European Journal of Operational Research 262 (2017) 40–45

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Impact of deadline intervals on behavior of solutions to the random Sequencing Jobs with Deadlines problem

Krzysztof Szkatuła[a,b,*]

[a] Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, Warszawa 01-447, Poland
[b] Siedlce University of Natural Sciences and Humanities, ul. Konarskiego 2, Siedlce 08-110, Poland

A R T I C L E I N F O

Article history:
Received 16 April 2015
Accepted 18 March 2017
Available online 24 March 2017

Keywords:
Scheduling
Combinatorial optimization
Probabilistic analysis
Approximate algorithm
Profit

A B S T R A C T

The paper analyzes the influence, exerted by the mutual relations of deadline intervals on behavior of the optimal solution values for the random Sequencing Jobs with Deadlines (SJD) problems. An asymptotically sub-optimal algorithm is proposed. It is assumed that the problem coefficients are realizations of independent uniformly distributed random variables and deadlines are deterministic. The results, presented in the paper, significantly extend knowledge on behavior of the optimal solutions to the SJD problem in the asymptotical case.

1. Introduction

The *Sequencing Jobs with Deadlines* problem (SJD) consists in maximizing the weighted number of jobs processed before their deadlines. Deadlines may be considered as special cases of due windows (due intervals), see (Janiak, Janiak, Krysiak, & Kwiatkowski, 2015). Each job j ($j = 1, \dots, n$) is to be processed on a single machine. It requires a processing time t_j and has a deadline $d_j(n)$. Deadlines are assumed to be the functions of n in order to allow for the asymptotical analysis of SJD, when $n \to \infty$. If the job is completed before its deadline, a profit p_j is earned. The objective is to maximize the total profit, which could be considered as equivalent to minimizing the total cost or minimizing the weighted sum of late jobs.

From the point of view of the deterministic scheduling problems theory, the SJD problem belongs to the class of the single machine scheduling (SMS) problems. More precisely, it is considered as a scheduling problem with optimization criteria involving due dates, classified, according to Graham notation, as $1||\Sigma w_j U_j$, see (Błażewicz, Ecker, Pesch, Schmidt, and Weglarz, 1996, p. 106). There are many research papers that deal with SMS problems, both due to their own scientific value and as parts of more generalized and complex problems. The SJD problem often occurs as a sub-problem in various sequencing and scheduling problems. In Baptiste, Croce, Grosso, and T'kindt (2010) job sequencing problems with due dates and deadlines were considered. Paper by Catanzaro, Gouveiac, and Labbé (2015) addressed the job sequencing problems with tool switching. In Baptiste and Le-Pape (2005) scheduling problems with setup constraints were analyzed. Detienne (2014) considered scheduling problems with machine availability constraints. These ones are only few, out of many, problems where the SJD problem is included as the sub-problem. In many cases, the SJD problem may be used as relaxation of the more complex problems.

It is assumed, with insignificant loss of generality, that jobs are indexed according to

$$d_1(n) \le d_2(n) \le \dots \le d_n(n). \quad (1)$$

Then, the SJD problem can be formulated as a binary (0–1) programing problem (cf. Lawler & Moore, 1969):

$$z_{OPT}(n) = \max \sum_{j=1}^{n} p_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{i} t_j x_j \le d_i(n), \quad i = 1, \dots, n$$

$$\text{where} \quad x_j = 0 \text{ or } 1, \quad j = 1, \dots, n \quad (2)$$

where $x_j = 1$ only if job j is completed before its deadline. Jobs on time should be processed in the order conform to (1), while completion of the late jobs is of no importance, since no profit is earned. If all $p_j = 1$, $j = 1, \dots, n$, then the optimization goal is to maximize the number of jobs performed within the deadlines or, equivalently, to minimize the number of tardy jobs. Without loss

* Corresponding author at: Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, 01-447 Warszawa, Poland.
E-mail address: szkatulk@ibspan.waw.pl

http://dx.doi.org/10.1016/j.ejor.2017.03.047

of generality we also assume that

$0 < t_j \le d_j(n)$ and $p_j > 0, j = 1, \dots, n.$

SJD is well known to be an NP-hard problem, see (Garey & Johnson, 1979), but it can be solved in a pseudopolynomial time by a dynamical programing method of Sahni (1976). In the literature, many algorithms have been proposed to solve the sequencing or scheduling jobs on a single machine. Many of the proposed solution techniques are based on (mixed) integer linear programing problem formulations, cf. Baptiste et al. (2010), Catanzaro et al. (2015) and Detienne (2014). Another general technique which could be used is Branch and Bound method, see Baptiste and Le-Pape (2005). There were also attempts to use genetic algorithms, see Sevaux and Dauzère-Pérès (2003) or the neural network approach, cf. El-Bouri, Balakrishnan, and Popplewell (2000). In the paper by Levner and Elalouf (2014), an improved version of the polynomial-time approximation algorithm to solve the SJD problem was presented. The above list of references has illustrative purpose and it is far from being exhaustive.

In the literature, a certain simplified version of the SJD problem was considered. In this case all jobs have identical processing times, i.e. $t_i = c,\ c > 0,\ i = 1, \dots, n$ where c is some constant. For this version of the SJD problem many efficient greedy type algorithms were proposed, cf. Puntambekar (2009). Moreover, greedy type algorithms are often used in this context in the teaching process at the universities, cf. Kocur (2010).

It can be easily observed that SJD is a special case of the well known binary (0–1) multi-constraint knapsack problem, cf. Kellerer, Pferschy, and Pisinger (2004), according to the following formulation:

$$z_{OPT}(n) = \max \sum_{i=1}^{n} c_i x_i \text{ s.t. } \sum_{i=1}^{n} a_{ji} x_i \le b_j(n), \quad x_i \in \{0, 1\}, \quad i, j = 1, \dots, n \tag{3}$$

where, in (3), $c_j = p_j,\ a_{ij} = t_j,\ 1 \le i \le j,\ a_{ij} = 0,\ j < i \le n,$ $b_j(n) = d_j(n),\ j = 1, \dots, n$. When all constraints, but last, in (3) are dropped, then SJD problem is reduced to the classical (single constraint) knapsack problem:

$$z_{OPT}(n) = \max \sum_{i=1}^{n} p_i x_i \text{ s.t. } \sum_{i=1}^{n} t_i x_i \le d_n(n), \ x_i \in \{0, 1\}, \quad i = 1, \dots, n. \tag{4}$$

It is well known that multi-constraint knapsack problem is NP hard in the strong sense, while both SJD and single-constraint knapsack problems are NP hard but not in the strong sense, cf. Garey and Johnson (1979).

There are various approaches to deal with uncertainty, e.g. defined as randomness of the problem data (coefficients), in the case of job sequencing or scheduling problems, cf. Xia, Chen, and Yue (2008). In the paper by Szkatuła (1998) asymptotic growth (as $n \to \infty$) of the value of $z_{OPT}(n)$ for the class of random SJD problems was analyzed. The goal of the present paper is to investigate the influence of the intervals of deadlines on asymptotical behavior (as $n \to \infty$) of the optimal solution values $z_{OPT}(n)$ in the case of random version of the SJD problem, where intervals of deadlines are defined by behavior of $d_1(n), d_2(n) - d_1(n), \dots, d_n(n) - d_{n-1}(n)$, more precisely by their mutual relations. A simple heuristic algorithm for solving the SJD problems is proposed and it is proven that in the average case it is asymptotically sub-optimal. The obtained results are significantly extending the ones presented in the paper mentined above.

The results achieved constitute a contribution to the field of scheduling problems as well as to the probabilistic analysis of the combinatorial optimization problems. These results could be also useful for constructing and testing approximate algorithms for solving SJD problems.

The following notation is used throughout the paper: $V_n \approx Y_n$, $n \to \infty$ denotes:

- $Y_n \cdot (1 - o_n(1)) \le V_n \le Y_n \cdot (1 + o_n(1))$ if V_n and Y_n are sequences of numbers;
- $\lim_{n\to\infty} P\{Y_n \cdot (1 - o_n(1)) \le V_n \le Y_n \cdot (1 + o_n(1))\} = 1$ if V_n is a sequence of random variables and Y_n is a sequence of numbers or random variables, where $o_n(1)$ is function fulfilling: $o_n(1) \ge 0$ and $\lim_{n\to\infty} o_n(1) = 0$.

In Section 2 some useful duality estimations of (2) are presented. These estimations are exploited in Section 3, which presents probabilistic analysis of the SJD problem. Both Sections 2 and 3 are partially based on the paper by Szkatuła (1998). For further details the reader is kindly referred to this paper. Section 4 contains the main results of the paper, related to the deadline intervals and the approximate algorithm. Section 5 discusses obtained results.

2. Lagrange function and dual estimations

Let us consider the Lagrange function of (2):

$$F_n(x, \Lambda) = \sum_{j=1}^{n} p_j x_j + \sum_{i=1}^{n} \lambda_i \cdot \left(d_i(n) - \sum_{j=1}^{i} t_j x_j \right) = \sum_{i=1}^{n} \lambda_i d_i(n) + \sum_{j=1}^{n} \left(p_j - \Lambda_j \cdot t_j \right) \cdot x_j$$

where $x = \{x_1, \dots, x_n\},\ \Lambda = \{\lambda_1, \dots \lambda_n\},\ \Lambda_j = \sum_{i=j}^{n} \lambda_i$. Let for every $\Lambda,\ \lambda_j \ge 0,\ j = 1, \dots, n$

$$\varphi_n(\Lambda) = \max_{x\in\{0,1\}^n} F_n(x, \Lambda) = \sum_{i=1}^{n} \lambda_i d_i(n) + \sum_{j=1}^{n} \left(p_j - \Lambda_j \cdot t_j \right) \cdot x_j(\Lambda_j) = \sum_{j=1}^{n} p_j(\Lambda_j) + \sum_{i=1}^{n} \lambda_i \cdot \left(d_i(n) - \sum_{j=1}^{i} t_j(\Lambda_j) \right)$$

where

$$\text{if } p_j - \Lambda_j t_j > 0 \text{ then } x_j(\Lambda_j) = 1;\ \ p_j(\Lambda_j) = p_j;\ \ t_j(\Lambda_j) = t_j;$$
$$\text{if } p_j - \Lambda_j t_j \le 0 \text{ then } x_j(\Lambda_j) = 0;\ \ p_j(\Lambda_j) = 0;\ \ t_j(\Lambda_j) = 0. \tag{5}$$

Let us denote:

$$z_n(\Lambda) = \sum_{j=1}^{n} p_j(\Lambda_j); \quad s_i(\Lambda) = \sum_{j=1}^{i} t_j(\Lambda_j);$$

$$\hat{p}(\Lambda_j) = \begin{cases} p_j(\Lambda_j) & \text{if } s_j(\Lambda) \le d_j(n) \\ 0 & \text{otherwise} \end{cases}; \quad \hat{z}_n(\Lambda) = \sum_{j=1}^{n} \hat{p}(\Lambda_j);$$

$$D_n(\Lambda) = \sum_{i=1}^{n} \lambda_i \cdot d_i(n); \quad S_n(\Lambda) = \sum_{i=1}^{n} \lambda_i \cdot s_i(\Lambda) = \sum_{j=1}^{n} \Lambda_j \cdot t_j(\Lambda_j); \quad \varphi_n(\Lambda) = z_n(\Lambda) + D_n(\Lambda) - S_n(\Lambda).$$

The problem dual to SJD (2) is then as follows:

$$\Phi_n^* = \min_{\Lambda \ge 0} \varphi_n(\Lambda).$$

By the construction of $z_n(\Lambda),\ \hat{z}_n(\Lambda),\ S_n(\Lambda),\ D_n(\Lambda),\ \varphi_n(\Lambda)$ and $\Phi_n^*(\Lambda)$ we have for any $\Lambda \ge 0$:

$$z_n(\Lambda) \ge S_n(\Lambda)$$

and

$$\hat{z}_n(\Lambda) \le z_{OPT}(n) \le \Phi_n^* \le \varphi_n(\Lambda) = z_n(\Lambda) + D_n(\Lambda) - S_n(\Lambda). \tag{6}$$