



Scheduling a maintenance activity to minimize total weighted completion-time

Gur Mosheiov*, Assaf Sarig

School of Business Administration, The Hebrew University, Jerusalem 91905, Israel
Department of Statistics, The Hebrew University, Jerusalem 91905, Israel

ARTICLE INFO

Article history:

Received 9 April 2008

Received in revised form 22 October 2008

Accepted 8 November 2008

Keywords:

Scheduling

Single machine

Weighted completion time

Maintenance activity

Dynamic programming

Heuristics

ABSTRACT

We study a single machine scheduling problem. The processor needs to go through a maintenance activity, which has to be completed prior to a given deadline. The objective function is minimum total weighted completion time. The problem is proved to be NP-hard, and an introduction of a pseudo-polynomial dynamic programming algorithm indicates that it is NP-hard in the ordinary sense. We also present an efficient heuristic, which is shown numerically to perform well.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Lee and Chen [1] studied the problem of scheduling a maintenance activity on parallel identical machines with the objective of minimum total weighted completion time. There are two underlying assumptions in their model: (i) the maintenance activity requires a given period during which no production is possible, and (ii) the maintenance must be completed prior to (not after) a pre-specified deadline (since any delay increases significantly the risk of system failure). Clearly, both assumptions reflect most real-life systems which need maintenance.

Due to the many applications, scheduling a maintenance activity became a popular topic among researchers in recent years. There are studies assuming an optional maintenance which improves the processor's performance, see e.g. Lee and Leon [2]. Other models consider maintenance scheduling in a flow-shop environment [3], on parallel identical machines [4], a flexible maintenance [5], and a periodic maintenance (see e.g. [6–8]). A slightly different paper [9] focuses on scheduling a deteriorating maintenance in flow-shops and open-shops.

In this note we focus on the single machine version of the problem studied by Lee and Chen [1]. We assume that all n jobs are available for processing at time zero, and preemption is not allowed. The processing time of job j is denoted by P_j , $j = 1, \dots, n$. The weight of job j is denoted by W_j , $j = 1, \dots, n$. For a given job schedule, the completion time of job j is denoted by C_j , $j = 1, \dots, n$. We denote by t the duration of the maintenance activity, which must be performed and completed not after (the deadline) T . In order to guarantee a feasible solution, we assume that $t \leq T$. Thus, the maintenance activity can start at any time during the interval $[0, T - t]$. P_j , $j = 1, \dots, n$, where t and T are assumed to be integers.

The objective function is the classical minimum total weighted completion time: $\sum_{j=1}^n W_j C_j$. Let MA denotes the maintenance activity. Then, the problem studied here is $1/MA / \sum W_j C_j$.

* Corresponding author at: School of Business Administration, The Hebrew University, Jerusalem 91905, Israel. Tel.: +972 2 588 3108; fax: +972 2 588 1341.

E-mail addresses: msomer@mssc.huji.ac.il (G. Mosheiov), assafsarig@mssc.huji.ac.il (A. Sarig).

First we prove that the problem is NP-hard. Then we introduce a pseudo-polynomial dynamic programming algorithm, implying that the problem is NP-hard in the ordinary sense. We also suggest an efficient heuristic and a tight lower bound on the optimal weighted completion time value. In the last section we present the results of our numerical tests.

2. NP-hardness and a dynamic programming algorithm

Scheduling problems with *unavailability constraints* assume that a pre-specified time interval is given, during which the processor is stopped. Problems within this category have been studied by several researchers (see e.g. the survey paper by Lee [10]). As mentioned above, the setting studied here allows some flexibility: the unavailability (i.e. the maintenance) period can be performed at any time prior to a given deadline. We show that for minimum total weighted completion time on a single machine, this very realistic setting does not change the problem complexity:

Proposition 1. *Problem 1/MA/ $\sum W_j C_j$ is NP-hard.*

Proof. Similar to Lee [10] the proof is based on a reduction from the well-known *Partition Problem*:

Given a set $K = \{a_1, a_2, \dots, a_k\}$ of positive integers such that $\sum_{a_j \in K} a_j = 2A$, is there a subset $S \subseteq K$ such that $\sum_{a_j \in S} a_j = A$?

Given an instance of the *Partition Problem*, we construct the following instance of 1/MA/ $\sum W_j C_j$:

Let $n = k$. Job processing times are: $P_j = a_j$ for $j = 1, \dots, k$. Job weights are $W_j = a_j$ for $j = 1, \dots, k$. Let $T = t + A$, where t is the duration of the maintenance.

The objective function threshold is $y = \sum_{j \leq i} a_i a_j + tA$.

Since $P_j = W_j = a_j$ for all j , all job sequences are identical with respect to the index P_j/W_j . Let S_1 (S_2) be the set of all the jobs scheduled prior to (after) the maintenance activity. Let m denote the number of jobs contained in the set S_1 . It is easy to verify that for a given job schedule, and a maintenance of time t , the objective function value is:

$$\begin{aligned} Z &= P_1 W_1 + (P_1 + P_2) W_2 + \dots + (P_1 + \dots + P_m) W_m + (P_1 + \dots + P_m + t + P_{m+1}) W_{m+1} \\ &\quad + \dots + (P_1 + \dots + P_m + t + P_{m+1} + \dots + P_n) W_n \\ &= a_1 a_1 + (a_1 + a_2) a_2 + \dots + (a_1 + \dots + a_m) a_m + (a_1 + \dots + a_m + t + a_{m+1}) a_{m+1} \\ &\quad + \dots + (a_1 + \dots + a_m + t + a_{m+1} + \dots + a_n) a_n \\ &= \sum_{j \leq i} a_i a_j + t \sum_{a_j \in S_2} a_j. \end{aligned}$$

We show that there is a schedule for this instance such that $\sum_{j=1}^n W_j C_j \leq y$ if and only if there is a subset $S \subseteq K$ for *Partition* such that $\sum_{a_j \in S} a_j = A$.

(\Rightarrow): Suppose that there exists a subset $S \subseteq K$ for *Partition* such that $\sum_{a_j \in S} a_j = A$. We construct the following schedule: first the jobs of the set S are scheduled, then the maintenance activity, and finally the jobs of the set $K \setminus S$. This schedule is feasible since the maintenance activity is completed at time $A + t = T$, and the total weighted completion time is:

$$\sum_{j=1}^n W_j C_j = \sum_{j \leq i} a_i a_j + t \sum_{a_j \in S_2} a_j = \sum_{j \leq i} a_i a_j + t \sum_{a_j \in K \setminus S} a_j = \sum_{j \leq i} a_i a_j + tA = y.$$

(\Leftarrow): Assume that a feasible schedule exists with a total weighted completion time of no more than y . We show that the total weighted completion time cannot be strictly smaller than y . A total weighted completion time which is smaller than y implies that $\sum_{a_j \in S_2} a_j < A$, therefore $\sum_{a_j \in S_1} a_j > A$. However, in this case, the maintenance activity starts after time A , and consequently is completed after its upper bound $A + t$, in contrary to the assumption. Thus, $\sum_{j \leq i} a_i a_j + t \sum_{a_j \in K \setminus S} a_j = y$, implying that $\sum_{a_j \in K \setminus S} a_j = \sum_{a_j \in S} a_j = A$, i.e., there is a solution to the *Partition Problem*. \square

In the following we introduce a pseudo-polynomial dynamic programming algorithm (DP). The introduction of this DP implies that the problem 1/MA/ $\sum W_j C_j$ is NP-hard in the ordinary sense. A standard pair-wise interchange argument verifies that in an optimal schedule, the jobs scheduled prior to the maintenance activity ("early" jobs) and the jobs scheduled after it ("tardy" jobs), are ordered in a non-decreasing order of P_j/W_j . Thus, in the first step of our DP we sort the jobs in this order. Recall that the maintenance activity may start at any (integer) time in the interval $[0, T - t]$. We fix this starting time (denoted by s), and repeat the DP for each possible s value (i.e. $T - t + 1$ times). In each iteration of the DP (for a given s value), the next job in this list is added either to be last in the set of the early jobs (if possible), or to be last in the set of the tardy jobs. The state variables are: (i) the index of the next job to be scheduled (k), (ii) the completion time of the last job in the set of early jobs (t_1), (iii) the completion time of the last job in the set of tardy jobs (t_2). Let $f_k(k, t_1, t_2)$ denote the minimum weighted completion time of jobs $k, k + 1, \dots, n$, given that the maintenance activity starts at time s (and completed at time $s + t$), $k - 1$ jobs have already been scheduled, the early set is completed at time t_1 and the tardy set is completed at time t_2 .

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات