Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-12 (2016) 425–430

Strongly Fully Polynomial Time Approximation Scheme for the Weighted Completion Time Minimization Problem on Two-Parallel Capacitated Machines ⋆

Imed Kacem * **Myriam Sahnoune** ** **Günter Schmidt** ***

* *Université de Lorraine, LCOMS EA 7306, 57000, Metz, France (e-mail: imed.kacem@ univ-lorraine.fr).*
** *Université de Lorraine, LCOMS EA 7306, 57000, Metz, France (e-mail: myriam.sahnoune@ univ-lorraine.fr)*
*** *Universität des Saarlandes, ORBI, Sarbrucken, Germany (e-mail: gs@orbi.uni-saarland.de)*

Abstract: We consider the total weighted completion time minimization for the two-parallel capacitated machines scheduling problem. In this problem, one of the machines can process jobs until a certain time T_1 after which it is no longer available. The other machine is continuously available for performing jobs at any time. We prove the existence of a strongly Fully Polynomial Time Approximation Scheme (FPTAS) for the studied problem, which extends the results for the unweighted version (see Kacem, Lanuel and Sahnoune (2011)). Our FPTAS is based on the simplification of a dynamic programming algorithm.

Keywords: Approximation, scheduling, FPTAS, heuristic, availability constraint.

1. INTRODUCTION

Motivated by important real and industrial applications, the problems of scheduling with non-availability constraints have been a challenging subject for many research teams from different fields (Computer Science, Operational Research, Logistics...). Such applications include especially maintenance activities and non-availability of resources. Different classes of these problems have been recently studied in the literature. Various approaches have been proposed (for instance, the reader is invited to consult the state-of-the-art paper by Schmidt (2000)). In this context, this paper is devoted to study a specific model related to this class of scheduling problems. Indeed, we consider the total weighted completion time minimization for the two-parallel capacitated machines scheduling problem. In this problem, one of the machines can process jobs until a certain time T_1 after which it is no longer available. To the best of our knowledge, the weighted version of the problem studied in this paper has not been addressed in previous references. That is why this paper is a good attempt to study this problem and examine the existence of a strongly Fully Polynomial Time Approximation Scheme (FPTAS) for the above problem.

More precisely, the considered problem consists in scheduling n jobs on two parallel machines where one of these machines is not available after time T_1. The other machine is continuously available for processing jobs at any time. The objective is to elaborate a feasible schedule by minimizing the total weighted completion time. Given the aim of this paper, we recall some related works on the unweighted version of the studied problem. In Lee and Liman (1993), the authors elaborated a 3/2-approximation algorithm. In Liao, Chao and Lin (2009), the authors introduced some upper and lower bounds and proposed a branch-and-bound procedure for the same problem. It is worth-noting that other general approximation methods can be used to elaborate an FPTAS for the studied problem (as an example, the approach by Kovalyov and Kubiak (1999b) or Woeginger can be applied). Despite these interesting methods, the time complexity will not be strongly polynomial without adapted upper bound and lower bounds for the approximate values of some variables used in these algorithms. In contrast to the technique by Woeginger (2000), the introduction of these bounds leads to a reduced time complexity. Recently, Kacem, Lanuel and Sahnoune (2011) proposed an FPTAS that can be implemented in a strongly polynomial time for the unweighted case. The unconstrained version of the problem has been studied by Sahni (1976) who established the existence of an FPTAS with a strongly polynomial time.

The paper is organized as follows. In Section 2, the problem formulation is presented. Then, a dynamic programming algorithm is described in Section 3. The proposed FPTAS for the total weighted completion time minimiza-

⋆ This work has been supported by the Université de Lorraine (France) and the Universität des Saarlandes (Germany) during the year 2015.

10.1016/j.ifacol.2016.07.643

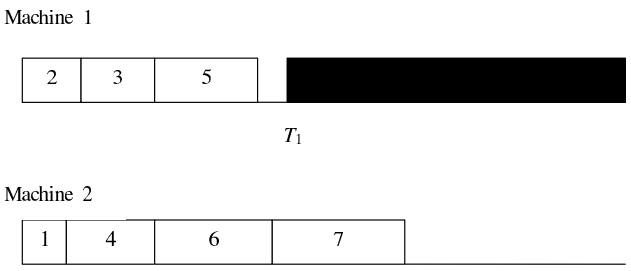

Fig. 1. Example illustration

tion problem is discussed in Section 4. Finally, Section 5 is a conclusion in which some perspectives are introduced.

2. WEIGHTED COMPLETION TIME MINIMIZATION ON CAPACITATED TWO-PARALLEL MACHINES

The problem consists in scheduling n jobs on two-parallel machines, with the aim of minimizing the total weighted completion time (i.e., the total weighted flow-time since all the jobs are ready for processing at time 0). The first machine is only available for a given period of time $[0, T_1]$ (i.e., after T_1 it can no longer process any job). This parameter T_1 is known in advance. The second machine is continuously available. Every machine can process at most one job at a time. Every job i is characterized by a processing time p_i and a weight w_i. Without loss of generality, we consider that all data are integers and that jobs are indexed according to the $WSPT$ rule:

$$p_1/w_1 \leq p_2/w_2 \leq ... \leq p_n/w_n \quad (1)$$

Due to the dominance of the $WSPT$ rule, an optimal solution is composed of two subsets (one subset for each machine) of jobs scheduled in nondecreasing order of their indexes (see Smith (1956)). In Fig. 1 , we present a feasible schedule for a 7-job instance characterized by the following data: $p_1 = 1$, $w_1 = 2$, $p_2 = 2$, $w_2 = 3$, $p_3 = 3$, $w_3 = 3$, $p_4 = 4$, $w_4 = 4$, $p_5 = 4$, $w_5 = 3$, $p_6 = 5$, $w_6 = 3$, $p_7 = 5$, $w_7 = 2$ and $T_1 = 10$.

In the remainder of the paper, we denote by $\mathcal{Q}$ the studied problem, by $\mathcal{F}^*(Q)$ the minimal sum of the weighted completion times for problem Q and by $\mathcal{F}_S(Q)$ the sum of the completion times of schedule S for problem Q. Some necessary standard definitions related to the approximation field are recalled for self-consistency:

Definition 1. A *ρ-approximation algorithm* for a problem of minimizing an objective function φ is an algorithm such that for every instance π of the problem it gives a solution S_π verifying $\varphi(S_\pi)/\varphi(OPT_\pi) \leq \rho$ where OPT_π is the optimal solution of π. Moreover, ρ is called the *worst-case bound* of the above algorithm.

Definition 2. A class of $(1+\varepsilon)$-*approximation algorithms* represents an FPTAS, if its running time is bounded by a polynomial function in $1/\varepsilon$ and the instance size for every $\varepsilon > 0$. It is well-known that an FPTAS is the best possible result for an NP-hard problem unless P=NP.

Definition 3. A class of $(1+\varepsilon)$-*approximation algorithms* is a PTAS (*Polynomial-Time Approximation Scheme*), if its running time is a polynomial function in the instance size and an arbitrary function in $1/\varepsilon$ for every $\varepsilon > 0$.

Proposition 4. If $\sum_{i=1}^{n} p_i \leq 2T_1$, then problem $(\mathcal{Q})$ has an FPTAS with a strongly polynomial time.

Proof. We relax the non-availability constraint (i.e., we assume that the first machine is continuously available). Then, the relaxed problem has an FPTAS of a strongly polynomial time according to Sahni (1976). Let σ_1 be the obtained schedule by applying such an FPTAS for the relaxed problem, B'_1 be the completion time of the last job scheduled on the first machine and B'_2 denote the completion time of the last job scheduled on the second machine. By assumption we know that $B'_1 + B'_2 \leq 2T_1$. Therefore, either $B'_1 \leq T_1$ or $B'_2 \leq T_1$ must hold. If $B'_1 \leq T_1$, then σ_1 is also a $(1+\epsilon)$-approximation for the original problem $\mathcal{Q}$. If $B'_2 \leq T_1$, then by swapping the two machines, a $(1+\epsilon)$-approximation schedule σ'_1 is obtained such that σ'_1 is also feasible for the original problem $\mathcal{Q}$. Thus, the proposition is established. ■

Based on Proposition 4, we limit our investigation to the case where

$$\sum_{i=1}^{n} p_i > 2T_1 \quad (2)$$

3. DYNAMIC PROGRAMMING PROCEDURE

In this section, we show that the studied problem can be optimally solved by applying the following standard dynamic programming procedure B. Such a procedure needs $n+1$ iterations in which it creates some sets of states. At every iteration k, a set $\mathcal{U}_k$ composed of states is generated $(0 \leq k \leq n)$. Each state $[t, f]$ in $\mathcal{U}_k$ is associated to a feasible schedule for the first k jobs. Variable t denotes the completion time of the last job scheduled on the first machine before T_1 and f is the total weighted completion time of the corresponding schedule. The following algorithm describes the proposed method:

Algorithm B

(i). $\mathcal{U}_0 := \{[0, 0]\}$.

(ii). For $k \in \{1, 2, ..., n\}$,
Initialize $\mathcal{U}_k := \varnothing$
For each state $[t, f]$ in $\mathcal{U}_{k-1}$:
1) Add state $\left[t, f + w_k\left(\sum_{i=1}^{k} p_i - t\right)\right]$ to $\mathcal{U}_k$ (i.e., job k is performed on M_2)
2) Add state $[t + p_k, f + w_k(t + p_k)]$ to $\mathcal{U}_k$ if $t + p_k \leq T_1$ (i.e., job k is performed on M_1)

Remove $\mathcal{U}_{k-1}$

(iii). $\mathcal{F}^*(\mathcal{Q}) := \min\{f \mid [t, f] \in \mathcal{U}_n\}$.

As an illustration of this dynamic programming algorithm, Table 1 depicts the generated states when we apply this method to the instance previously mentioned for the first 4 iterations.