Theoretical Computer Science 562 (2015) 377–394

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Performance guarantees for a scheduling problem with common stepwise job payoffs

Yasmina Seddik *, Christophe Gonzales, Safia Kedad-Sidhoum

Laboratoire d'Informatique de Paris 6, Université Paris 6 Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France

ARTICLE INFO

Article history:
Received 26 April 2013
Received in revised form 15 September 2014
Accepted 1 October 2014
Available online 7 October 2014
Communicated by G. Ausiello

Keywords:
Absolute approximation guarantee
Relative approximation guarantee
Scheduling
Stepwise payoffs
Release dates

ABSTRACT

We consider a single machine scheduling problem with unequal release dates and a common stepwise payoff function for the jobs. The goal is to maximize the sum of the jobs payoffs, which are defined with regard to some common delivery dates. The problem is strongly NP-hard. In this paper, we propose a polynomial time approximation algorithm with both absolute and relative performance guarantees. The time complexity of the approximation algorithm is $O(KN \log N)$, N being the number of jobs and K the number of delivery dates.

1. Introduction

We consider a single machine scheduling problem with unequal release dates and a common stepwise payoff function for the jobs. The problem addressed in this paper originates from a real issue in book digitization, where a manufacturer digitizes the collection of the French National Library. The books to be digitized arrive at the manufacturer over time. The client wishes to receive the digitized books as soon as possible. For this reason, some delivery dates are set by the client. Each digitized book yields a payoff to the manufacturer: the earlier its delivery, the greater the payoff. Each book has the same significance regarding the payoffs.

We therefore consider a single machine scheduling problem where a set $\mathcal{J}^{all}$ of N jobs $J_1, \dots, J_N$ must be scheduled nonpreemptively on a single machine. Each job J_i has a processing time $p_i > 0$ and a release date $r_i \geq 0$. K delivery dates are given: $0 < D_1 < D_2 < \cdots < D_K$; as well as K payoff values $\gamma_1 > \gamma_2 > \cdots > \gamma_K > 0$. We assume that all parameters are integer.

A schedule S is defined as a vector of job completion times. We denote by $C_i(S)$ the completion time of job J_i in S. The payoff $v_i(S)$ of job J_i in schedule S is defined by the following decreasing stepwise payoff function.

* Corresponding author.
E-mail addresses: yasmina.seddik@lip6.fr (Y. Seddik), christophe.gonzales@lip6.fr (C. Gonzales), safia.kedad-sidhoum@lip6.fr (S. Kedad-Sidhoum).

http://dx.doi.org/10.1016/j.tcs.2014.10.010
0304-3975/

$$v_i(S) = \begin{cases} \gamma_1 & \text{if } 0 < C_i(S) \le D_1 \\ \gamma_2 & \text{if } D_1 < C_i(S) \le D_2 \\ \vdots & \\ \gamma_K & \text{if } D_{K-1} < C_i(S) \le D_K \\ 0 & \text{if } D_K < C_i(S) \end{cases}$$

The payoff $v(S)$ of a schedule S is computed as: $v(S) = \sum_{i=1}^{N} v_i(S)$. The objective is to maximize $v(S)$. Extending the three-field notation of Graham et al. [1], the addressed problem is denoted as $1|r_i|\sum v_i$.

The particular case of $1|r_i|\sum v_i$ where the values $\gamma_1, \dots, \gamma_K$ are fixed as $\gamma_k = K - k + 1$ for $k = 1, \dots, K$, has been proven to be strongly NP-hard [2]. Hence $1|r_i|\sum v_i$ is strongly NP-hard.

The contribution of this paper is to provide a polynomial time approximation algorithm for $1|r_i|\sum v_i$, with an absolute performance guarantee of $\sum_{k=2}^{K} \gamma_k$, and an approximation ratio equal to $1/2$. To the best of our knowledge, only a few works (detailed below) consider scheduling problems with stepwise payoff/cost functions, and none of them proposes approximation algorithms.

The criterion $\sum v_i$ is a special case of stepwise job payoff function criterion, since all the jobs have the same stepwise payoff function. The most general case is the one where each job has a distinct stepwise payoff function, while an intermediate case is the one where the breakpoints (where the function value changes) are common but the payoff values are job related.

Detienne et al. [3] consider the single machine problem without release dates and with job related stepwise cost functions. They provide an exact method dealing with a graph representation of the scheduling problem and Lagrangian bounds. The problem with release dates is also considered but the proposed method is much less efficient in this case. Curry and Peters [4] deal with an online problem on parallel machines with reassignments, where stepwise increasing job cost functions must be minimized. At each job arrival a new schedule is computed with the available jobs, using a Branch and Price method. Janiak and Krysiak [5] consider the single machine problem without release dates and with job related stepwise payoff functions. They provide some list strategies heuristics for this problem, based on processing times and payoff values of the jobs; and two strategies based on modifications of Moore–Hodgson's algorithm for $1||\sum U_i$ [6]. Moore–Hodgson's algorithm is also exploited by Tseng et al. [7] for the same problem, as a part of a constructive heuristic algorithm. Moreover, Tseng et al. [7] define some neighborhood structures and a Variable Neighborhood Search method. An adaptation of the Moore–Hodgson's algorithm [6] is also used in this paper, and is presented as SDD-algorithm in Section 2.2.

The special case where the breakpoints are common to all the jobs is considered by Yang [8], which provides a Branch and Bound method. For the same problem, where additionally the number of breakpoints is fixed, Janiak and Krysiak [5] provide a pseudopolynomial time algorithm. Janiak and Krysiak [9] consider the variant of the objective function where jobs have the same breakpoints, but the payoff stepwise function is nondecreasing, which implies that some breakpoints are not significant for some of the jobs. They propose several heuristics to solve the problem with unrelated parallel processors.

The works cited above do not consider release dates. Notice that the problem addressed in this paper becomes polynomial if release dates are not considered [2]. As for problems considering job related stepwise cost functions and release dates, Detienne et al. [10] propose a column generation approach for solving the parallel machines problem, while Sahin and Ahuja [11] propose mathematical programming and heuristic approaches for the single machine problem. Both these works consider job related stepwise cost functions. Seddik et al. [2] consider a single machine problem with a particular case of common stepwise payoff function for the jobs: complexity results are established, and a pseudopolynomial time algorithm for the problem with two delivery dates is proposed.

Finally, two other kinds of criteria are related to $\sum v_i$, regarding the presence of a set of common delivery or due dates. First, Hall et al. [12] study the class of problems with fixed delivery dates. They consider several classical scheduling criteria, always including the following variant: the cost of a job J_i depends on the earliest delivery date occurring after the completion of J_i. In addition, complexity results are established for several problems, with different criteria and machine configurations. Second, there exists another class of problems related to the common due dates: the generalized due date problem [13], where due dates are not related to the jobs. Instead, global due dates are defined, and before each of them, one job must complete. Then, given a schedule, the i-th scheduled job is related to the i-th due date, and its cost is computed in relation to that due date, as for classical due dates. Complexity results have been established by Hall et al. [13] for this class of problems.

The rest of this paper is organized as follows. Section 2 states some preliminaries. In Section 3 the polynomial time approximation algorithm is presented, as well as a sketch of proof of its absolute performance guarantee and some lemmas; the complete proof is given in Sections 4 and 5. In Section 6 we prove the approximation ratio of the algorithm, and in Section 7 we present some experimental results. Finally, we draw some conclusions in Section 8.

2. Preliminaries

In this section, we first give some general definitions and notations; then we propose the polynomial time SDD-algorithm (which is an extension of the algorithm given in [2]) for solving a special case of $1|r_i|\sum v_i$, which is needed for the design of the approximation algorithm.