Available online at www.sciencedirect.com

SCIENCE DIRECT®

Fuzzy Sets and Systems 150 (2005) 77–86

FUZZY sets and systems

www.elsevier.com/locate/fss

A possibilistic approach to sequencing problems with fuzzy parameters

Adam Kasperski*

Institute of Industrial Engineering and Management, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

Received 13 November 2003; received in revised form 30 August 2004; accepted 14 September 2004
Available online 7 October 2004

Abstract

In this paper a possibilistic approach to sequencing is proposed. For each parameter, whose value is not precisely known, a possibility distribution is given. The objective is to calculate a sequence of jobs, for which the possibility (necessity) of delays of jobs is minimal. Five sequencing problems are formulated and the computational complexity of all of them is explored.

Keywords: Scheduling; Single machine; Possibility distribution; Fuzzy number

1. Introduction

Sequencing problem is a special case of a more general scheduling problem, in which each schedule can be represented by a sequence of jobs. A wide review of the classical sequencing models, together with some complexity results, can be found in [1]. In the classical problems there are some parameters given (processing times, due dates, weights, etc.), whose values must be fixed before the calculation of the optimal solution. The assumption that all the parameters are precisely known may be restrictive. For most of the real-world processes the exact values of parameters are not known a priori and this uncertainty must be taken into account.

* Tel.: +71 320 23 81; fax: +71 320 34 32.
E-mail address: adam.kasperski@pwr.wroc.pl (A. Kasperski).

0165-0114/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2004.09.007

The natural approach to modelling the uncertainty in scheduling is a stochastic one. Unfortunately, stochastic models are often hard to solve. Moreover, it may be hard or expensive to assume any specific probability distributions for unknown parameters. During the past decades an alternative approach to modelling the uncertainty, based on the fuzzy sets theory, has been proposed. Ishii et al. [10], Han et al. [9] and Tanaka and Vlach [16] have studied the single machine problems in which the due dates of jobs are fuzzy. The fuzzy due date of a job expresses the degree of satisfaction with completion time of this job. The single machine problems with fuzzy processing times have been studied by Itoh and Ishii [11], Wang et al. [17], Sung and Vlach [15] and Chanas and Kasperski [3]. In these papers different criteria have been applied to calculate the optimal solution. In [4] Chanas and Kasperski have proposed the indices of possible and necessary optimality of a given feasible sequence under uncertainty. In this approach, the notion of "optimal" becomes imprecise and the degree of optimality of a given sequence can be characterized by a number from the interval $[0, 1]$. A lot of results on the fuzzy scheduling and sequencing can also be found in [2].

In this paper we propose an approach to sequencing, which is based on the possibility theory. For each unknown parameter in the problem there is given a possibility distribution which expresses the uncertain knowledge about this parameter. The interpretation of the possibility distribution and some methods of obtaining it from the possessed knowledge are presented in detail in [6]. In order to calculate the optimal solution the indices proposed by Dubois and Prade [6] are applied. This paper is an extension of results presented in [3]. Some results obtained in [3] are recalled and some new problems are investigated. The main goal of this paper is to explore the computational complexity of all defined problems.

2. Basic notions of the possibility theory

In this section we recall some basic notions of the possibility theory, which will be used in the next part of the paper. Let X be a single-valued variable, whose value is not precisely known. There is given a normal, quasi concave and upper semicontinuos function $\mu_X : \mathbb{R} \to [0, 1]$ called the *possibility distribution* for X. The value of $\mu_X(x)$ for $x \in \mathbb{R}$ denotes the possibility of an event that X takes the value of x, i.e. $\mu_X(x) = Pos(X = x)$. The possibilistic variable X is called a *fuzzy number*. A crisp number $u \in \mathbb{R}$ can be viewed as a special case of the fuzzy number with $\mu_u(u) = 1$ and $\mu_u(x) = 0$ for $x \neq u$. We say that a fuzzy number X is *nonnegative* if $\mu_X(x) = 0$ for all $x < 0$. The interpretation of the possibility distribution and some methods of obtaining it from the possessed knowledge about variable X are explained in detail in [6].

A *trapezoidal fuzzy number* is a special case of the fuzzy number, whose possibility distribution is defined as follows (see also Fig. 1):

$$\mu_X(x) = \begin{cases} 1 & \textit{for } x \in [\underline{x},\ \overline{x}], \\ 1 - \frac{\underline{x}-x}{\alpha} & \textit{for } x \in [\underline{x} - \alpha,\ \underline{x}), \\ 1 - \frac{x-\overline{x}}{\beta} & \textit{for } x \in (\overline{x},\ \overline{x} + \beta], \\ 0 & \textit{for } x \in (-\infty,\ \underline{x} - \alpha) \cup (\overline{x} + \beta,\ \infty). \end{cases}$$

Each trapezoidal fuzzy number X can be described by a quadruple $(\underline{x}, \overline{x}, \alpha, \beta)$, where $\overline{x} \geqslant \underline{x}$, $\alpha > 0$, $\beta > 0$. The support of X, i.e. the set $[\underline{x} - \alpha, \overline{x} + \beta]$, is chosen so as to be sure that the value of X will not fall outside it. The core of X, i.e. the set $[\underline{x}, \overline{x}]$, includes the most plausible values of X.

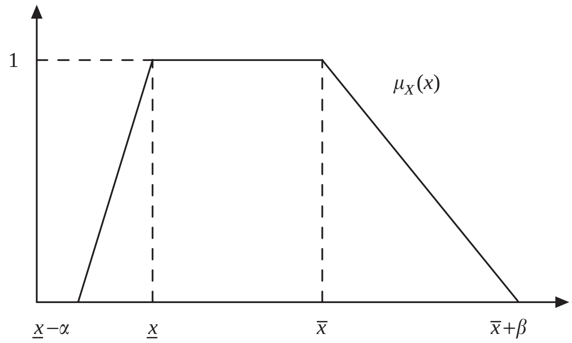

Fig. 1. A trapezoidal fuzzy number $X=(\underline{x},\overline{x},\alpha,\beta)$.

Consider two fuzzy numbers X and Y with possibility distributions, respectively, μ_X and μ_Y. The possibility distribution of the fuzzy number $Z=X+Y$ is defined by means of the Zadeh extension principle as follows:

$$\mu_Z(z)=\sup_{x\in\mathbb{R}}\min\{\mu_X(x),\mu_Y(z-x)\}.$$

If $X=(\underline{x},\overline{x},\alpha,\beta)$ and $Y=(\underline{y},\overline{y},\gamma,\delta)$ are trapezoidal fuzzy numbers, then $Z=X+Y$ is also a trapezoidal fuzzy number and

$$Z=X+Y=(\underline{x}+\underline{y},\overline{x}+\overline{y},\alpha+\gamma,\beta+\delta). \tag{1}$$

Assume that X and Y are two fuzzy numbers. We are interested in comparing X to Y, i.e. we want to characterize the possibility of the event that the value taken by X will be greater (or not less) than the value taken by Y. In [6,7] Dubois and Prade proposed the following indices (see also Fig. 2):

$$Pos(X\geqslant Y)=\sup_{x\geqslant y}\min\{\mu_X(x),\mu_Y(y)\}, \tag{2}$$

$$Pos(X>Y)=\sup_{x}\inf_{y\geqslant x}\min\{\mu_X(x),1-\mu_Y(y)\}. \tag{3}$$

It is easy to notice that the values of both indices belong to the interval [0, 1].

Let

$$Nec(X>Y)=1-Pos(Y\geqslant X). \tag{4}$$

Thus $Nec(X>Y)$ characterizes the necessity of the event that X will be greater than Y. It is easy to check that $Pos(X\geqslant Y)\geqslant Pos(X>Y)\geqslant Nec(X>Y)$ for all fuzzy numbers X and Y (see [6]).

Let $X=(\underline{x},\overline{x},\alpha,\beta)$ and $Y=(\underline{y},\overline{y},\gamma,\delta)$ be two trapezoidal fuzzy numbers. Then it holds [6]

$$Pos(X\geqslant Y)=\max\left(0,\min\left(1,1+\frac{\overline{x}-\underline{y}}{\beta+\gamma}\right)\right), \tag{5}$$

$$Pos(X>Y)=\max\left(0,\min\left(1,\frac{\overline{x}-\overline{y}+\beta}{\beta+\delta}\right)\right), \tag{6}$$

$$Nec(X>Y)=\max\left(0,\min\left(1,\frac{\underline{x}-\overline{y}}{\alpha+\delta}\right)\right). \tag{7}$$

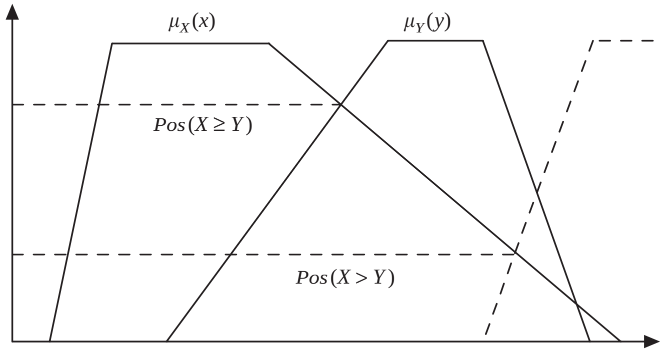

Fig. 2. The indices $Pos(X > Y)$ and $Pos(X \geqslant Y)$.

Let $u \in \mathbb{R}$ be a crisp number and $Y = (\underline{y}, \overline{y}, \gamma, \delta)$ be a trapezoidal fuzzy number. Then we have

$$Pos(u \geqslant Y) = \max\left(0, \min\left(1, 1 + \frac{u - \underline{y}}{\gamma}\right)\right), \tag{8}$$

$$Pos(u > Y) = Nec(u > Y) = \max\left(0, \min\left(1, \frac{u - \overline{y}}{\delta}\right)\right). \tag{9}$$

Finally, assume that $u, v \in \mathbb{R}$ are crisp numbers. Then, it is easy to check that $Pos(u > v) = Nec(u > v) = 1$ if and only if $u > v$ and 0 otherwise. Thus, both indices can be viewed as a generalization of the relation $>$ to the fuzzy case.

We can also characterize the possibility of the event that the value taken by X will be equal to the value taken by Y. It can be done by means of the following index:

$$Pos(X = Y) = \min\{Pos(X \geqslant Y), Pos(Y \geqslant X)\}. \tag{10}$$

Let

$$Nec(X \neq Y) = 1 - Pos(X = Y). \tag{11}$$

Thus index $Nec(X \neq Y)$ characterizes the necessity of the event that the values taken by X and Y will be different. Index $Nec(X \neq Y)$ can be easily calculated if X and Y are trapezoidal fuzzy numbers (see formulae (5), (10) and (11)). The following property holds:

Property 1. *Let $X = (\underline{x}, \overline{x}, \alpha, \beta)$ and $Y = (\underline{y}, \overline{y}, \gamma\ \delta)$ be two trapezoidal fuzzy numbers. Then $Nec(X \neq Y) \leqslant 0$ if and only if $[\underline{x}, \overline{x}] \cap [\underline{y}, \overline{y}] \neq \emptyset$.*

Proof. Assume that $[\underline{x}, \overline{x}] \cap [\underline{y}, \overline{y}] \neq \emptyset$. Then $\overline{x} \geqslant \underline{y}$ and $\overline{y} \geqslant \underline{x}$, which implies $Pos(X \geqslant Y) = 1$ and $Pos(Y \geqslant X) = 1$ (see formula (5)). From definition of $Nec(X \neq Y)$ (see (10) and (11)) we get that $Nec(X \neq Y) = 0$.

Assume now that $Nec(X \neq Y) \leqslant 0$. Then, from (11) we get $Pos(X \geqslant Y) \geqslant 1$ and $Pos(Y \geqslant X) \geqslant 1$. From (5) we get that $\overline{x} \geqslant \underline{y}$ and $\overline{y} \geqslant \underline{x}$, which implies $[\underline{x}, \overline{x}] \cap [\underline{y}, \overline{y}] \neq \emptyset$. $\square$

3. Formulation of the sequencing problems

We are given a set of jobs $J = \{1, \ldots, n\}$ to be processed on a single machine. All the jobs are ready for processing at time 0. Pre-emption of the jobs and idle machine times are not allowed. Each schedule is represented by a sequence (permutation) of jobs $\pi = (\pi(1), \ldots, \pi(n))$, $\pi(i) \in J$, $i = 1, \ldots, n$. There may be given some precedence constraints between jobs: if $i \to j$, $i, j \in J$, then job i must appear before job j in each *feasible* sequence π (job j is called a *successor* of job i). We will denote by Π the set of all the feasible sequences. For each job $i \in J$ there are given: a processing time P_i, a due date D_i and a weight w_i. It is assumed that the processing times and the due dates are nonnegative fuzzy numbers and all the weights are positive, crisp numbers. The due date of job $i \in J$ expresses the desired completion time of job i and the weight expresses the importance of job i. Let $C_i(\pi)$, $i \in J$, denote the completion time of the ith job in a given sequence π. If $i = \pi(k)$, $k = 1, \ldots, n$, then $C_i(\pi) = \sum_{j=1}^{k} P_{\pi(j)}$. Note that completion time $C_i(\pi)$, $i \in J$, is a fuzzy number, whose membership function expresses the possibility distribution for the completion time of the ith job. Consider the following sequencing problems:

$$\text{PS1: } \min_{\pi \in \Pi} \max_{i \in J} \{w_i Pos(C_i(\pi) > D_i)\},$$
$$\text{PS2: } \min_{\pi \in \Pi} \max_{i \in J} \{w_i Nec(C_i(\pi) > D_i)\},$$
$$\text{PS3: } \min_{\pi \in \Pi} \sum_{i \in J} w_i Pos(C_i(\pi) > D_i),$$
$$\text{PS4: } \min_{\pi \in \Pi} \sum_{i \in J} w_i Nec(C_i(\pi) > D_i).$$

For a given job $i \in J$, the value of $Pos(C_i(\pi) > D_i)$ denotes the possibility of a tardy completion, i.e. the possibility of the event that the completion time of i will exceed the due date D_i. Similarly, the value of $Nec(C_i(\pi) > D_i)$ denotes the necessity of a tardy completion. In problems PS1 and PS2 the greatest weighted possibility (necessity) of tardy completion is minimized, while in problems PS3 and PS4 the sum of weighted possibilities (necessities) of tardy completions is minimized. Consider now the following sequencing problem:

$$\text{PS5: } \min_{\pi \in \Pi} \max_{i \in J} \{w_i Nec(C_i(\pi) \neq D_i)\}.$$

Problem PS5 is different from problems PS1–PS4. In PS5 all the jobs should be completed as close to their due dates as possible and the greatest weighted necessity of both tardy and early completion is minimized. Thus problem PS5 belongs to the class of *just in time* sequencing problems.

Let us illustrate problems PS1–PS5 in a simple example. Assume that $J = \{1, 2, 3\}$ with $1 \to 2$, thus J consists of three jobs and job 1 must precede job 2 in every feasible sequence. The set Π consists of three feasible sequences, that is $\Pi = \{(1, 2, 3), (1, 3, 2), (3, 1, 2)\}$. The processing times and the due dates are given as trapezoidal fuzzy numbers: $P_1 = (2, 2, 1, 4)$, $P_2 = (4, 5, 3, 3)$, $P_3 = (5, 6, 2, 5)$, $D_1 = (6, 6, 5, 3)$, $D_2 = (8, 8, 2, 3)$, $D_3 = (5, 7, 2, 2)$. Weights are given as positive crisp numbers: $w_1 = 2$, $w_2 = 1$, $w_3 = 3$. Consider the feasible sequence $\pi = (1, 2, 3)$. The completion times of all the jobs in π can be calculated by means of formula (1), thus $C_1(\pi) = (2, 2, 1, 4)$, $C_2(\pi) = (6, 7, 4, 7)$, $C_3(\pi) = (11, 13, 6, 12)$. The corresponding values of the indices for π are presented in Table 1. Using the values presented in Table 1 it is easy to evaluate the sequence π for all of problems PS1–PS5. In the same way the other feasible sequences, that is (1, 3, 2) and (3, 1, 2), can be evaluated. One can easily verify

Table 1
The values of the indices for $\pi = (1, 2, 3)$

i	$Pos(C_i(\pi) > D_i)$	$Nec(C_i(\pi) > D_i)$	$Nec(C_i(\pi) \neq D_i)$
1	0	0	0.44
2	0.6	0	0.11
3	1	0.5	0.5

that sequence $(1, 2, 3)$ is optimal to PS3, sequence $(1, 3, 2)$ is optimal to PS4 and sequence $(3, 1, 2)$ is optimal to PS1, PS2 and PS5.

The aim of this paper is to explore the computational complexity of problems PS1–PS5. In the next sections we will show that PS1 and PS2 can be solved in polynomial time, while PS3, PS4 and PS5 are $\mathcal{NP}$-hard even if all the weights are equal to 1 and there are no precedence constraints between jobs.

4. A polynomial algorithm for problems PS1 and PS2

An algorithm for solving PS1 and PS2 was constructed in [3], so in this paper we only recall its formulation. In order to simplify calculations we assume that $P_i = (\underline{p}_i, \overline{p}_i, \alpha_i, \beta_i)$ and $D_i = (\underline{d}_i, \overline{d}_i, \gamma_i, \delta_i)$, $i \in J$, are nonnegative trapezoidal fuzzy numbers. Then, the algorithm for solving problem PS1 is presented in the form of Algorithm 1.

Algorithm 1. The algorithm for solving PS1

Require: $n, (P_i)_{i=1}^n, (D_i)_{i=1}^n, prec$
Ensure: π

1. $\mathbf{S} \leftarrow \{1, \ldots, n\}$
2. **for** $k \leftarrow n$ **downto** 1 **do**
3. $\quad T \leftarrow \left(\sum_{i\in\mathbf{S}} \underline{p}_i, \sum_{i\in\mathbf{S}} \overline{p}_i, \sum_{i\in\mathbf{S}} \alpha_i, \sum_{i\in\mathbf{S}} \beta_i\right)$
4. $\quad$ Find $j \in \mathbf{S}$ which has no successor in $\mathbf{S}$ and has a minimal value $w_j Pos(T > D_j)$
5. $\quad \pi(k) \leftarrow j$
6. $\quad \mathbf{S} \leftarrow \mathbf{S} \setminus \{j\}$
7. **end for**
8. **return** π

Note that Algorithm 1 is very similar to the well-known Lawlers' algorithm, which is used for solving the classical sequencing Problem $1|prec|f_{max}$ with nondecreasing cost functions (see [1,12]). Algorithm 1 can be viewed as a generalization of Lawlers' algorithm to the fuzzy case. In line 3 of Algorithm 1, the fuzzy number T is a sum of the processing times of all jobs belonging to **S**. Since all the processing times are trapezoidal fuzzy numbers, T is also a trapezoidal fuzzy number and its value can be calculated by means of formula (1). In line 4 the value of $Pos(T > D_j)$, $j \in J$, can be calculated by means of formula (6). The computational complexity of Algorithm 1 is $O(n^2)$. Algorithm 1 can be easily transformed to the algorithm for solving problem PS2. It is enough to replace expression $w_j Pos(T > D_j)$ in line 4 with expression $w_j Nec(T > D_j)$. The value of $Nec(T > D_j)$, $j \in J$, can be calculated by means of formula (7). Thus, both problems PS1 and PS2 can be solved in $O(n^2)$ time.

5. The complexity of problems PS3 and PS4

In this section we explore the computational complexity of problems PS3 and PS4. We will show that these problems are much more hard to solve than problems PS1 and PS2 presented in the previous section. Consider first the special cases of PS3 and PS4, in which all the processing times and all the due dates are crisp numbers. Then, it is easy to observe that the completion times of all the jobs in a given sequence π are also crisp numbers and for each job $i \in J$ it holds

$$Pos(C_i(\pi) > D_i) = Nec(C_i(\pi) > D_i) = U_i(\pi) = \begin{cases} 1 & \text{if } C_i(\pi) > D_i, \\ 0 & \text{if } C_i(\pi) \leqslant D_i. \end{cases} \tag{12}$$

From (12) it follows that:

$$\sum_{i \in J} w_i Pos(C_i(\pi) > D_i) = \sum_{i \in J} w_i Nec(C_i(\pi) > D_i) = \sum_{i \in J} w_i U_i(\pi). \tag{13}$$

Note that (13) is a weighted number of late jobs in π so, if all the parameters are crisp numbers, then both problems PS3 and PS4 are equivalent to the classical sequencing problem $1|prec|\sum w_i U_i$. Since problem $1|prec|\sum w_i U_i$ is strongly $\mathcal{NP}$-hard [1] it follows that the more general problems PS3 and PS4 are also strongly $\mathcal{NP}$-hard. Moreover, problems $1|prec|\sum U_i$ and $1||\sum w_i U_i$ are also $\mathcal{NP}$-hard [1], so PS3 and PS4 remain $\mathcal{NP}$-hard even if there are no precedence constraints between jobs or all the weights are equal to 1. But, if we assume that there are no precedence constraints and all the weights are equal to 1 in problem $1|prec|\sum w_i U_i$, then we get problem $1||\sum U_i$, which can be solved in $O(n \log n)$ time by Moor's algorithm [14]. Thus, it is interesting to explore the complexity of the generalization of $1||\sum U_i$. Let PS3′ and PS4′ be the special cases of problems PS3 and PS4, respectively, in which there are no precedence constraints (i.e. $prec = \emptyset$) and all the weights are equal to 1 (i.e. $w_i = 1, i \in J$). Note that PS3′ and PS4′ can be solved in polynomial time if all the parameters are crisp numbers since, in this case, they are equivalent to $1||\sum U_i$. The following theorem holds:

Theorem 1. *Problem PS3′ is $\mathcal{NP}$-hard even if all the processing times are crisp numbers.*

Proof. We show that the classical, $\mathcal{NP}$-hard sequencing problem $1||\sum T_i$ is polynomially reducible to PS3′. Let us recall that an instance of $1||\sum T_i$ consists a set of jobs $J = \{1, \dots, n\}$, positive processing times p_i and positive due dates d_i given for all jobs $i \in J$. It is assumed that there are no precedence constraints between jobs. Let $T_i(\pi) = \max(0, C_i(\pi) - d_i)$ denote the tardiness of job $i \in J$ in a given sequence π. The objective is to calculate a sequence π for which the value of $\sum_{i \in J} T_i(\pi)$ is minimal. Problem $1||\sum T_i$ is $\mathcal{NP}$-hard in the ordinary sense [5]. Let $I = (n, (p_i)_{i=1}^n, (d_i)_{i=1}^n)$ be a given instance of $1||\sum T_i$. Let us define $K = \sum_{i=1}^n p_i$. The corresponding instance I' of problem PS3′ is constructed as follows:

- $J = \{1, \dots, n\}$,
- $P_i = p_i, i = 1, \dots, n$,
- $D_i = (d_i, d_i, 1, K), i = 1, \dots, n$.

Let π be a given sequence of jobs. Since all the processing times P_i, $i = 1, \dots, n$, are crisp numbers it follows that all the completion times $C_i(\pi)$, $i = 1, \dots, n$, are also crisp numbers. For each job $i \in J$ it holds (see formula (9))

$$Pos(C_i(\pi) > D_i) = \max\left(0, \min\left(1, \frac{C_i(\pi) - d_i}{K}\right)\right). \tag{14}$$

Since $C_i(\pi) \leqslant K$ and $d_i > 0$, it holds $(C_i(\pi) - d_i)/K < 1$ and we can rewrite (14) as follows:

$$Pos(C_i(\pi) > D_i) = \max\left(0, \frac{C_i(\pi) - d_i}{K}\right) = \frac{1}{K}\max(0, C_i(\pi) - d_i).$$

It holds

$$\sum_{i=1}^{n} Pos(C_i(\pi) > D_i) = \frac{1}{K}\sum_{i=1}^{n}\max(0, C_i(\pi) - d_i) = \frac{1}{K}\sum_{i=1}^{n} T_i(\pi). \tag{15}$$

Equality (15) implies that the optimal solution to $1||\sum T_i$ for instance I is the same as the optimal solution to PS3′ for instance I' (note that K is a constant, whose value does not depend on the solution). It is clear that instance I' can be obtained from I in polynomial time. This means that having a polynomial algorithm for problem PS3′ we would be able to solve the $\mathcal{NP}$-hard problem $1||\sum T_i$ in polynomial time. This means that problem PS3′ is $\mathcal{NP}$-hard. □

Theorem 2. *Problem PS4′ is $\mathcal{NP}$-hard even if all the processing times are crisp numbers.*

Proof. If all the processing times are crisp numbers, then from (9) we get $Pos(C_i(\pi) > D_i) = Nec(C_i(\pi) > D_i)$, $i \in J$, so problem PS4′ is equivalent to PS3′ in this case. This means that problem PS4′ is also $\mathcal{NP}$-hard. □

6. The complexity of problem PS5

In this section we will prove that problem PS5 is $\mathcal{NP}$-hard. Let us start by observing that PS5 is a min–max problem so it is similar to problems PS1 and PS2 considered in Section 4. Despite this similarity, problem PS5 cannot be solved by means of Algorithm 1. The reason is as follows: if A is a nonnegative fuzzy number then the value of $Nec(C_i(\pi) + A \neq D_i)$, $i \in J$, can be less than the value of $Nec(C_i(\pi) \neq D_i)$. In other words, increasing the completion time of a job by a nonnegative fuzzy number A may result in decreasing of the value of the cost function. If such a situation may take place, then Algorithm 1 cannot be used (see [3]).

Consider a special case of problem PS5 in which there are no precedence constraints between jobs and all the weights are equal to 1. Let us denote such a problem by PS5′.

Theorem 3. *Problem PS5′ is $\mathcal{NP}$-hard.*

Proof. We shall prove the $\mathcal{NP}$-hardness of PS5′ by a reduction from the following problem:

PARTITION

Instance: Collection $A = (a_1, \ldots, a_n)$ of positive integers.

Question: Is there a subset $Q \subset \{1, \ldots, n\}$ such that $\sum_{i \in Q} a_i = \frac{1}{2}\sum_{i=1}^{n} a_i$?

Partition is known to be $\mathcal{NP}$-complete in the ordinary sense [8]. We give a polynomial time reduction from this problem to PS5′ such that a proper subset Q exists if and only if there exists a sequence with the cost less or equal to 0. Let $A = (a_1, \ldots, a_n)$ be a given instance of Partition. Let us define $S = \frac{1}{2}\sum_{i=1}^{n} a_i$. The corresponding instance of PS5′ is constructed as follows:

- $J = \{1, \ldots, n, n+1\}$,
- $P_i = (a_i, a_i, 1, 1)$, $i = 1, \ldots, n$,
- $P_{n+1} = (a_{n+1}, a_{n+1}, 1, 1) = (1, 1, 1, 1)$,
- $D_i = (1, 2S+1, 1, 1)$, $i = 1, \ldots, n$,
- $D_{n+1} = (S+1, S+1, 1, 1)$.

Note that all the parameters P_i, D_i, $i = 1, \ldots, n+1$, are nonnegative, trapezoidal fuzzy numbers. Let π be a given sequence of jobs. Let us denote by π^i, $i \in J$, the set of all the jobs processed before job i in the sequence π. Using formula (1) we obtain

$$C_i(\pi) = \left(\sum_{j \in \pi^i} a_j + a_i, \sum_{j \in \pi^i} a_j + a_i, |\pi^i| + 1, |\pi^i| + 1 \right), \; i = 1, \ldots, n+1,$$

where $|\pi^i|$ denotes the number of jobs processed before job i in π. It is easy to notice that for each job $i = 1, \ldots, n$ it holds

$$\left[\sum_{j \in \pi^i} a_j + a_i, \sum_{j \in \pi^i} a_j + a_i \right] \cap [1, 2S+1] \neq \emptyset. \tag{16}$$

Thus, from Property 1 we conclude that for each sequence π it holds

$$Nec(C_i(\pi) \neq D_i) \leqslant 0, \; i = 1, \ldots, n. \tag{17}$$

Consider now job $n+1$. From Property 1 and equality $P_{n+1} = (1, 1, 1, 1)$ we obtain

$$Nec(C_{n+1}(\pi) \neq D_{n+1}) \leqslant 0 \Leftrightarrow \left[\sum_{j \in \pi^{n+1}} a_j + 1, \sum_{j \in \pi^{n+1}} a_j + 1 \right] \cap [S+1, S+1] \neq \emptyset,$$

which is equivalent to the following condition:

$$Nec(C_{n+1}(\pi) \neq D_{n+1}) \leqslant 0 \Leftrightarrow \sum_{j \in \pi^{n+1}} a_j = S. \tag{18}$$

From (17) and (18) we conclude that

$$F(\pi) = \max_{i \in J}\{Nec(C_i(\pi) \neq D_i)\} \leqslant 0 \Leftrightarrow \sum_{j \in \pi^{n+1}} a_j = S. \tag{19}$$

Now we shall prove that the answer to Partition is yes if and only if there exists sequence π such that $F(\pi) \leqslant 0$.

Assume that the answer to Partition is yes, i.e. there exists a subset Q such that $\sum_{i \in Q} a_i = S$. Let σ denote any permutation of the set Q and ρ any permutation of the set $\{1, \ldots, n\} \setminus Q$. Consider the sequence $\pi = (\sigma, n+1, \rho)$. It holds $\sum_{j \in \pi^{n+1}} a_j = S$, so from (19), we conclude that $F(\pi) \leqslant 0$.

Assume that there exists a sequence π such that $F(\pi) \leqslant 0$. Consider job $n+1$ processed in π. From (19) it follows that $\sum_{j \in \pi^{n+1}} a_j = S$. Let $Q = \pi^{n+1}$, i.e. Q contains all the jobs processed before $n+1$ in π. It is clear that Q is the subset of $\{1, \dots, n\}$ for which the partition holds and the answer to PARTITION is yes. □

From Theorem 3 we get at once that the more general problem PS5 is $\mathcal{NP}$-hard.

7. Conclusions

In this paper a possibilistic approach to sequencing problems with fuzzy parameters is proposed. In this approach for each parameter, whose value is not precisely known, a possibility distribution is given. In order to evaluate the solutions the indices proposed by Dubois and Prade are applied. Five, different problems are formulated and the computational complexity of all of them is explored. It turns out that two problems can be solved in polynomial time, while the remaining three problems are $\mathcal{NP}$-hard, even in some restrictive cases. For the $\mathcal{NP}$-hard problems PS3–PS5 some heuristics or approximation algorithms should be constructed.

References

[1] P. Brucker, Scheduling Algorithms, Second Revised and Enlarged Edition, Springer, Berlin, 1998.
[2] M. Hapke, R. Slowinski (Eds.), Scheduling under Fuziness, Physica-Verlag, Heidelberg, 2000.
[3] S. Chanas, A. Kasperski, Minimizing maximum lateness in a single machine scheduling problem with fuzzy processing times and fuzzy due dates, Eng. Appl. Artif. Intell. 14 (2001) 377–386.
[4] S. Chanas, A. Kasperski, Possible and necessary optimality of solutions in the single machine scheduling problem with fuzzy parameters, Fuzzy Sets and Systems 142 (2004) 359–371.
[5] J. Du, J.Y.-T. Leung, Minimizing total tardiness on one machine is NP-hard, Math. Oper. Res. 15 (1990) 483–495.
[6] D. Dubois, H. Prade, Possibility Theory: An Approach to Computerized Processing of Uncertainty, Plenum Press, New York, 1988.
[7] D. Dubois, H. Prade, The use of fuzzy numbers in decision analysis, Fuzzy Information and Decision Processes, North-Holland Publishing Company, 1982, pp. 309–321.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability—A Guide to the Theory of NP-Completness, Amsterdam, Freeman, New York, 1979.
[9] S. Han, H. Ishii, S. Fujii, One machine scheduling problem with fuzzy duedates, European J. Oper. Res. 79 (1994) 1–12.
[10] H. Ishii, M. Tanaka, T. Masuda, Two scheduling problems with fuzzy due-dates, Fuzzy Sets and Systems 46 (1992) 339–347.
[11] T. Itoh, H. Ishii, Fuzzy due-date scheduling problem with fuzzy processing time, Internat. Trans. Oper. Res. 6 (1999) 639–647.
[12] E.L. Lawler, Optimal sequencing of a single machine subject to precedence constraints, Management Sci. 19 (1973) 544–546.
[13] K. Muthusamy, S.C. Sung, M. Vlach, H. Ishii, Scheduling with fuzzy delays and fuzzy precedences, Fuzzy Sets and Systems 134 (2003) 387–395.
[14] J.M. Moore, An n job, one machine sequencing algorithm for minimzing the number of late jobs, Management Sci. 15 (1968) 102–109.
[15] S.C. Sung, M. Vlach, Single machine scheduling to minimize the number of late jobs under uncertainty, Fuzzy Sets and Systems 139 (2003) 421–430.
[16] K. Tanaka, M. Vlach, Single machine scheduling with fuzzy due dates, Proc. Seventh IFSA World Congress, Prague, 1997, pp. 195–199.
[17] C. Wang, D. Wang, W.H. Ip, D.W. Yuen, The single machine ready time scheduling problem with fuzzy processing times, Fuzzy Sets and Systems 127 (2002) 117–129.