rest of paper is organized as follows. In Section 2, some remarks concerning Lawler and Moore's DP algorithm are given. Section 3 is dedicated to the single machine scheduling problem ($m = 1$). In this section, we describe a new DP algorithm. In Sections 4 and 5 the DP algorithm is generalized to the identical and uniform parallel machines scheduling problem.

2. Remarks on the DP recursion proposed by Lawler and Moore

Lawler and Moore [13] proposed a general dynamic programming recursion which can be adapted for solving a high variety of scheduling problems. One of these problems is the single machine total weighted tardiness scheduling problem denoted by $1\|\sum w_iT_i$.

As presented in Section 1 of [13], "*Let $f(i,t)$ be the minimum total loss for the first ith jobs, subject to the constraint that job J_i is completed no later than time t. The recursion is defined by:*

$$f(0,t) = 0 \quad (\forall t \geqslant 0)$$
$$f(i,t) = +\infty \quad (\forall i \in \{0,1,\ldots,n\}; \quad \forall t < 0)$$
$$f(i,t) = \min\left\{\begin{array}{l} f(i,t-1), \\ \alpha_i(t) + f(i-1,t-a_i), \\ \beta_i(t) + f(i-1,t-b_i) \end{array}\right\} \quad (\forall i \in \{1,\ldots,n\}; \quad \forall t \geqslant 0)$$

where: the processing of job J_i requires a_i units of time in one mode and b_i units in the other. In the first mode, a loss of $\alpha_i(t)$ units is incurred upon the completion of the job at time t, and $\beta_i(t)$ units in the other. The problem is solved by the calculation of $f(n,T)$, where T is a sufficiently large number. The overall computation requires on the order of nT computational steps." The authors propose an application of this recursive relation to several scheduling problems in Section 10 of their paper, they consider the problem denoted by $1|d_i = d|\sum w_iT_i$.

Let (A), (B) and (C) denote respectively the set of early jobs, the fully tardy jobs and the straddling job. The jobs are numbered according to WLPT order (Weighted Longest Processing Time first, i.e. $w_i/p_i \leqslant w_{i+1}/p_{i+1}$). First, the authors assume that there is a straddling job denoted by J_k. They define $f^{(k)}(n,t)$, the best solution for the subset of jobs $\mathscr{J} \setminus \{J_k\} = \{J_1, J_2, \ldots, J_{k-1}, J_{k+1}, \ldots, J_n\}$. They define $\alpha_i(t) = 0$, $\beta_i(t) = w_it$, $a_i = p_i$ and $b_i = 0$ so that the recursive function takes the following form:

$$f^{(k)}(0,t) = 0 \quad (\forall t \geqslant 0)$$
$$f^{(k)}(i,t) = +\infty \quad (\forall i \in \{0,1,\ldots,n\} \setminus \{k\}; \ \forall t < 0)$$
$$f^{(k)}(i,t) = \min\left\{\begin{array}{l} f^{(k)}(i,t-1), \\ f^{(k)}(i-1,t-p_i), \\ f^{(k)}(i-1,t) + w_it \end{array}\right\} \quad (\forall i \in \{1,\ldots,n\} \setminus \{k\}; \ \forall t \geqslant 0).$$

The second term of the recursive relation denotes the case where J_i is in (A) and completes at time t. The last term considers that J_i is in (B). Suppose J_i is in B. Then, because the jobs are taken in WLPT order, job J_i has to be the first tardy job (see Fig. 1), so that tardy jobs are in WSPT order. With $P_j = \sum_{k=1}^{j} p_k$ we have $C_i = P_n - P_{i-1} + t$ and the cost generated by J_i is equal to $w_i(C_i - d)$.

A more concise expression of the recursive relation is:

$$f^{(k)}(0,t) = 0 \quad (\forall t \geqslant 0),$$
$$f^{(k)}(i,t) = +\infty \quad (\forall i \in \{0,1,\ldots,n\} \setminus \{k\}; \ \forall t < 0)$$
$$f^{(k)}(i,t) = \min\left\{\begin{array}{l} f^{(k)}(i-1,t-p_i), \\ f^{(k)}(i-1,t) + w_i(P_n - P_{i-1} + t - d) \end{array}\right\} \quad (\forall i \in \{1,\ldots,n\} \setminus \{k\}; \ \forall t \geqslant 0).$$

Then, the cost due to the straddling job is added to $f^{(k)}(n,t)$ and the values of k and t for which $f^{(k)}(n,t) + w_k(t + p_k - d)$ is minimum "*indicate the identity of the job in Class (C) and its time of completion, while $f^{(k)}(n,t)$ can be used to determine the jobs in Classes (A) and (B).*" Note that for the single machine case, the completion time of the last job is always equal to P_n, which is the makespan of any semi-active schedule. This makes easy the determination of the completion time of the straddling job.

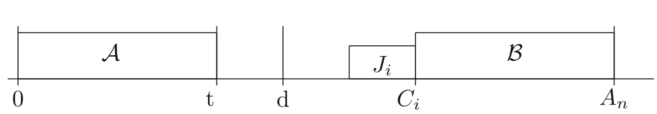

Fig. 1. Single machine case – position of a new tardy job.