Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 54 (2008) 401–410
computers & industrial engineering
www.elsevier.com/locate/dsw

Approximation algorithm for the weighted flow-time minimization on a single machine with a fixed non-availability interval

Imed Kacem *

ICD-LOSI CNRS FRE 2848, Université de Technologie de Troyes, 12 rue Marie Curie, B.P. 2060, Troyes 10010, France

Received 13 June 2007; received in revised form 4 August 2007; accepted 12 August 2007
Available online 29 August 2007

Abstract

In this article, we consider the non-resumable case of the single machine scheduling problem with a fixed non-availability interval. We aim to minimize the weighted sum of completion times. No polynomial 2-approximation algorithm for this problem has been previously known. We propose a 2-approximation algorithm with $O(n^2)$ time complexity where n is the number of jobs. We show that this bound is tight. The obtained result outperforms all the previous polynomial approximation algorithms for this problem.

Keywords: Scheduling; Non-availability constraint; Approximation; Weighted flow-time

1. Introduction

This paper focuses on scheduling a set of jobs on a single machine on which a maintenance task has to be performed under the non-resumable scenario. The objective is to minimize the total weighted completion time. The machine is unavailable during a fixed interval. This type of problems has been studied in the literature under various criteria (Gharbi & Haouari, 2005; Lee & Chen, 2000). Given the aim of our study, we present a brief overview of previous works related to this subject.

The simplest case, that is minimizing total completion time with a single fixed non-availability interval (denoted $1,h_1\|\sum C_i$), was proved to be NP-Hard by Adiri, Bruno, Frostig, and Rinnooy Kan (1989) and Lee and Liman (1992). Several references studied the worst-case performance of heuristic methods (a sample of these papers includes Adiri et al. (1989), Kacem (2007), Lee & Liman (1992), Sadfi, Penz, Rapine, Błażewicz, & Formanowicz (2005), He, Zhong, & Gu (2006) & Breit (2006)).

* Tel.: +33 3 25 71 58 54; fax: +33 3 25 71 56 49.
E-mail address: imed.kacem@utt.fr

0360-8352/$ - see front matter
doi:10.1016/j.cie.2007.08.005

Recently, numerous references considered the problem of simultaneously scheduling jobs and maintenance tasks on a single machine (see for example Qi, Chen, & Tu (1999) and Qi (2007) who considered the minimization of the sum of completion times, or Chen (2006) who proposed a branch-and-bound algorithm for solving a similar problem). Others numerous references addressed the shop scheduling problems (parallel machine, flow shop and job shop problems) and they proposed exact and heuristic methods (Aggoune, 2004; Aggoune & Portmann, 2006; Allaoui & Artiba, 2006; Allaoui, Artiba, Elmaghraby, & Riane, 2006; Kubzin & Strusevich, 2006; Lee, 1996, 2004; Lee & Chen, 2000; Lee & Liman, 1993; Mosheiov, 1994; Schmidt, 2000).

The resumable version of the weighted flow-time minimization problem with arbitrary number of unavailability periods was studied by Wang, Sun, and Chu (2005). They also studied the special case with one single unavailability period. Recently, Kacem and Chu (2006) studied the $1, h_1 \| \sum w_i C_i$ problem and showed that both WSPT[1] and MWSPT[2] rules have a tight worst-case performance ratio of 3 under some conditions. They also proposed exact methods to solve this problem (Kacem & Chu, 2007; Kacem, Chu, & Souissi, 2008). Kellerer and Strusevich (2006) proposed a 4-approximation by converting the resumable solution of Wang et al. (2005) into a feasible solution for the non-resumable scenario. Despite the fact that the approach developed in this paper is also an extension of Wang et al.'s method, it is easy to see that such an approach is usually more effective than the conversion by Kellerer and Strusevich. Note that Kellerer and Strusevich proposed also an FPTAS for this problem with $O(n^4/\varepsilon^2)$ time complexity, which leads to an algorithm of $O(n^4)$ if we convert it into a polynomial 2-approximation algorithm (Kellerer & Strusevich, 2006). For these reasons, this paper is a successful attempt to develop a polynomial 2-approximation algorithm for the studied problem.

The paper is organized as follows. In Section 2, we present a description of the studied problem. Section 3 provides a description of the studied heuristic. In Section 4, we show that the above heuristic has a tight worst-case performance bound of 2. Finally, Section 5 concludes the paper.

2. Problem formulation

We have to schedule a set of n jobs $J = \{1, 2, \ldots, n\}$ on a single machine. The aim is to minimize the total weighted completion time. Every job i has a processing time p_i and a weight w_i. The machine is unavailable between T_1 and T_2 and it can process at most one job at a time. The fixed non-availability interval length is denoted by ΔT where $\Delta T = T_2 - T_1$. With no loss of generality, we consider that all data are integer and that jobs are indexed according to the WSPT rule (i.e., $\frac{p_1}{w_1} \leqslant \frac{p_2}{w_2} \leqslant \cdots \leqslant \frac{p_n}{w_n}$). Let C_i denote the completion time of job i. Due to the dominance of the WSPT order, an optimal schedule is composed of two sequences of jobs scheduled in non-decreasing order of their indexes.

If all the jobs can be inserted before T_1, the problem studied ($\mathcal{P}$) has obviously a trivial optimal solution obtained by the WSPT rule (Smith, 1956). We therefore consider only the problems in which all the jobs cannot be scheduled before T_1.

In the remainder of this paper, $\varphi^*(Q)$ denotes the minimal weighted sum of the completion times for problem Q and $\varphi_S(Q)$ is the weighted sum of the completion times of schedule S for problem Q.

3. Heuristic description

This heuristic is based on the following algorithm, which extends the one proposed by Wang et al. (2005) for the resumable version of the problem. It was also successfully applied to the makespan minimization (Kacem, 2007). This algorithm generates iteratively a set of sequences. At each iteration l of the algorithm, we identify a critical job (i.e., the first job scheduled after the non-availability interval in the sequence) and we update a subset $\mathcal{G}_l$. The jobs in $\mathcal{G}_l$ represent the identified critical jobs and they will be scheduled before T_1 in the next iteration of this algorithm. The total processing times of jobs in $\mathcal{G}_l$ cannot be more than T_1 (this is the stop condition). The output of this algorithm is the best generated sequence.

[1] WSPT: Weighted Shortest Processing Time.
[2] MWSPT: Modified Weighted Shortest Processing Time.