Raport Badawczy

Research Report

RB/26/2015

Random scheduling (sequencing) jobs with deadlines problem: Deadliness intervals impact on the optimal solution values behavior

K. Szkatuła

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Zakładu zgłaszający pracę:
Prof. dr hab. inż. Zbigniew Nahorski

Warszawa 2015

Random scheduling (sequencing) jobs with deadlines problem: Deadliness intervals impact on the optimal solution values behavior

Krzysztof SZKATUŁA
Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warszawa, Poland
Siedlce University of Natural Sciences and Humanities
ul. Konarskiego 2, 08-110 Siedlce, Poland
E-mail: Krzysztof.Szkatula@ibspan.waw.pl

April 16, 2015

Abstract

In the paper influence of the deadliness mutual relations (deadliness intervals) on the asymptotical optimal solution values behavior, where total profit is maximized (i.e. total cost is minimized), is considered for the case of the random Sequencing Jobs with Deadlines (SJD) problems. Asymptotically sub-optimal algorithm has been proposed. It is assumed that problem coefficients are realizations of independent, uniformly distributed over $[0, 1)$ random variables, $n \to \infty$, with deadlines remaining deterministic.

Keywords: Scheduling, Knapsack Problem, Probabilistic Analysis, Approximate Algorithm, Profit, Cost Criterion

1 Introduction

The *sequencing jobs with deadlines* problem (SJD) is to maximize the weighted number of jobs processed before their deadlines. Deadlines are special cases of due windows (due intervals) see eg. Janiak et al. [5]. Each job j ($j = 1, ..., n$) is to be processed on a single machine. It requires a processing time t_j and has a deadline $d_j(n)$. If the job is processed before its deadline, a profit p_j is earned. The objective is to maximize the total profit, which could be considered as equivalent to minimize the total cost.

From the point of view of the deterministic scheduling problems theory SJD problem belongs to the class of the single machine scheduling (SMS) problems. More precisely it is considered as scheduling problem with optimisation criteria involving due dates, classified as $1 \,||\, \Sigma w_j U_j$, see Błażewicz et al. [1], p. 106. Many research papers are dealing with SMS problems due to their own research value as well as a part of more generalized and complex problems.

Assuming that

$$d_1(n) \leq d_2(n) \leq \cdots \leq d_n(n) \tag{1}$$

the SJD problem can be formulated as a binary (0-1) programming problem (cf. Lawler and Moore [8]):

$$\begin{aligned} & z_{OPT}(n) = \max \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{i} t_j x_j \leq d_i(n), \quad i = 1, \ldots, n \\ & \text{where} \quad x_j = 0 \text{ or } 1, \quad j = 1, \ldots, n \end{aligned} \tag{2}$$

where $x_j = 1$ only if job j is processed before its deadline. Jobs on time should be processed according to (1), while late jobs may be processed in an arbitrary order or even not processed. Without loss of generality we may assume that

$$0 < t_j \leq d_j(n) \text{ and } p_j > 0, \; j = 1, \ldots, n$$

SJD is well known to be an NP-hard problem, see Garey and Johnson [4], but it can be solved in a pseudopolynomial time by a dynamical programming method of Sahni [11]. In the paper by Dudziński and Szkatuła [2] simple and efficient heuristic algorithms to solve SJD was presented. In the literature simplified version of the SJD problem, where instead of processing times t_i, $i = 1, \ldots, n$, each job takes 1 unit of time, i.e. $t_i = c$, where c is certain constant. For the simplified version of SJD many efficient greedy type algorithms were proposed, cf. Puntambekar [10]. Moreover greedy type algorithms for simplified SJD are often used in the teaching process at the universities, cf Kocur [7].

It could be easily observed that SJD is a special case of the well known binary (0-1) multi-constraint knapsack problem, cf. Kellner et al. [6], in the following formulation:

$$z_{OPT}(n) = \max \sum_{i=1}^{n} c_i \cdot x_i \text{ s.t.} \sum_{i=1}^{n} a_{ji} \cdot x_i \leq b_j(n), \; x_i = 0 \text{ or } 1, \; i, \; j = 1, \ldots, n$$

where $c_j = p_j$, $a_{ij} = t_j$, $1 \leq i \leq j$, $a_{ij} = 0$, $j < i \leq n$, $b_j(n) = d_j(n)$, $j = 1, \ldots, n$. When all constraints, but last, in (2) are dropped, then SJD problem is reduced to classical (single constraint) knapsack problem:

$$z_{OPT}(n) = \max \sum_{i=1}^{n} p_i \cdot x_i \text{ s.t.} \sum_{i=1}^{n} t_i \cdot x_i \leq d_n(n), \; x_i = 0 \text{ or } 1, \; i = 1, \ldots, n \tag{3}$$

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 [4].

Probabilistic properties of the random version of the multi-constraint binary knapsack problem were analyzed in the papers by Frieze and Clarke [3], Mamer and Schilling [9], Schilling [12],[13] and Szkatuła [14],[15]. Due to the very substantial differences between the general knapsack and the SJD problems those results can not be adopted for the case of the SJD problem in the straightforward manner. In the Szkatuła paper [16] 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 this paper is to investigate the influence of the deadliness intervals on the optimal solution values $z_{OPT}(n)$ asymptotical behavior (as $n \to \infty$) in the case of random version of the SJD problem, where deadliness intervals are defined by $d_j(n),\ j = 1, \dots, n$ mutual relations. Simple heuristic algorithm solving SJD problems is proposed and it is proven that in the average case it is asymptotically sub-optimal.

The results achieved make 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:

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

In Section 2 some useful duality estimations of (2) are presented. Those estimations are exploited in the Section 3 presenting probabilistic analysis of the SJD problem. Section 4 contains the main results of the paper related to deadliness intervals and approximate algorithm. Section 5 discuss obtained results.

2 Lagrange and dual estimations

Let us consider the Lagrange function of (2), cf. Szkatuła [16]:

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

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

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

where

$$x_j(\Lambda_j) = \begin{cases} 1 & \text{if } p_j - \Lambda_j t_j > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\begin{aligned} p_j(\Lambda_j) &= \begin{cases} p_j & \text{if } p_j - \Lambda_j t_j > 0 \\ 0 & \text{otherwise} \end{cases}; \\ t_j(\Lambda_j) &= \begin{cases} t_j & \text{if } p_j - \Lambda_j t_j > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Let us denote:

$$\begin{aligned} z_n(\Lambda) &= \sum_{j=1}^{n} p_j(\Lambda_j);\ s_i(\Lambda) = \sum_{j=1}^{i} t_j(\Lambda_j); \\ \hat{p}(\Lambda_j) &= \begin{cases} p_j(\Lambda_j) & \text{if } s_j(\Lambda) \leq d_j(n) \\ 0 & \text{otherwise} \end{cases};\ \hat{z}_n(\Lambda) = \sum_{j=1}^{n} \hat{p}(\Lambda_j); \\ D_n(\Lambda) &= \sum_{i=1}^{n} \lambda_i \cdot d_i(n);\ S_n(\Lambda) = \sum_{i=1}^{n} \lambda_i \cdot s_i(\Lambda) = \\ &= \sum_{j=1}^{n} \Lambda_j \cdot t_j(\Lambda_j);\ \varphi_n(\Lambda) = z_n(\Lambda) + D_n(\Lambda) - S_n(\Lambda) \end{aligned}$$

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

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

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 $\Lambda \geq 0$:

$$z_n(\Lambda) \geq S_n(\Lambda)$$

and

$$\hat{z}_n(\Lambda) \leq z_{OPT}(n) \leq \Phi_n^* \leq \varphi_n(\Lambda) = z_n(\Lambda) + D_n(\Lambda) - S_n(\Lambda) \tag{5}$$

3 Probabilistic analysis

The following random model of SJD problem (2), cf. Szkatuła [16], will be considered:

- $n \geq 1$, n is positive integer, $n \to \infty$, $i,\ j = 1, \ldots, n$
- t_j, p_j, are realizations of mutually independent random variables and moreover t_j, p_j, are uniformly distributed over the $(0, 1]$ interval;
- $0 \leq d_1(n) \leq d_2(n) \leq \cdots \leq d_n(n)$ and $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ are deterministic, $d_j(n)$ are functions of n.

Let us compute the distributions and expectations of $t_i(\Lambda_i)$, $p_i(\Lambda_i)$, as functions of n, Λ_i in the asymptotical case, when $n \to \infty$:

$$
\begin{aligned}
G_i(\Lambda_i, x) &= P\{t_i(\Lambda_i) < x\} = P\{t_i < x \cup t_i > x \cap p_i < \Lambda_i \cdot t_i\} = \\
&= \left\{ \begin{array}{cc} \left.\begin{array}{cc} \frac{\Lambda_i}{2} + x\left(1 - \frac{x\cdot\Lambda_i}{2}\right) & \Lambda_i \le 1 \\ 1 - \frac{1}{2\Lambda_i} + x\cdot\left(1 - \frac{x\cdot\Lambda_i}{2}\right) & \Lambda_i > 1 \end{array}\right\} & 0 < x \le \min\left\{1, \frac{1}{\Lambda_i}\right\} \\ 1 & x > \min\left\{1, \frac{1}{\Lambda_i}\right\} \end{array} \right.
\end{aligned}
$$

$$
\begin{aligned}
H_i(\Lambda_i, x) &= P\{H_i(\Lambda_i) < x\} = P\{p_i < x \cup t_i > x \cap p_i < \Lambda_i \cdot t_i\} = \\
&= \left\{ \begin{array}{l} 1 - \frac{1-x^2}{2\cdot\Lambda_i} \quad \text{if} \quad \Lambda_i \ge 1 \\ \frac{\Lambda_i}{2} + \frac{x^2}{2\cdot\Lambda_i} \quad \text{if} \quad x \le \Lambda_i \le 1 \\ x \quad \text{if} \quad \Lambda_i \le x \le 1 \\ 1 \quad \text{if} \quad x \ge 1 \end{array} \right.
\end{aligned}
$$

$$
\begin{aligned}
E(t_i(\Lambda_i)) &= \int_0^1 x \cdot dG_i(\Lambda_i, x) = \int_0^{\min\{1/\Lambda_i\}} x \cdot (1 - \Lambda_i \cdot x) dx = \\
&= \left\{ \begin{array}{cc} \frac{1}{6\cdot\Lambda_i^2} & \text{if } \Lambda_i \ge 1 \\ \frac{1}{2} - \frac{\Lambda_i}{3} & \text{otherwise } 0 \le \Lambda_i \le 1 \end{array} \right.
\end{aligned}
$$

$$
\begin{aligned}
E(p_i(\Lambda_i)) &= \int_0^1 x \cdot dH_i(\Lambda_i, x) = \int_0^1 x \cdot G_i\left(\Lambda_i, x/\Lambda_i\right) dx = \\
&= \left\{ \begin{array}{cc} \frac{1}{3\cdot\Lambda_i} & \text{if } \Lambda_i \ge 1 \\ \frac{1}{2} - \frac{\Lambda_i^2}{6} & \text{otherwise } 0 \le \Lambda_i \le 1 \end{array} \right.
\end{aligned}
$$

We are looking for such $\lambda_1, \ldots, \lambda_n \ge 0$ that

$$
E(s_i(\Lambda_i(n))) \le d_i(n), \text{ for all } i = 1, \ldots, n \tag{6}
$$

By construction $\Lambda_j = \sum_{i=j}^{n} \lambda_i,\ \lambda_j \ge 0,\ j = 1, \ldots, n$ and

$$
\Lambda_1 \ge \Lambda_2 \ge \cdots \ge \Lambda_n \text{ and therefore } E(t_1(\Lambda_1)) \le E(t_2(\Lambda_2)) \le \cdots \le E(t_n(\Lambda_n)). \tag{7}
$$

Let us observe that if for certain i, $1 \le i \le n$

$$
E(s_i(\Lambda)) = d_i(n) \text{ and } d_{i+1}(n) - d_i(n) < E(t_{i+1}(\Lambda_{i+1})) \tag{8}
$$

then

$$
E(s_{i+1}(\Lambda) > d_{i+1}(n)
$$

which means that if for some deadliness $d_1(n), d_2(n), \cdots, d_n(n)$ vector Λ, such that (8) holds, is determined, then (6) will not be fulfilled for all $i = 1, \ldots, n$. This will be caused by monotonicity of Λ_j and $E(t_j(\Lambda_j))$, refer to (7). Hence in Szkatuła [16] recursive Algorithm 1 determining $\Lambda(n)$, $\lambda_i(n) \ge 0$, $i = 1, \ldots, n$, guaranteeing that for each $0 \le d_1(n) \le d_2(n) \le \cdots \le d_n(n)$ (6) will be fulfilled, was proposed.

Algorithm 1 *Procedure to determine $\delta_j(n)$ and $\Lambda_j(n)$, $j = 1, \ldots, n$*

Initialization Step: Let $l \leftarrow 0$, $d_0(n) \leftarrow 0$
Main Recursive Step: Let

$$j^* = \max_{l<m\leq n} \left\{ m \;\middle|\; \frac{d_m(n) - d_l(n)}{m-l} = \min_{l<j\leq n} \frac{d_j(n) - d_l(n)}{j-l} \right\} \tag{9}$$

and

$$\delta_k(n) = \min\left\{ \frac{d_{j^*}(n) - d_l(n)}{j^* - l}, \frac{1}{2} \right\};\ \Lambda_k(n) = \arg\{E(t_k(\Lambda_k)) = \delta_k(n)\} \tag{10}$$

for $k = l+1, \ldots, j^*$.
Checking Step: if $j^* = n$ then the procedure is completed. Otherwise, we put $l \leftarrow j^*$ and **Main Recursive Step** is repeated until $j^* = n$.

In the Algorithm 1 values of $\delta_1(n), \delta_2(n), \ldots, \delta_n(n)$ and $\Lambda_1(n), \Lambda_2(n), \ldots, \Lambda_n(n)$ are determined. Below certain features of their construction are investigated:

- $\delta_1(n) \leq \delta_2(n) \leq \ldots \leq \delta_n(n)$;
- $\sum_{j=1}^{i} \delta_j(n) \leq d_i(n)$. If $\delta_i(n) < \delta_{i+1}(n)$ then $\sum_{j=1}^{i} \delta_j(n) = d_i(n)$;
- $\Lambda_j(n) = \begin{cases} \sqrt{\frac{1}{6\cdot\delta_j(n)}} & \text{if } 0 < \delta_j(n) \leq \frac{1}{6} \\ \frac{3}{2} - 3\cdot\delta_j(n) & \text{if } \frac{1}{6} < \delta_j(n) \leq \frac{1}{2} \end{cases}, j = 1, \ldots, n$;
- $\Lambda_1(n) \geq \Lambda_2(n) \geq \ldots \geq \Lambda_n(n)$;
- If $\delta_j(n) = \delta_{j+1}(n)$ then $\Lambda_j(n) = \Lambda_{j+1}(n)$, $\lambda_j(n) = 0$, $E(s_j(\Lambda(n)) \leq d_j(n)$;
- if $\delta_j(n) < \delta_{j+1}(n)$ then $\Lambda_j(n) < \Lambda_{j+1}(n)$, $\lambda_j(n) > 0$, $E(s_j(\Lambda(n))) = d_j(n)$;
- $\sum_{i=1}^{n} \lambda_i(n) \cdot \left(\sum_{j=1}^{i} \delta_j(n) \right) = \sum_{i=1}^{n} \lambda_i(n) \cdot E(s_i(\Lambda(n))) = \sum_{i=1}^{n} \lambda_i(n) \cdot d_i(n)$;
- $E(p_j(\Lambda_j(n))) = \begin{cases} \sqrt{\frac{2}{3}} \cdot \delta_j(n) & \text{if } 0 < \delta_j(n) \leq \frac{1}{6} \\ \frac{1}{8} + \frac{3}{2}\cdot\delta_j(n)\cdot(1-\delta_j(n)) & \text{if } \frac{1}{6} \leq \delta_j(n) \leq \frac{1}{2} \end{cases}.$

Hence:

$$\begin{aligned} E(Z_n(\Lambda(n))) &= \sum_{j=1}^{n} E(p(\Lambda_j(n))) = \\ &= \sum_{j=1}^{n} \left(\tau_j(n) \left(\frac{1}{8} + \frac{3}{2}\delta_j(n)(1-\delta_j(n)) \right) + \bar{\tau}_j(n) \sqrt{\frac{2\cdot\delta_j(n)}{3}} \right) \end{aligned}$$

where $\tau_j(n) = \begin{cases} 1 & \text{if } \frac{1}{6} < \delta_j(n) \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$ and $\bar{\tau}_j(n) = 1 - \tau_j(n)$, $j = 1, \ldots, n$.

The main result of the Szkatuła [16] paper is stated in the theorem given below.

Theorem 1 *Let p_j, t_j, $j = 1, \dots, n$, be realizations of mutually independent random variables uniformly distributed over $(0, 1]$, $d_1(n) \le d_2(n) \le \dots \le d_n(n)$, $d_j(n)$ be deterministic, and $\delta_j(n)$ be defined by the above Algorithm 1. If*

$$\frac{\ln(n)}{n \cdot \delta_1(n)} \approx 0$$

then:

$$z_{OPT}(n) \approx \sum_{j=1}^{n} \left(\tau_j(n) \left(\frac{1}{8} + \frac{3 \cdot \delta_j(n)}{2}(1 - \delta_j(n)) \right) + \bar{\tau}_j(n) \sqrt{\frac{2 \cdot \delta_j(n)}{3}} \right) \tag{11}$$

The main idea of the Theorem 1 proof is based on showing that:

$$\hat{z}_n(\Lambda(n)) \approx E(z_n(\Lambda(n))) \approx \varphi_n(\Lambda(n)) \text{ and } E(z_n(\Lambda(n))) \approx z_n(\Lambda(n))$$

and using (5). For further details refer to Szkatuła [16].

4 Deadliness intervals and approximate algorithm

Construction of the $\delta_j(n)$, $j = 1, \dots, n$, provided by the Algorithm 1 and described in the Section 3 indicates that values of the $\delta_j(n)$ as well as their properties significantly depend on the mutual relations between $d_j(n)$, $j = 1, \dots, n$. It is assumed that $d_j(n)$ are monotonic cf. (1). Formula (9) in the Algorithm 1 may be considered in different formulation, namely

$$j^* = \max_{l < j \le n} \left\{ j \;\middle|\; d_{j-1}(n) \ge \frac{j-1}{j} \cdot d_j(n) \right\} \tag{12}$$

where $l \leftarrow 0$ at the initialization step and then in the recursive steps if $j^* < n$ then $l \leftarrow j^*$and when $j^* = n$ Algorithm 1 is finished. Formula (12) is clearly indicating that values $\delta_1(n), \delta_2(n), \dots, \delta_n(n)$ depends on

$$d_1(n), d_2(n) - d_1(n), \dots, d_n(n) - d_{n-1}(n).$$

To be more precise from the construction of the $\delta_j(n)$, $\Lambda_j(n) = \sum_{i=j}^{n} \lambda_i(n)$, $j = 1, \dots, n$, it follows that:

$$\begin{aligned} &\text{if } \delta_j(n) = \delta_{j+1}(n) \text{ then } \Lambda_j(n) = \Lambda_{j+1}(n),\ \lambda_j(n) = 0 \text{ and } E(s_j(\Lambda(n))) \le d_j(n); \\ &\text{if } \delta_j(n) < \delta_{j+1}(n) \text{ then } \Lambda_j(n) > \Lambda_{j+1}(n),\ \lambda_j(n) > 0 \text{ and } E(s_j(\Lambda(n))) = d_j(n). \end{aligned}$$

From the duality theory it follows that if $\lambda_i(n) = 0$ then corresponding constraint in (2) is "inactive". It means that it is fulfilled by other "active" constraints for which $\lambda_j(n) > 0$, $i, j \in \{1, \dots, n\}$. Formula (12) is enabling to distinguish 3 different classes of $\delta_j(n)$, $\Lambda_j(n)$, $\lambda_i(n)$, $i, j = 1, \dots, n$.

Lemma 1 *If for all $j = 2, \dots, n$*

$$d_{j-1}(n) \ge \frac{j-1}{j} \cdot d_j(n) \tag{13}$$

then

$$\delta_1(n) = \delta_2(n) = \ldots = \delta_n(n); \Lambda_i(n) = \Lambda_{i+1}(n) = \lambda_n(n), \lambda_i(n) = 0, i = 1, \ldots, n-1. \tag{14}$$

Proof. If (13) holds then in Algorithm 1, according to formulas (9) and (12), Main Recursive Step is performed only once with $j^* = n$ and (14) will follow immediately from (10). ∎

In the case considered in Lemma 1 only last constraint is active and SJD problem is reduced to the single-constraint knapsack problem (3).

Lemma 2 *If for all* $j = 2, \ldots, n$

$$d_{j-1}(n) < \frac{j-1}{j} \cdot d_j(n) \tag{15}$$

then

$$\delta_1(n) < \delta_2(n) < \ldots < \delta_n(n); \ \Lambda_1(n) > \Lambda_2(n) > \ldots > \Lambda_n(n), \ \lambda_i(n) > 0, \tag{16}$$

for $i = 1, \ldots, n$.

Proof. If (15) holds then in Algorithm 1 , according to formulas (9) and (12), Main Recursive Step is performed n times with $j^* = 1, 2, \ldots, n$, and (16) will follow immediately from (10). ∎

Lemma 2 is dealing with the situation when all n constraints are active.

Lemma 3 *If there exists* $j\prime$, $1 < j\prime < n-1$ *such that*

$$d_{j-1}(n) \geq \frac{j-1}{j} \cdot d_j(n), \ j = 2, \ldots, j\prime; \text{ and } d_{j\prime+1}(n) < \frac{j\prime+1}{j\prime+2} \cdot d_{j\prime+2}(n) \tag{17}$$

then

$$\delta_1(n) = \ldots = \delta_{j\prime}(n) < \delta_{j\prime+1}(n); \ \Lambda_1(n) = \ldots = \Lambda_{j\prime}(n) > \Lambda_{j\prime+1}(n), \ \lambda_i(n) = 0, \tag{18}$$

where $i = 1, \ldots, j\prime$.

Proof. In this case when (17) holds then in Algorithm 1, according to formulas (9) and (12), first execution of the Main Recursive Step is providing $j^* = j\prime$. Then starting with $l = j\prime + 1$ Main Recursive Step will be performed at least once more and (18) will follow immediately from (10). ∎

In the Lemma 3 mixed case is considered where some, at least first $j\prime$, constraints are inactive, while some constraints, at least $j\prime + 1$ one, are active. It also may happen, that this situation will be repeated several times. For example there may exist $j\prime\prime$ and $j\prime\prime\prime$, $j\prime < j\prime\prime < j\prime\prime\prime < n$, such that constraints $j\prime + 1, \ldots, j\prime\prime$ are active, constraints $j\prime\prime + 1, \ldots, j\prime\prime\prime$ are inactive and so on.

For any given set of deadliness $d_1(n), d_2(n), \ldots, d_n(n)$ Lemmas 1, 2 or 3 are covering all possible relations between deadliness and resulting "activity" status of constraints.

Three lemmas presented above are allowing to introduce recursive intervals of deadliness corresponding to three cases considered. Theorem below presents the main result of this paper.

Theorem 2 *If all* $j = 2, \ldots, n-1$

$$d_j(n) \in \left[\frac{j}{j+1} \cdot d_{j+1}(n), \frac{j}{j-1} \cdot d_{j-1}(n)\right] \tag{19}$$

holds then Lemma 1 holds and (14) is describing mutual relations between $\delta_j(n)$, $\Lambda_j(n)$, $\lambda_i(n)$, $i, j = 1, \ldots, n$. *If for all* $j = 2, \ldots, n-1$

$$d_j(n) \in \left(\frac{j}{j-1} \cdot d_{j-1}(n), \frac{j}{j+1} \cdot d_{j+1}(n)\right) \tag{20}$$

holds then Lemma 2 holds and (16) is describing mutual relations between $\delta_j(n)$, $\Lambda_j(n)$, $\lambda_i(n)$, $i, j = 1, \ldots, n$.
If There exists $j\prime$, $2 < j\prime < n-1$ *such that for all* $j = 2, \ldots, j\prime - 1$

$$d_j(n) \in \left[\frac{j}{j+1} \cdot d_{j+1}(n), \frac{j}{j-1} \cdot d_{j-1}(n)\right] \text{ and } d_{j\prime+1}(n) < \frac{j\prime+1}{j\prime+2} \cdot d_{j\prime+2}(n) \tag{21}$$

holds then Lemma 3 holds and (18) is describing mutual relations between $\delta_j(n)$, $\Lambda_j(n)$, $\lambda_i(n)$, $i, j = 1, \ldots, n$.

Proof. In order to prove the Theorem 2 is sufficient to observe that Lemma 1 is proving (19), Lemma 2 is proving (20) and Lemma 3 is proving (21). ■

Formulas (19), (20) and (21) are defining in recursive manner deadliness intervals. If deadliness $d_j(n)$, $j = 2, \ldots, n-1$, are belonging to the corresponding deadliness intervals, as presented in the (19), (20) and (21) then it is guaranteed that $\delta_j(n)$, $\Lambda_j(n)$, $\lambda_i(n)$, $i, j = 1, \ldots, n$, will belong to one of the 3 different classes defined by Lemmas 1-3 and Theorem 2. This provides certain flexibility in defining deadliness actual values since they may vary in the proposed deadlines intervals. Because of the recursive definition of the deadlines intervals first and last deadliness may have special influence on the intervals construction. Presented below 3 examples are illustrating above defined classes of $\delta_j(n)$, $\Lambda_j(n)$, $\lambda_i(n)$, $i, j = 1, \ldots, n$.

Example 1

Let

$$d_j(n) = \frac{j}{2} \text{ and then } \delta_j(n) = \frac{1}{2},\ j = 1, \ldots, n;\ d_n(n) = \frac{n}{2}.$$

In this case assumptions (13) of Lemma 1 and of Theorem 1 are fulfilled and according to (11) we have:

$$z_{OPT}(n) \approx \frac{1}{2} \cdot n \tag{22}$$

which means that in this case only last constraint is active, optimal solution has maximum possible value and all jobs will be processed before their deadliness in the asymptotical case for the considered random model of SJD problem.

Example 2

Let

$$d_j(n) = \frac{j^2}{2n} \text{ and then } \delta_j(n) = \frac{2j-1}{2n},\ j = 1, \ldots, n;\ d_n(n) = \frac{n}{2}.$$

In this case assumptions (15) of Lemma 2 and of Theorem 1 are fulfilled. All constarints are active and all jobs will be processed before their deadliness and (22) will hold.

Example 3

Let

$$\begin{aligned} d_j(n) &= \frac{j}{4} \text{ and then } \delta_j(n) = \frac{1}{4},\ j = 1, \ldots, n^*,\ n^* = \left\lceil \frac{n}{2} \right\rceil, \\ d_j(n) &= \frac{j^2}{2n} \text{ and then } \delta_j(n) = \frac{2j-1}{2n},\ j = n^*+1, \ldots, n;\ d_n(n) = \frac{n}{2}. \end{aligned}$$

In this case assumptions (17) of Lemma 3, where $j\prime = n^*$,and of Theorem 1 are fulfilled. According to (10) $\delta_j(n) = \frac{1}{2}$, $j = n^*+1, \ldots, n$,and from (11) we have:

$$z_{OPT}(n) \approx \frac{1}{2} \cdot n - \frac{3}{32} \cdot n^* - \frac{1}{2},$$

which means that in this case some constraints will be active while some inactive and optimal solution value is smaller than the possible maximum one. Some jobs will not be processed before their deadliness (i.e. providing no profit) in the asymptotical case for the considered random model of the SJD problem.

Below the heuristic, greedy type algorithm designed to solve SJD problem (2) in the general deterministic case is presented. This algorithm is using procedure equivalent to one applied in the Algorithm 1.

Algorithm 2

Initialization Step: Let $l \leftarrow 0$, $d_0(n) \leftarrow 0$
Main Recursive Step: Let

$$j^* = \max_{l<m\leq n} \left\{ m \;\middle|\; \frac{d_m(n) - d_l(n)}{m-l} = \min_{l<j\leq n} \frac{d_j(n) - d_l(n)}{j-l} \right\}$$

and

$$\Lambda_{j^*} = \min_{\Lambda \in \Phi_{j^*}} \left\{ \Lambda \;\middle|\; \sum_{j=l+1}^{j^*} t_j(\Lambda) \leq d_{j^*}(n) \right\};\ \Phi_{j^*} = \left\{ \frac{p_{l+1}}{t_{l+1}}, \ldots, \frac{p_{j^*}}{t_{j^*}} \right\} \tag{23}$$

$$x_j \leftarrow x_j(\Lambda_{j^*}),\ j = l+1, \ldots, j^*, \text{ refer to (4).}$$

Checking Step: if $j^* = n$ then the Algorithm 2 is completed. Otherwise, we put $l \leftarrow j^*$ and **Main Recursive Step** is repeated until $j^* = n$.

Algorithm 2 has extremely low computational complexity of order of $O(n)$. It does not require sorting of elements as greedy type algorithms usually do. From computational point of view most expensive are max and min operations, which have computational cost of $O(k)$, where k is number of elements of the corresponding set. In the case when max and min operations are repeated k times over sets of order of k elements then computational complexity will be of order of $O(k^2)$. According to Lemmas 1 - 3 this situation may not occur since either Main Recursive Step will be repeated only once (Lemma 1) or it will be repeated several times (n in the case of Lemma 2) but number of elements of the corresponding sets will be small, i.e. of order of $\frac{n}{m}$, where n is number of jobs (i.e. size of the problem) and m number of necessary repetitions of the Main Recursive Step. Therefore overall computational complexity of Algorithm 2 will be of order of $O(n)$.

In the sense of the worst case analysis this algorithm is always providing feasible solutions of the SJD problem (2) which is guaranteed by (23), but it does not have any guarantees of the accuracy of the solutions provided. Therefore in the sense of the worst case analysis Algorithm 2 is the heuristic algorithm. However for the considered random model of the problem (2) Algorithm 2 is asymptotically sub-optimal. In the random case Algorithm 2 is behaving identically to the computational procedure described in the Algorithm 1. Asymptotical sub-optimality of the Algorithm 2 follows immediately from the Theorem 1.

5 Concluding remarks

In the present paper one of the classical scheduling problems - Sequencing Jobs with Deadlines problem (SJD) was considered. On the basis of the author's previous results, cf. Szkatuła [16], probabilistic analysis of the impact of the deadlines mutual relations and functional properties was performed for the random model of the SJD problem. As the result of this analysis three specific categories of the deadlines mutual relations were identified. Then, on the basis of the results achieved in Lemmas 1, 2, and 3 Theorem 2 was formulated. In the Theorem 2 recursive deadliness intervals for the considered random model of the SJD problem were defined. In this framework roles of the first and last constraint, $d_1(n)$ and $d_n(n)$ respectively, are crucial. Deadliness intervals may provide substantial flexibility in formulating SJD problems, because deadlines mutual relations may be analyzed in more convenient manner and their influence on the SJD problem solutions (list of the jobs to be performed before their deadliness and providing profit) is demonstrated in convenient and convincing manner.

Another interesting outcome of the paper is simple heuristic algorithm of very low computational complexity, which in the average case, i.e. for the considered random model of the SJD problem is asymptotically sub-optimal.

Those results are enriching knowledge base for the Scheduling Problems, especially Sequencing Jobs with Deadlines problem, theory and also may positively influence solvability of the SJD problem instances in practical applications.

Acknowledgement

The research was partially supported by National Science Center Poland (NCN), decision no.: DEC-2011/03/B/ST6/00364

References

[1] J. Błażewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Scheduling Computer and Manufacturing Processes.* Springer Verlag, 1996.

[2] K. Dudziński and K. Szkatuła. A note on sequencing jobs with deadlines problem. *European Journal of Operational Research*, 59:333–336, 1992.

[3] A. Frieze and M. Clarke. Approximation algorithms for the m-dimensional 0-1 knapsack problem: Worst case and probabilistic analysis. *European Journal of Operational Research*, 15:100–109, 1984.

[4] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[5] A. Janiak, W. A. Janiak, T. Krysiak, and T. Kwiatkowski. A survey on scheduling problems with due windows. *European Journal of Operational Research*, 242:347–357, 2015.

[6] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems.* Springer Verlag, 2004.

[7] G. Kocur. *Coures on Computer Algorithms in Systems Engineering, Greedy algorithms (lecture 10): knapsack, job sequence.* Massachusetts Institute of Technology: MIT OpenCourseWare, 2010.

[8] E. Lawler and J. Moore. A functional equation and its application to resource alocation and sequencing problems. *Management Science*, 16:77–84, 1969.

[9] J. Mamer and K. Schilling. On the growth of random knapsacks. *Discrete Applied Mathematics*, 28:223–230, 1990.

[10] A. Puntambekar. *Analysis of Algorithm and Design.* Technical Publications, 2009.

[11] S. Sahni. Algorithms for scheduling independent jobs. *Journal of ACM*, 23:116–127, 1976.

[12] K. Schilling. The growth of m-constraint random knapsacks. *European Journal of Operational Research*, 46:109–112, 1990.

[13] K. Schilling. Random knapsacks with many constraints. *Discrete Applied Mathematics*, 48:163–174, 1994.

[14] K. Szkatuła. On the growth of multi-constraint random knapsacks with various right-hand sides of the constraints. *European Journal of Operational Reserch*, 73:199–204, 1994.

[15] K. Szkatuła. The growth of multi-constraint random knapsacks with large right-hand sides of the constraints. *Operations Research Letters*, 21:25–30, 1997.

[16] K. Szkatuła. Random sequencing jobs with deadlines problem: growth of the optimal solutions values. *European Journal of Operational Research*, 109:160–169, 1998.