Possibilistic Minmax Regret Sequencing Problems with Fuzzy Parameters

Adam Kasperski and Paweł Zieliński

***Abstract*—In this paper a class of sequencing problems with uncertain parameters is discussed. The uncertainty is modeled by using fuzzy intervals, whose membership functions are regarded as possibility distributions for the values of unknown parameters. It is shown how to use possibility theory to find robust solutions under fuzzy parameters - this paper presents a general framework together with applications to some classical sequencing problems. First, the interval sequencing problems with the minmax regret criterion are discussed. The state of the art in this area is recalled. Next, the fuzzy sequencing problems, in which the classical intervals are replaced with fuzzy ones, are investigated. A possibilistic interpretation of such problems, solution concepts, and algorithms for computing a solution are described. In particular, it is shown that every fuzzy problem can be efficiently solved if a polynomial algorithm for the corresponding interval problem with the minmax regret criterion is known. Some methods of dealing with NP-hard problems are also proposed and the efficiency of these methods is explored.**

***Index Terms*—Sequencing, minmax regret, possibility theory, fuzzy interval, fuzzy optimization.**

I. Introduction

In a sequencing problem we wish to find a feasible order of elements, called jobs, to achieve some goal. This goal typically depends on job completion times and may also depend on some other job parameters such as due dates or weights. There are a lot of deterministic sequencing problems with different computational properties and a comprehensive description of them can be found, for example, in [1]. Unfortunately, most of sequencing problems turned out to be NP-hard, but there are also some important problems for which efficient polynomial algorithms exist.

Sequencing problems involve many parameters whose exact values are often ill-known. For instance, a job processing time, which is the crucial parameter in all sequencing problems, is seldom precisely known. Also such job parameters as due dates or weights may be ill-known. This difficulty has been noticed very early in [2], where the author proposed to model uncertain job processing times by probability distributions and tried to minimize the expected sum of weighted completion times of jobs. Since then, there has been an extensive literature on *stochastic scheduling*, i.e. scheduling problems, in which uncertain parameters are modeled as random variables (see [3] for a survey of recent results and [4] for a bibliography). In the overwhelming part of the stochastic scheduling literature, it is assumed that the probability distributions describing uncertain parameters are known in advance. Usually, special classes of distributions such as exponential or Gaussian are applied to model the uncertainty of parameters and typically the expected cost of a solution is minimized (see, e.g., [5]). Unfortunately, most of the stochastic scheduling problems are at least NP-hard [6], [7] (#P-hard) and they are tractable only when some assumptions are imposed. Another difficulty, not always pointed out, is the possible lack of statistical data validating the choice of the parameter distributions. In fact, the probability distributions permit us to model the variability of repetitive parameters, but this approach becomes debatable when dealing with the uncertainty caused by a lack of information [8], [9]. Even if statistical data are available, they may be partially inadequate, because each problem may take place in a specific environment, and is not the exact replica of the past ones.

In practice, modeling the uncertainty of parameters in the form of intervals is natural and simple - a decision maker just needs to provide a minimal value of a parameter and a maximal one. Moreover, allowing a collection of the minimal and maximal values may be felt more realistic, if the elicitation process forces the decision maker to provide as narrow intervals as possible. Assigning some interval to a parameter means that it will take some value within the interval, but it is not possible to predict at present which one. In this paper, each precise instantiation of the parameter values will be called a *scenario* (it is also called a *configuration* or a *possible world* in formal logic). So a *scenario set*, the set containing all the possible realizations of the parameters which may occur, is the Cartesian product of all the uncertainty intervals. No probability distribution over the scenario set is given. A natural criterion for choosing a solution under this interval uncertainty is the *maximal regret*, which expresses the maximum “distance” of a solution from optimality over all scenarios [10]. A deeper discussion on using the maximal regret criterion in decision making under uncertainty can be found in [10], [11].

The maximal regret criterion has been applied to many interval versions of basic combinatorial optimization problems such as: the minimum spanning tree, the shortest path and the minimum assignment (see [12] for a survey). It has been also successfully applied to energy [13], water [14] and waste [15], [16] management. The class of minmax regret sequencing problems with interval parameters has been discussed in a number of papers, for example in [17], [18], [19], [20], [21], [22], [23], [24]. We describe the known results in this area in

Adam Kasperski is with the Institute of Industrial Engineering and Management, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland (email: adam.kasperski@pwr.wroc.pl).

Paweł Zieliński is with the Institute of Mathematics and Computer Science, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland (email: pawel.zielinski@pwr.wroc.pl).

This work was partially supported by Polish Committee for Scientific Research, grant N N206 492938.

Section III-A, because they form a basis of the more general framework for dealing with sequencing problems with ill-known parameters presented in this paper.

A decision maker may find it difficult to provide narrowed intervals for parameters (because she/he may be wrong). The interval uncertainty may also be considered as poorly expressive. So we do not propose the use of intervals as the final answer to scheduling under uncertainty. A more elaborate approach could be to collect both the intervals and plausible values from decision makers and, in this case, fuzzy intervals may be useful. Resorting to fuzzy sets and possibility theory [25] for modelling ill-known parameters, the model considered in this paper may help building a trade-off between the lack of expressive power of mere intervals and the computational difficulties of the stochastic scheduling techniques. In recent years, the application of fuzzy sets to modeling the imprecision in optimization has attracted a considerable attention. A good review of different concepts in fuzzy optimization can be found in [26]. In papers [27], [28], [29] some single machine sequencing problems with fuzzy processing times, fuzzy due dates and fuzzy precedence constraints have been discussed. In these papers, a fuzzy due date expresses a degree of satisfaction with a job completion time and a sequence is computed which maximizes the minimum satisfaction or the sum of satisfactions over all jobs. In [30], the model of uncertainty is the same as the one considered in this paper, but the assumed solution concepts are different. Namely, the possibility or necessity of job delays is minimized. In [31] an optimality evaluation of sequences under fuzzy parameters has been investigated and this approach is extended here.

In this paper we propose a general framework for dealing with sequencing problems with uncertain parameters. We generalize the minmax regret approach to the fuzzy case by extending the interval uncertainty representation to the fuzzy interval one. Fuzzy parameters induce a possibility distribution over the scenario set, which becomes then richer in information. We provide a possibilistic interpretation of the fuzzy problem obtained and describe a solution concept being an adaptation of the elegant one, originally proposed in [32], [33] for fuzzy linear programming. This solution concept has been recently adopted in [34] for a class of combinatorial optimization problems. Apart from showing a general framework, we also point out some difficulties which arise when one tries to solve a particular problem. Contrary to the class of combinatorial optimization problems described in [34], the sequencing problems are typically harder to solve and there are very few general properties which are valid for all these problems. However, as for the problems described in [34], the main computational difficulties of sequencing problems under uncertainty are in the interval case and the algorithms known for the interval uncertainty representation can be generalized to the fuzzy case by using a binary search. This method is efficient if the corresponding minmax regret sequencing problem is polynomially solvable. If the minmax regret sequencing problem is NP-hard, then we propose a mixed integer programing formulation, which can be solved by using some available software. For some particular problems, a parametric approach can also be applied.

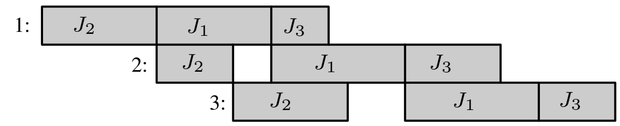

Fig. 1. A permutation flow shop problem with three machines and a schedule that corresponds to the sequence $(2,1,3)$.

We present several applications of the proposed framework to some classical sequencing problems with fuzzy parameters. Namely, we discuss single-machine sequencing problems with the objective function of maximum lateness, maximum weighted tardiness, total flow time and weighted number of late jobs. We also consider the two-machine flow shop problem. Hence we provide a set of efficient algorithms for these classical sequencing problems with fuzzy parameters.

This paper is organized as follows. In Section II we recall a definition of the classical deterministic sequencing problem. In Section III we present the minmax regret approach to sequencing problems with interval parameters - we provide a general formulation and recall some known results in this area. In Section IV we introduce a class of sequencing problems with fuzzy parameters. We give a link between the fuzzy problems and the minmax regret ones. We present a general framework and show some methods of solving the fuzzy problems. Finally, in Section V we discuss several applications of the proposed framework to the classical sequencing problems with fuzzy parameters.

II. Deterministic Sequencing Problems

We are given a set of *jobs* $J = \{J_1, \ldots, J_n\}$, which may be partially ordered by some *precedence constraints* of the form $i \to j$, where $i, j \in J$. For the simplicity of notation we will identify each job $J_i \in J$ with its index $i \in \{1, \ldots, n\}$. A solution is a *sequence* (permutation) $\sigma = (\sigma(1), \ldots, \sigma(n))$ of J and it represents an order in which the jobs are processed. A sequence σ is *feasible* if $i \to j$ implies that job j occupies position after i in σ. We will denote by $\mathbf{\Omega}$ the set of all feasible sequences. We will use $C_i(\sigma)$ to denote the *completion time* of job i in sequence σ. In a *single machine* case, a processing time p_i is given for each job $i \in J$ and if $i = \sigma(k)$, then $C_i(\sigma) = \sum_{j=1}^{k} p_{\sigma(j)}$. If every job must be processed on $m > 1$ machines, first on machine 1, next on machine 2 and so on, then we get a *permutation flow shop problem*. In this case p_{ij} is a processing time of job i on machine j and $C_i(\sigma)$ is the time when job i is finished on the m-th machine (see Fig. 1).

For each job $i \in J$, there is a function $f_i : \mathbb{R} \to \mathbb{R}$, which measures the cost of completing i at time t. The popular cost functions are *lateness*, $L_i(t) = t - d_i$, *tardiness* $T_i(t) = \max\{0, t - d_i\}$ and *unit penalty* $U_i(t) = 1$ if $t > d_i$ and $U_i(t) = 0$ otherwise. One can also consider the weighted versions of these functions, namely $w_i L_i(t)$, $w_i T_i(t)$ and $w_i U_i(t)$. Finally, $F(\sigma)$ denotes a cost of sequence σ. There are two general types of this cost function, namely a *bottleneck cost function* $F(\sigma) = \max_{i=1,\ldots,n} f_i(C_i(\sigma))$ and a *sum cost function* $F(\sigma) = \sum_{i=1}^{n} f_i(C_i(\sigma))$.

In a deterministic sequencing problem we seek a feasible sequence with the minimum cost, that is we wish to solve the following optimization problem:

$$\min_{\sigma\in\mathbf{\Omega}} F(\sigma). \tag{1}$$

Our analysis in the next sections of this paper will be based on the concept of a *deviation*. In the deterministic case the deviation of a sequence σ expresses its "distance" to the optimum and it is defined as follows:

$$\delta_\sigma = F(\sigma) - \min_{\rho\in\mathbf{\Omega}} F(\rho). \tag{2}$$

Obviously, σ is optimal if and only if $\delta_\sigma = 0$. If the deviation is positive, then it measures a "distance" of σ from optimality.

Sequencing problems are usually described by using the convenient Graham's notation (see, e.g., [1]). Namely, every sequencing problem can be denoted by a triple $\alpha|\beta|\gamma$, where α is the machine environment ($\alpha = 1$ for the single machine case), β specifies the job characteristic and γ describes the objective function. The following sequencing problems are important from both theoretical and practical point of view (see also [1]):

- $1|prec|L_{\max}$. In this problem we seek a feasible sequence, which minimizes the maximum *lateness*. The problem can be solved in polynomial time by using the well known Lawler's algorithm [35]. Furthermore, if there are no precedence constraints between jobs, then an optimal sequence can be obtained by ordering jobs with respect to nondecreasing due dates.
- $1|prec|\max w_iT_i$. In this problem wish to find a feasible sequence minimizing the maximum weighted *tardiness*. This problem can also be solved in polynomial time by Lawler's algorithm.
- $1||\sum C_i$. In this problem there are no precedence constraints between jobs and we seek a sequence for which the sum of completion times of all jobs, i.e. the *total flow time*, is minimal. An optimal sequence can be easily obtained by ordering jobs with respect to nondecreasing processing times (see, e.g., [1]).
- $1|p_i = 1|\sum w_iU_i$. In this problem there are no precedence constraints between jobs and all the jobs have unit processing times. The cost of σ is the weighted number of late jobs. This problem can be solved in polynomial time by a greedy algorithm [36].
- $Fm||C_{\max}$. This is the permutation flow shop problem with $m > 1$ machines. There are no precedence constraints between jobs and the cost of σ is the completion time of the last job on the last machine. This problem is polynomially solvable only when $m = 2$ by Johnson's algorithm (see, e.g., [1]) and becomes strongly NP-hard for $m \geq 3$ (see [37]).

The above examples illustrate a large variety of basic sequencing models. As we will see in the next section, they have quite different computational properties under uncertainty.

III. Interval Sequencing Problems

In practice, the exact values of the parameters in a sequencing problem such as processing times, due dates or weights may be not precisely known. Assume that we have l parameters and the value of a parameter ξ_i, $i = 1,\dots,l$, may fall within a closed interval $[\underline{\xi}_i, \overline{\xi}_i]$ independently of the values of the other parameters, but it is not possible to predict at present which value from $[\underline{\xi}_i, \overline{\xi}_i]$ the parameter takes. We call a parameter ξ_i *precise* if $\underline{\xi}_i = \overline{\xi}_i$. Every vector $S = (s_1,\dots,s_l) \in \mathbb{R}^l$ such that $s_i \in [\underline{\xi}_i, \overline{\xi}_i]$ is called a *scenario* and it expresses a possible state of the world, where $\xi_i = s_i$ for $i = 1,\dots,l$. A scenario is called *extreme* if all the parameters take the lower or upper bounds in their uncertainty intervals. We use Γ to denote the set of all the possible scenarios. Hence Γ is the Cartesian product of all the uncertainty intervals. Now, the cost of a sequence σ depends on scenario $S \in \Gamma$ and we will denote it by $F(\sigma, S)$. We will also denote by $F^*(S)$ the cost of an optimal sequence under scenario S. In order to obtain the value of $F^*(S)$ we need to solve problem (1) under the fixed realization of parameters S. It is clear that the deviation of σ also depends on scenario S and we will denote it by $\delta_\sigma(S) = F(\sigma, S) - F^*(S)$.

Now the optimality of a sequence σ can be characterized by a *deviation interval* $[\underline{\delta}_\sigma, \overline{\delta}_\sigma]$, where

$$\underline{\delta}_\sigma = \min_{S\in\Gamma} \delta_\sigma(S) = \min_{S\in\Gamma}\{F(\sigma,S) - F^*(S)\}, \tag{3}$$

$$\overline{\delta}_\sigma = \max_{S\in\Gamma} \delta_\sigma(S) = \max_{S\in\Gamma}\{F(\sigma,S) - F^*(S)\}. \tag{4}$$

The lower bound $\underline{\delta}_\sigma$ is the minimal deviation and the upper bound $\overline{\delta}_\sigma$ is the maximal deviation of σ over the set of scenarios Γ. In the existing literature, the quantity $\overline{\delta}_\sigma$ is called the *maximal regret* of σ (see, e.g. [10]) and it expresses the largest "distance" of σ from the optimum over the scenario set Γ. A scenario S_σ, for which the deviation of σ attains maximum, is called the *worst case scenario* for σ. So, under the interval uncertainty representation, we only know that $\delta_\sigma \in [\underline{\delta}_\sigma, \overline{\delta}_\sigma]$ and we can give the following characterization of a feasible sequence: a sequence σ is *possibly optimal* if $\underline{\delta}_\sigma = 0$ and it is *necessarily optimal* if $\overline{\delta}_\sigma = 0$. Notice that a sequence is possibly optimal if and only if it is optimal for some scenario $S \in \Gamma$ and it is necessarily optimal if and only if it is optimal for all scenarios $S \in \Gamma$.

Now the question arises, which sequence of $\mathbf{\Omega}$ should be chosen. A necessarily optimal one is a natural choice. However, it may rarely exist in most practical situations, because the necessary optimality is very strong criterion. A set of potential solutions can be characterized by the concept of a *minimal dominant set* [38], [39], that is a minimal under inclusion set $\mathbf{\Omega}^D \subseteq \mathbf{\Omega}$ of sequences, which contains an optimal sequence under each scenario $S \in \Gamma$. It is easy to see that the minimal dominant set is a subset of the set of all possibly optimal sequences. Furthermore, $\mathbf{\Omega}^D$ contains only one sequence σ if and only if σ is necessarily optimal. The cardinality of $\mathbf{\Omega}^D$ can be seen as a measure of uncertainty in a sequencing problem with interval data and, in the extreme case, $\mathbf{\Omega}^D = \mathbf{\Omega}$.

In this paper, in order to choose a sequence under the interval structure of uncertainty, we will consider the following *minmax regret sequencing problem* (or shortly *interval prob-*

lem):

$$\min_{\sigma\in\mathbf{\Omega}}\overline{\delta}_{\sigma}=\min_{\sigma\in\mathbf{\Omega}}\max_{S\in\Gamma}\delta_{\sigma}(S). \tag{5}$$

We call an optimal solution to (5) a *minmax regret sequence*. In the next section we briefly recall some known facts on problem (5) and its particular cases.

A. Complexity of Minmax Regret Sequencing Problems

The first problem arising while analyzing a particular minmax regret sequencing problem is the computation of the maximal regret of a given sequence σ, that is the quantity $\overline{\delta}_{\sigma}$ (see (4)). Unfortunately, contrary to the class of problems discussed in [34], there is no general method of performing this task. For the problems considered in [34], it is possible to find two extreme scenarios that maximize or minimize the deviation and, consequently, the computation of the maximal regret has the same complexity as the deterministic problem. For sequencing problems the situation is much more complex. First of all, for some problems there may be no extreme scenario that maximizes (minimizes) the deviation [40]. Furthermore, computing the maximal regret may be much more time consuming than solving a deterministic problem. For instance, in the minmax regret $1||\sum C_i$ problem with interval processing times, computing $\overline{\delta}_{\sigma}$ requires solving an assignment problem, while an optimal sequence under a given scenario can be computed in $O(n\log n)$ time [10], [17]. An extreme case has been described in [21], where a permutation flow shop problem with m machines, interval processing times, and with only 2 jobs has been discussed. Since the solution set contains only two sequences, the total computational effort is focused on computing the maximal regret of a given sequence. Also, for the class of combinatorial problems discussed in [34], it is easy to show that every minmax regret solution is possibly optimal. No such general property is known for sequencing problems. However, this is the case for some particular problems, for example $1||\sum C_i$ with interval processing times [10].

It is not surprising that the minmax regret sequencing problems are typically hard to solve. There are only several problems which are known to be polynomially solvable. A polynomial algorithm for $1|prec|L_{\max}$ with interval processing times and interval due dates has been constructed in [18]. A polynomial algorithm for $1|prec|\max w_iT_i$ with interval weights, precise processing times and precise due dates has been proposed in [20] and extended to interval processing times and interval due dates in [41]. Apart from these two problems, only some very special cases are known to be polynomially solvable, for instance $Fm||C_{\max}$ with interval processing times and only two jobs [21] and $1|p_i=1|\sum w_iU_i$ with interval weights and a precise common due date ($d_1=d_2=\cdots=d_n$) [40]. This latter problem is equivalent to the minmax regret version of the selecting items problem, which is known to be polynomially solvable [40], [42].

Among the negative results known to date, the most important one has been obtained in [22], where it has been shown that the minmax regret $1||\sum C_i$ problem with interval processing times is NP-hard. This problem is also known to be approximable within a ratio of 2 [24] and can be solved by using a mixed integer programming formulation proposed in [23]. However, the complexity status of a number of basic problems is still unknown. One does not know whether $F2||C_{\max}$ with interval processing times is NP-hard. One can, however, compute in polynomial time the maximal regret of a given sequence and solve the problem by using a branch and bound algorithm [19], [10]. Similarly, the problem $1|p_i=1|\sum w_iU_i$ with interval weights and arbitrary precise due dates and the problem $1|prec|\max w_iL_i$ with interval processing times, interval due dates and precise weights are open. Note that, the former problem can be solved by a mixed integer programming formulation [40]. There are also a large number of different sequencing problems whose minmax regret versions have never been investigated and they should be the subject of further research.

IV. Fuzzy Sequencing Problems

In this section we extend the minmax regret approach, presented in the previous section, to the fuzzy case. We model the uncertain parameters by means of fuzzy intervals and apply possibility theory to define the solution concepts. For a comprehensive description of possibility theory we refer the reader to [25].

A. Basic Notions on Possibility Theory

A *fuzzy interval* $\tilde{A}$ is a fuzzy set in $\mathbb{R}$ whose membership function $\mu_{\tilde{A}}$ is normal, quasi concave and upper semicontinuous. It is typically assumed that the support of a fuzzy interval is bounded. The main property of a fuzzy interval is the fact that all its λ-cuts, i.e. the sets $\tilde{A}^{[\lambda]}=\{x:\mu_{\tilde{A}}(x)\geq\lambda\}$, $\lambda\in(0,1]$, are closed intervals. We will assume that $\tilde{A}^{[0]}$ is the smallest closed set containing the support of $\tilde{A}$. So, every fuzzy interval $\tilde{A}$ can be represented as a family of closed intervals $\tilde{A}^{[\lambda]}=[\underline{a}^{[\lambda]},\overline{a}^{[\lambda]}]$, parametrized by the value of $\lambda\in[0,1]$. In many practical applications, the class of *trapezoidal fuzzy intervals* is used. A trapezoidal fuzzy interval, denoted by a quadruple $(\underline{a},\overline{a},\alpha,\beta)$, can be represented as the family of intervals $[\underline{a}-\alpha(1-\lambda),\overline{a}+\beta(1-\lambda)]$ for $\lambda\in[0,1]$. Notice that this representation contains classical intervals ($\alpha=\beta=0$) and real numbers (additionally $\underline{a}=\overline{a}$) as the special cases.

In this paper we adopt a possibilistic interpretation of a fuzzy interval [25]. Assume that for an uncertain real quantity ξ a fuzzy interval with membership function μ_{ξ} is given. This membership function expresses a *possibility distribution* for the values of ξ, namely $\pi_{\xi}=\mu_{\xi}$, which describes the set of more or less plausible, mutually exclusive values of the quantity ξ. It plays a role similar to a probability density, while it encodes a family of probability functions [43]. The value of $\pi_{\xi}(x)$ represents the possibility degree of the assignment $\xi=x$, i.e. $\Pi(\xi=x)=\pi_{\xi}(x)=\mu_{\xi}(x)$, where $\Pi(\xi=x)$ is the possibility of the event that ξ will take the value of x. It is easily seen that each λ-cut $[\underline{\xi}^{[\lambda]},\overline{\xi}^{[\lambda]}]$, $\lambda\in[0,1]$, contains all the values of ξ, whose possibility of occurrence is not less than λ. In particular, the interval $[\underline{\xi}^{[0]},\overline{\xi}^{[0]}]$ contains all the possible values of ξ, while the interval $[\underline{\xi}^{[1]},\overline{\xi}^{[1]}]$, called a *core*, contains the most plausible ones. Some methods of

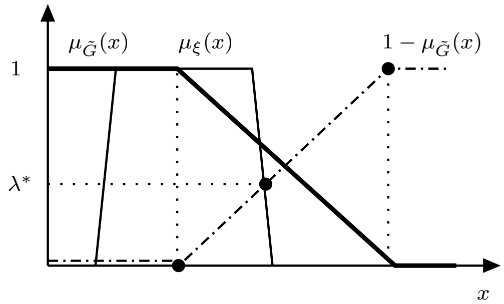

Fig. 2. $\mathrm{N}(\xi \in \tilde{G}) = 1 - \lambda^*$.

obtaining the possibility distribution of an unknown quantity can be found in [25], [44].

Let $\tilde{G}$ be a fuzzy set in $\mathbb{R}$ with membership function $\mu_{\tilde{G}}$. Then $\xi \in \tilde{G}$ is a *fuzzy event* and the necessity that $\xi \in \tilde{G}$ holds is defined in the following way (see, e.g., [45]):

$$\mathrm{N}(\xi \in \tilde{G}) = 1 - \Pi(\xi \notin \tilde{G}) = 1 - \sup_{x \in \mathbb{R}} \min\{\mu_{\xi}(x), 1 - \mu_{\tilde{G}}(x)\} \tag{6}$$

where $1 - \mu_{\tilde{G}}(x)$ is the membership function of the complement of $\tilde{G}$. It is not difficult to see that if $\tilde{G} = (0, \overline{g}, 0, \beta)$, then the following equality is true:

$$\mathrm{N}(\xi \in \tilde{G}) = 1 - \inf\{\lambda \in [0,1] : \overline{\xi}^{[\lambda]} \leq \overline{g}^{[1-\lambda]}\} \tag{7}$$

and $\mathrm{N}(\xi \in \tilde{G}) = 0$ if $\overline{\xi}^{[1]} > \overline{g}^{[0]}$. Equality (7) is illustrated in Fig. 2.

B. Possibilistic Sequencing Problem

Assume that for each ill-known parameter ξ_i, $i = 1, \ldots, l$, in a sequencing problem a fuzzy interval with membership function μ_{ξ_i} is specified. According to the possibilistic interpretation, μ_{ξ_i} is a possibility distribution for the values of ξ_i. As in Section III, $S = (s_1, \ldots, s_l)$ is a scenario denoting a particular realization of the problem parameters. Under the assumption that all the parameters are unrelated, there is a possibility distribution over all scenarios $S = (s_1, \ldots, s_l) \in \mathbb{R}^l$ defined as follows (see, e.g., [46]):

$$\begin{aligned} \pi(S) &= \Pi\left(\bigwedge_{i=1}^{l} [\xi_i = s_i]\right) = \min_{i=1,\ldots,l} \Pi(\xi_i = s_i) \\ &= \min_{i=1,\ldots,l} \mu_{\xi_i}(s_i). \end{aligned} \tag{8}$$

So the value of $\pi(S)$ is the possibility of the event that the scenario S will occur. Notice that we generalize in this way the scenario set $\Gamma \subset \mathbb{R}^l$ described in Section III. Indeed, for the interval uncertainty representation $\pi(S) = 1$ if $S \in \Gamma$ and $\pi(S) = 0$ otherwise. In the fuzzy case $\pi(S)$ may take any value in the interval $[0,1]$, so fuzzy intervals allow us to model the uncertainty in a more sophisticated manner.

In the interval case the value of deviation δ_σ falls within a closed interval. Analogously, in the fuzzy case it falls within a fuzzy interval with membership function μ_{δ_σ}. Of course, μ_{δ_σ} is a possibility distribution for the values of δ_σ and, according to possibility theory, it is defined as follows:

$$\mu_{\delta_\sigma}(x) = \Pi(\delta_\sigma = x) = \sup_{\{S:\, \delta_\sigma(S) = x\}} \pi(S). \tag{9}$$

Recall that the statement "'σ is optimal"' is equivalent to the assertion $\delta_\sigma = 0$. Consequently, we can define the degrees of possible and necessary optimality of a given sequence as follows:

$$\Pi(\sigma \text{ is optimal}) = \Pi(\delta_\sigma = 0) = \mu_{\delta_\sigma}(0), \tag{10}$$

$$\begin{aligned} \mathrm{N}(\sigma \text{ is optimal}) &= \mathrm{N}(\delta_\sigma = 0) = 1 - \Pi(\delta_\sigma > 0) \\ &= 1 - \sup_{x > 0} \mu_{\delta_\sigma}(x). \end{aligned} \tag{11}$$

As in the interval case, the question of which sequence should be chosen arises. In order to provide an answer we adopt a concept first applied to fuzzy linear programming in [32], [33]. Assume that a decision maker knows his/her preference about the sequence deviation and expresses it by using a fuzzy interval $\tilde{G} = (0, \overline{g}, 0, \beta)$. The values of the deviation in $[0, \overline{g}]$ are fully accepted, the values in $[\overline{g} + \beta, \infty)$ are not at all accepted and the degree of acceptance decreases from 1 to 0 in $[\overline{g}, \overline{g} + \beta]$. Our aim is to compute a feasible sequence $\sigma \in \mathbf{\Omega}$, for which the necessity of the event $\delta_\sigma \in \tilde{G}$ is maximal, namely we wish to solve the following optimization problem, called a *fuzzy problem*:

$$\max_{\sigma \in \mathbf{\Omega}} \mathrm{N}(\delta_\sigma \in \tilde{G}). \tag{12}$$

Observe that (12) can also be expressed as

$$\min_{\sigma \in \mathbf{\Omega}} \Pi(\delta_\sigma \in \tilde{G}^d),$$

where $\tilde{G}^d$ is the complement of $\tilde{G}$ with membership function $1 - \mu_{\tilde{G}}(x)$. If we fix $\tilde{G} = (0,0,0,0)$ in (12), then we get a special case of the fuzzy problem (12), in which we seek a feasible sequence that maximizes the degree of necessary optimality:

$$\max_{\sigma \in \mathbf{\Omega}} \mathrm{N}(\delta_\sigma = 0) = \max_{\sigma \in \mathbf{\Omega}} \mathrm{N}(\sigma \text{ is optimal}). \tag{13}$$

According to equality (7), the problem (12) is equivalent to the following optimization problem:

$$\begin{array}{ll} \min & \lambda \\ \text{s.t.} & \overline{\delta}_\sigma^{[\lambda]} \leq \overline{g}^{[1-\lambda]} \\ & \sigma \in \mathbf{\Omega} \\ & \lambda \in [0,1] \end{array} \tag{14}$$

If λ^* is the optimal objective value and σ^* is an optimal solution to problem (14), then $\mathrm{N}(\delta_{\sigma^*} \in \tilde{G}) = 1 - \lambda^*$. If problem (14) is infeasible, then $\mathrm{N}(\delta_\sigma \in \tilde{G}) = 0$ for all feasible sequences σ. Of course, if we replace expression $\overline{g}^{[1-\lambda]}$ with 0 in (14), then we get an equivalent formulation of problem (13).

Let us focus now on the quantity $\overline{\delta}_\sigma^{[\lambda]}$. The closed interval $[\underline{\delta}_\sigma^{[\lambda]}, \overline{\delta}_\sigma^{[\lambda]}]$, $\lambda \in [0,1]$, contains all the values of deviation δ_σ, whose possibility of occurrence is not less than λ. So

$$\overline{\delta}_\sigma^{[\lambda]} = \sup_{\{S\,:\pi(S) \geq \lambda\}} \{F(\sigma, S) - F^*(S)\} \tag{15}$$

is the least upper bound on δ_σ, whose possibility of occurrence is not less than λ. From the definition of $\pi(S)$ (see (8)), it is easy to see that $\{S : \pi(S) \geq \lambda\} = [\underline{\xi}_1^{[\lambda]}, \overline{\xi}_1^{[\lambda]}] \times \cdots \times [\underline{\xi}_l^{[\lambda]}, \overline{\xi}_l^{[\lambda]}] = \Gamma^{[\lambda]}$. Consequently, the quantity $\overline{\delta}_\sigma^{[\lambda]}$ is the maximal regret of σ in the minmax regret version of the underling sequencing problem with scenario set $\Gamma^{[\lambda]}$ (see (4)). In particular, the condition $\overline{\delta}_{\sigma}^{[\lambda]} = 0$ means that σ is necessarily optimal under $\Gamma^{[\lambda]}$.

The fuzzy problem (12) is a generalization of the minmax regret sequencing problem (5). It follows from the fact that the interval uncertainty representation is a special case of the fuzzy one (a closed interval is a special case of a trapezoidal fuzzy interval). If we additionally fix $\tilde{G} = (0, 0, 0, M)$ for a sufficiently large M, then (14) is equivalent to computing a minmax regret sequence. Hence the fuzzy problem is not simpler than the corresponding minmax regret one and, in particular, it is NP-hard if the underlying minmax regret problem is NP-hard. We now discuss three methods of solving the fuzzy problem (12).

1) Binary Search Method: The most general method of solving the fuzzy problem is binary search (see, e.g., [47]). It is based on the fact that $\overline{\delta}_{\sigma}^{[\lambda]}$ is nonincreasing function of $\lambda \in [0, 1]$ and $\overline{g}^{[1-\lambda]}$ is nondecreasing one. In consequence, the formulation (14) can be solved by the standard binary search if we can only decide whether there is a feasible sequence $\sigma \in \mathbf{\Omega}$ fulfilling inequality $\overline{\delta}_{\sigma}^{[\lambda]} \leq \overline{g}^{[1-\lambda]}$ for a fixed $\lambda \in [0, 1]$. This task can be done, if we have an algorithm for the corresponding minmax regret sequencing problem with interval parameters (5). After solving this problem for scenario set $\Gamma^{[\lambda]}$, we get a minmax regret sequence σ^*. Then $\overline{\delta}_{\sigma}^{[\lambda]} \leq \overline{g}^{[1-\lambda]}$ for some $\sigma \in \mathbf{\Omega}$ if and only if $\overline{\delta}_{\sigma^*}^{[\lambda]} \leq \overline{g}^{[1-\lambda]}$. The binary search algorithm is shown in Fig. 3. In lines 2 and 8 an algorithm for solving the minmax regret sequencing problem with interval parameters (5) is used. It is easy to check that if this algorithm runs in $f(n)$ time, then the fuzzy problem (12) is solvable in $O(f(n) \log \epsilon^{-1})$ time, where $\epsilon > 0$ is a given error tolerance. Therefore, if $f(n)$ is a polynomial in n then the fuzzy problem is polynomially solvable. We thus see that the problem with fuzzy parameters can be reduced to that of solving a small number of minmax regret sequencing problems. So, every exact algorithm for the minmax regret problem can easily be adapted to the more general fuzzy case. Notice that in the problem (13) it is enough to detect a necessarily optimal sequence (i.e. such that $\overline{\delta}_{\sigma}^{[\lambda]} = 0$) for scenario set $\Gamma^{[\lambda]}$. This may be computationally easier than solving the minmax regret problem.

2) Mixed Integer Programming Formulation: The binary search algorithm, shown in the previous point, is the most general method of solving (12). But it requires an exact algorithm for the minmax regret sequencing problem (5) to be executed multiple times, which may be time consuming if the underlying interval problem is NP-hard. The formulation (14) sometimes allows us to design an exact algorithm based on a mixed integer linear programming (MIP) formulation (see, e.g., [48]). The obtained MIP model can be then solved by using some standard off-the-shelf MIP solvers. We will illustrate this approach in Section V.

3) Parametric Approach: If a decision maker specifies a fuzzy goal $\tilde{G}$, then he/she gets an optimal solution according to this goal. However, it is still possible to provide a solution concept for the fuzzy problem (12) even if the fuzzy goal is

1: {Call an algorithm for the minmax regret sequencing problem (5)}
2: Find a minmax regret sequence σ under $\Gamma^{[1]}$
3: **if** $\overline{\delta}_{\sigma}^{[1]} > \overline{g}^{[0]}$ **then return** $\emptyset$
4: $\lambda_1 \leftarrow 0.5$, $k \leftarrow 1$, $\lambda_2 \leftarrow 0$
5: **while** $|\lambda_1 - \lambda_2| \geq \epsilon$ **do**
6: $\quad \lambda_2 \leftarrow \lambda_1$
7: $\quad$ {Call an algorithm for the minmax regret sequencing problem (5)}
8: $\quad$ Find a minmax regret sequence ρ under $\Gamma^{[\lambda_1]}$
9: $\quad$ **if** $\overline{\delta}_{\rho}^{[\lambda_1]} \leq \overline{g}^{[1-\lambda_1]}$ **then**
10: $\quad\quad \lambda_1 \leftarrow \lambda_1 - 1/2^{k+1}$, $\sigma \leftarrow \rho$
11: $\quad$ **else**
12: $\quad\quad \lambda_1 \leftarrow \lambda_1 + 1/2^{k+1}$
13: $\quad$ **end if**
14: $\quad k \leftarrow k + 1$
15: **end while**
16: **return** σ

Fig. 3. The binary search algorithm for solving the problem (12) with a given error tolerance $\epsilon \in (0, 1)$. The algorithm returns $\emptyset$ if $\mathrm{N}(\delta_\sigma \in \tilde{G}) = 0$ for all $\sigma \in \mathbf{\Omega}$.

not given a priori. Let us define

$$\overline{\delta}^{[\lambda]} = \min_{\sigma \in \mathbf{\Omega}} \overline{\delta}_{\sigma}^{[\lambda]}, \ \lambda \in [0, 1]. \tag{16}$$

Notice that (16) is a parametric version of the minmax regret sequencing problem (5). A solution to this problem is a partition of the unit interval $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_k = 1$ together with sequences $\sigma_1, \ldots, \sigma_k$ such that $\sigma_i = \arg\min_{\sigma \in \mathbf{\Omega}} \overline{\delta}_{\sigma}^{[\lambda]}$ for all $\lambda \in [\lambda_{i-1}, \lambda_i]$. In other words, σ_i minimizes the maximal regret under scenario set $\Gamma^{[\lambda]}$ for each $\lambda \in [\lambda_{i-1}, \lambda_i]$. In the absence of fuzzy goal, we can regard the set of sequences $\mathbf{\Omega}^* = \{\sigma_1, \ldots, \sigma_k\}$ as a solution to the fuzzy problem. We can provide the following interpretation of the solution set $\mathbf{\Omega}^*$. The first sequence σ_1 is the most conservative one. It minimizes the maximal regret over all scenarios S such that $\pi(S) > 0$. It should be chosen by very pessimistic or very risk-averse decision maker. On the other hand, the last sequence σ_k minimizes the maximal regret only over the most plausible scenarios S, i.e. such that $\pi(S) = 1$. It may be chosen by an optimistic decision maker, who considers only the most possible states of the world. So the sequences in $\mathbf{\Omega}^*$ represent the solutions of different degree of risk or conservatism. It is easy to see that if a fuzzy goal is introduced, then one of the sequences in $\mathbf{\Omega}^*$ is an optimal solution to (12). Indeed, according to (14), we choose the sequence σ_i such that $\lambda^* \in [\lambda_{i-1}, \lambda_i]$, where

$$\lambda^* = \arg\min_{\lambda \in [0,1]} \{\overline{\delta}^{[\lambda]}, \overline{g}^{[1-\lambda]}\}. \tag{17}$$

Solving the parametric problem (16) may be time consuming (see [49], [50], [51] and the references given there) and a solution algorithm should be constructed for every particular sequencing problem (see, e.g., [52], [53]). An example of an efficient method for a particular sequencing problem will be provided in Section V.

V. Applications

In this section, we provide several applications of the framework presented in Section IV. Namely, we show how to solve the fuzzy counterparts of the classical sequencing problems presented in Section II. We propose algorithms, which allow us to solve large problems arising in practice.

A. The fuzzy $1|prec|L_{\max}$ problem

Assume that for each job $i \in J$ a possibility distribution μ_{p_i} for its processing time and a possibility distribution μ_{d_i} for its due date are specified. Under each scenario, the cost of σ is the maximal lateness in σ. The fuzzy $1|prec|L_{\max}$ problem can be solved in $O(n^4 \log \epsilon^{-1})$ time by the binary search algorithm shown in Fig. 3, where $\epsilon > 0$ is a given error tolerance. In lines 2 and 8 the $O(n^4)$ algorithm proposed in [18] is called as a subroutine.

We now consider a restricted version of the $1|prec|L_{\max}$ problem, where only the due dates are fuzzy. For the simplicity of presentation, we will also assume that there are no precedence constraints between the jobs (the reasoning can be easily extended to the problem with arbitrary precedence constraints). We will show how to apply the parametric approach, described in Section IV-B3, to this problem. Let $S_j^{[\lambda]} \in \Gamma^{[\lambda]}$ be scenario under which the due date of job j is $\underline{d}_j^{[\lambda]}$ and the due dates of all the remaining jobs $i \in J \setminus \{j\}$ are $\overline{d}_i^{[\lambda]}$. Notice that the job processing times are the same under all scenarios, which follows from the assumption that all the processing times are precise. The following theorem is a direct consequence of the result proven in [18]:

Theorem 1: Given $\lambda \in [0,1]$ an optimal minmax regret sequence under scenario set $\Gamma^{[\lambda]}$ can be obtained by ordering jobs with respect to nondecreasing values of $\underline{d}_i^{[\lambda]} + F^*(S_i^{[\lambda]})$.

Consider a sample problem shown in Fig. 4. Recall that

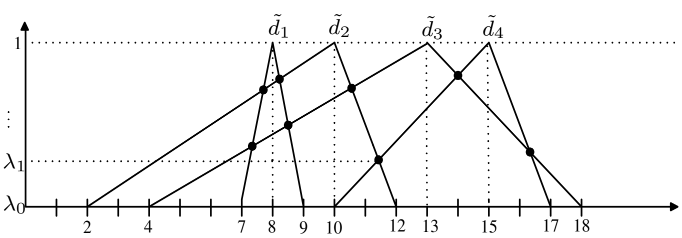

Fig. 4. A sample problem with 4 jobs, deterministic processing times $p_1 = 8$, $p_2 = 4$, $p_3 = 5$, $p_4 = 10$ and fuzzy due dates.

$F^*(S_i^{[\lambda]})$ can be obtained by sorting the jobs with respect to nondecreasing due dates under $S_i^{[\lambda]}$. Observe that the ordering of the due dates under $S_i^{[\lambda]}$ is the same between the subsequent intersection points $\lambda_0, \lambda_1, \lambda_2, \ldots$ of the fuzzy due dates (see Fig. 4). So, the sequence σ such that $F(\sigma, S_i^{[\lambda]}) = F^*(S_i^{[\lambda]})$ is the same for all $\lambda \in [\lambda_{j-1}, \lambda_j]$. In consequence, it is easy to compute the function $\underline{d}_i^{[\lambda]} + F^*(S_i^{[\lambda]})$ for $\lambda \in [0,1]$, which is a piecewise linear one with the possible breakpoints in $\lambda_0, \lambda_1, \ldots$. Furthermore, the number of such breakpoints is polynomial in the number of jobs. The functions $\underline{d}_i^{[\lambda]} + F^*(S_i^{[\lambda]})$ for all $i = 1, \ldots, 4$ are shown

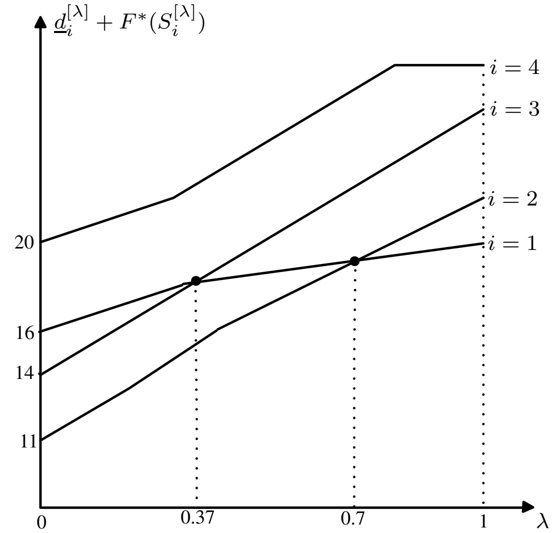

Fig. 5. The functions $\underline{d}_i^{[\lambda]} + F^*(S_i^{[\lambda]})$ for $i = 1, \ldots, 4$.

in Fig. 5. Now, according to Theorem 1, a solution to the parametric problem (16) is the partition $0 < 0.37 < 0.7 < 1$ together with the sequences $(2,3,1,4), (2,1,3,4), (1,2,3,4)$. So, the solution to the fuzzy $1|prec|L_{\max}$ problem is $\mathbf{\Omega}^* = \{(2,3,1,4), (2,1,3,4), (1,2,3,4)\}$. Introducing a fuzzy goal $\tilde{G}$, which expresses the preferences of a decision maker, allows us to choose one of the sequences of $\mathbf{\Omega}^*$.

B. The fuzzy $1|prec|\max w_i T_i$ problem

In this problem, for each job $i \in J$ the possibility distributions μ_{p_i}, μ_{d_i} and μ_{w_i} for its processing time, due date and weight are given. Under each scenario the cost of σ is the maximum weighted tardiness in σ. Under some restrictions, the fuzzy $1|prec|\max w_i T_i$ problem can be solved in polynomial time by the binary search algorithm shown in Fig. 3. If only the weights are fuzzy (the processing times and due dates are precise), then in lines 2 and 8 the $O(n^3)$ algorithm designed in [20] can be applied. In this case, the fuzzy $1|prec|\max w_i T_i$ problem is solvable in $O(n^3 \log \epsilon^{-1})$ time. If the weights and due dates are fuzzy (the processing times are precise), then in lines 2 and 8 the $O(n^4)$ algorithm proposed in [41] can be used and the fuzzy problem is solvable in $O(n^4 \log \epsilon^{-1})$ time. This algorithm can also be applied if the processing times are fuzzy. In this case we must, however, make a technical assumption that $\underline{w}_i^{[\lambda]} = 0$ for all $i \in J$ and $\lambda \in [0,1]$ (see [41]).

C. The fuzzy $1||\sum C_i$ problem

Assume that for each job $i \in J$ a possibility distribution μ_{p_i} for its processing time is specified. Under each scenario, the cost of σ is the total flow time in σ. The fuzzy $1||\sum C_i$ problem is computationally difficult, because its interval counterpart is NP-hard [22]. However, using the formulation (14), we will provide a method of solving the problem based on the mixed integer linear programming.

Let $\mathcal{A}$ be the set of all binary vectors (x_{ij}), $i = 1, \ldots, n$, $j = 1 \ldots, n$, fulfilling the so called *assignment constraints*, i.e. $\sum_{i=1}^{n} x_{ij} = 1$ for all $j = 1, \dots, n$ and $\sum_{j=1}^{n} x_{ij} = 1$ for all $i = 1, \dots, n$. A vector $(x_{ij}) \in \mathcal{A}$ represents the sequence σ in which $x_{ij} = 1$ if job $i \in J$ occupies position j in σ. Obviously, there is one to one correspondence between the sequences of the set of jobs J and the vectors in $\mathcal{A}$ (recall that there are no precedence constraints in J). If $(x_{ij}) \in \mathcal{A}$ corresponds to sequence σ, then the maximal regret of σ under scenario set $\Gamma^{[\lambda]}$ can be computed in the following way [10], [23]:

$$\overline{\delta}_{\sigma}^{[\lambda]} = \max_{(z_{ij}) \in \mathcal{A}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^{[\lambda]} z_{ij}, \tag{18}$$

where

$$c_{ij}^{[\lambda]} = \overline{p}_i^{[\lambda]} \sum_{k=1}^{j} (j-k) x_{ik} + \underline{p}_i^{[\lambda]} \sum_{k=j+1}^{n} (j-k) x_{ik}. \tag{19}$$

Observe that, for a fixed λ, (18) is the classical assignment problem with the cost coefficients of form (19). We can construct the dual of (18) and it is well known that this dual has the same optimal objective function value as (18). So, it holds:

$$\begin{aligned} \overline{\delta}_{\sigma}^{[\lambda]} = \quad & \min \quad \sum_{i=1}^{n} \theta_i + \sum_{i=1}^{n} \omega_i \\ & \text{s.t.} \quad \theta_i + \omega_j \geq c_{ij}^{[\lambda]} \text{ for } i, j = 1, \dots, n \end{aligned} \tag{20}$$

where θ_i and ω_i, $i = 1, \dots, n$, are unrestricted dual variables associated with the assignment constraints. Now, using formulation (14), we can represent the fuzzy $1||\sum C_i$ problem in the following way:

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & \sum_{i=1}^{n} \theta_i + \sum_{i=1}^{n} \omega_i \leq \overline{g}^{[1-\lambda]} \\ & \theta_i + \omega_j \geq c_{ij}^{[\lambda]} \text{ for } i, j = 1, \dots, n \\ & (x_{ij}) \in \mathcal{A} \\ & \lambda \in [0, 1] \end{aligned} \tag{21}$$

where $c_{ij}^{[\lambda]}$ has the form (19). If trapezoidal fuzzy intervals $(\underline{p}_i, \overline{p}_i, \alpha_i, \beta_i)$, $i = 1, \dots, n$, are used to model the uncertain processing times, then we can substitute $\overline{p}_i^{[\lambda]} = \overline{p}_i + (1-\lambda)\beta_i$ and $\underline{p}_i^{[\lambda]} = \underline{p}_i - (1-\lambda)\alpha_i$ in (19). The resulting model is still not linear because some expressions of the form λx_{ij} appear. However, we can linearize the model by substituting $t_{ij} = \lambda x_{ij}$ and adding additional constraints $t_{ij} - x_{ij} \leq 0$, $\lambda - t_{ij} + x_{ij} \leq 1$, $-\lambda + t_{ij} \leq 0$, $t_{ij} \geq 0$ for all $i, j = 1, \dots, n$. Hence, the resulting model is a mixed integer linear one and can be solved by using an available software.

In order to check the efficiency of the MIP formulation we performed some computational tests. We concluded that this efficiency depends on the number of jobs and the so called *degree of uncertainty*, which is the largest length of the support of the fuzzy processing times. Namely, the degree of uncertainty equal to D, means that the support of each processing time, i.e. the interval $[\underline{p}_i^{[0]}, \overline{p}_i^{[0]}]$, $i \in J$, is fully contained in the interval $[0, 100]$ and its length is at most D. For each number of jobs $n = 20, 30, \dots, 60$ and for each degree of uncertainty $D = 10, 20, \dots, 50$, we generated randomly 5 sample problems. For each instance we fixed the fuzzy goal $\tilde{G} = (0, g, 0, \beta_g)$, where $g = D$ and $\beta_g = n/3 * D$. We used CPLEX 12.1 solver and a computer equipped with Intel Core 2 CPU 1.83GHz processor and 1GB RAM to solve the generated instances. The obtained results are shown in Table I.

TABLE I
THE AVERAGE COMPUTATIONAL TIMES IN SECONDS FOR VARIOUS VALUES n AND D.

n/D	10	20	30	40	50
20	0.633	1.248	2.452	3.054	6.713
30	4.065	19.73	33.47	40.55	45.20
40	34.91	80.65	307.93	1489.05	2010.23
50	47.80	632.35	1225.17	2851.81	$>$3600
60	710.41	1808.84	$>$3600	$>$3600	$>$3600

As we can see from the obtained results, the MIP approach is efficient if the degree of uncertainty is not large ($D \leq 20$). However, if the degree of uncertainty becomes large, then one can solve efficiently the problems having up to 60 jobs.

The fuzzy $1||\sum C_i$ problem is NP-hard. However, its special case (13), in which we seek the most necessarily optimal sequence, can be solved in polynomial time. Namely, let us fix $\lambda \in [0, 1]$ and consider the interval $1||\sum C_i$ problem with scenario set $\Gamma^{[\lambda]} = [\underline{p}_i^{[\lambda]}, \overline{p}_i^{[\lambda]}] \times \dots \times [\underline{p}_n^{[\lambda]}, \overline{p}_n^{[\lambda]}]$. Consider the scenario $S \in \Gamma^{[\lambda]}$ under which the processing times $p_i(S)$, $i \in J$, are of the form $p_i(S) = \frac{1}{2}(\underline{p}_i^{[\lambda]} + \overline{p}_i^{[\lambda]})$ for all $i \in J$. Now we can compute in $O(n \log n)$ time an optimal sequence ρ under S. It turns out that $\overline{\delta}_{\rho}^{[\lambda]} \leq 2\overline{\delta}_{\sigma}^{[\lambda]}$ for all sequences σ (see [24]). In consequence, if there is a necessarily optimal sequence σ such that $\overline{\delta}_{\sigma}^{[\lambda]} = 0$, then ρ must also be necessarily optimal. We thus have an efficient method of detecting a necessarily optimal sequence and problem (13) can be solved in $O(n \log n \log \epsilon^{-1})$ time by using the binary search shown in Fig. 3.

D. The fuzzy $1|p_i = 1|\sum w_i U_i$ problem

In this problem each job $i \in J$ has a unit processing time, $p_i = 1$, and a precise due date d_i. The weights of jobs $i \in J$ are the uncertain parameters, for which possibility distributions μ_{w_i} are given. Under each scenario, the cost of σ is the weighted number of late jobs in σ. The computational complexity of the fuzzy problem is unknown. However, if all the due dates are equal, that is $d_1 = d_2 = \dots = d_n = d$, then the interval $1|p_i = 1|\sum w_i U_i$ problem can be solved in $O(n \min\{d, n-d\})$ time [42], [40] and, consequently, the fuzzy $1|p_i = 1|\sum w_i U_i$ problem is solvable in $O(n \min\{d, n-d\} \log \epsilon^{-1})$ time by the binary search shown in Fig. 3.

If the due dates are not equal, then a method based on a mixed integer programming model for solving the fuzzy $1|p_i = 1|\sum w_i U_i$ problem can be provided. Assume that the jobs are numbered so that $d_1 \leq d_2 \leq \dots \leq d_n$. Let us introduce binary variables $x_i \in \{0, 1\}$, $i \in J$, where $x_i = 1$ if the completion time of job i does not exceed d_i. It can be shown that under each scenario the number of such on-time jobs in an optimal solution is the same and equals p (see, e.g., [36]). Furthermore, the value of p can be determined in $O(n^2)$ time by a greedy algorithm. In consequence, the set of the feasible solutions can be described by the following system of constraints:

$$\begin{array}{ll} x_1 \le d_1 & \\ x_1 + x_2 \le d_2 & \\ \dots & \\ x_1 + x_2 + \cdots + x_n \le d_n & \\ x_1 + x_2 + \cdots + x_n = p & \\ x_i \in \{0,1\} & \text{for} \quad i \in J \end{array} \tag{22}$$

Having a solution to (22), we can construct a sequence by first processing all the on-time jobs (with $x_i = 1$) in order of nondecreasing due dates and then all the remaining jobs in any order. If solution (x_i) corresponds to sequence σ, then the maximal regret of σ under scenario set $\Gamma^{[\lambda]}$ can be computed in the following way [40]:

$$\begin{aligned} \overline{\delta}_\sigma^{[\lambda]} = \quad & \sum_{i=1}^{n} \overline{w}_i^{[\lambda]} x_i - \min \sum_{i=1}^{n} (\overline{w}_i^{[\lambda]} x_i + (\underline{w}_i^{[\lambda]} (1 - x_i)) y_i \\ & y_1 \le d_1 \\ & y_1 + y_2 \le d_2 \\ & \dots \\ & y_1 + y_2 + \cdots + y_n \le d_n \\ & y_1 + y_2 + \cdots + y_n = p \\ & y_i \in \{0,1\} \text{ for } i \in J \end{aligned} \tag{23}$$

Since the constraints matrix in (23) is totally unimodular, using the linear programming duality, we can transform (23) into the following equivalent formulation:

$$\begin{aligned} \overline{\delta}_\sigma^{[\lambda]} \quad & = \min \sum_{i=1}^{n} \overline{w}_i^{[\lambda]} x_i + \sum_{i=1}^{n} d_i \theta_i + \sum_{i=1}^{n} \omega_i - p\gamma \\ & - \sum_{i=j}^{n} \theta_i - \omega_j + \gamma \le \overline{w}_j^{[\lambda]} x_j + \underline{w}_j^{[\lambda]} (1 - x_j) \text{ for } j \in J \\ & \theta_i, \omega_i \ge 0 \text{ for } i \in J \end{aligned} \tag{24}$$

Now, using (14), the final model for the fuzzy $1|p_i = 1|\sum w_i U_i$ problem takes the following form:

$$\begin{array}{ll} \min & \lambda \\ \text{s.t.} & \sum_{i=1}^{n} \overline{w}_i^{[\lambda]} x_i + \sum_{i=1}^{n} d_i \theta_i + \sum_{i=1}^{n} \omega_i - p\gamma \le \overline{g}^{[1-\lambda]} \\ & - \sum_{i=j}^{n} \theta_i - \omega_j + \gamma \le \overline{w}_j^{[\lambda]} x_j + \underline{w}_j^{[\lambda]} (1 - x_j) \text{ for } j \in J \\ & x_1 \le d_1 \\ & x_1 + x_2 \le d_2 \\ & \dots \\ & x_1 + x_2 + \cdots + x_n \le d_n \\ & x_1 + x_2 + \cdots + x_n = p \\ & x_i \in \{0,1\} \text{ for } i \in J \\ & \theta_i, \omega_i \ge 0 \text{ for } i \in J \end{array} \tag{25}$$

We can use trapezoidal fuzzy intervals to model the uncertain weights and make the formulation (25) linear in the same way as in the fuzzy $1||\sum C_i$ problem (see Section V-C).

In order to check the efficiency of the MIP approach we performed some computational tests. As for the problem discussed in the previous section, this efficiency depends on the number of jobs and the degree of uncertainty, which is the largest length of the support of the fuzzy weights. Namely, the degree of uncertainty equal to D means, that the support of each fuzzy weight, i.e. the interval $[\underline{w}_i^{[0]}, \overline{w}_i^{[0]}]$, $i \in J$, is fully contained in the interval $[0, 100]$ and its length is at most D. For each number of jobs $n = 200, 300, \dots, 600$ and for each degree of uncertainty $D = 10, 20, \dots, 50$, we generated randomly 5 sample problems. For each instance we fixed the fuzzy goal $\tilde{G} = (0, g, 0, \beta_g)$, where $g = D$ and $\beta_g = n/3 * D$. We used CPLEX 12.1 solver and a computer equipped with Intel Core 2 CPU 1.83GHz processor and 1GB RAM to solve the generated instances. The obtained results are shown in Table II.

TABLE II
THE AVERAGE COMPUTATIONAL TIMES IN SECONDS FOR VARIOUS VALUES n AND D.

n/D	10	20	30	40	50
200	0.764	0.761	0.614	0.561	0.636
300	1.201	1.323	3.042	4.034	14.95
400	11.06	11.60	12.7	20.13	37.73
500	6.12	19.28	191.32	127.99	781.00
600	8.707	28.79	60.55	428.21	$>$3600

As we can see from the obtained results, the MIP formulation allows us to solve quite large problems, having up to 600 jobs. The computation times depends on the degree of uncertainty, especially for the instances with large number of jobs.

E. *The fuzzy $F2||C_{\max}$ problem*

In this problem, each job $i \in J$ must be processed on two machines, first on machine 1 and then on machine 2. So we have possibility distributions $\mu_{p_{i1}}$ and $\mu_{p_{i2}}$ for the processing times of job i on machine 1 and 2, respectively. The computational complexity of the interval $F2||C_{\max}$ problem is unknown. However, it can be solved by the branch and bound algorithm proposed in [17], which performs quite well. Thus, one can use that algorithm as a subroutine in the binary search presented in Fig. 3 (lines 2 and 8) and obtain a method of solving the fuzzy $F2||C_{\max}$ problem. Of course, the effort required to obtain an optimal solution grows fast with the problem size. Therefore, developing a better algorithm is a subject of further research.

VI. CONCLUSIONS

In this paper we have proposed a general framework for dealing with sequencing problems with uncertain parameters. The uncertainty has been modeled by possibility distributions for the values of unknown parameters. Our approach has several advantages. It collects from a decision maker both the interval uncertainty representation and likely values of the parameters. Thus, getting this information may be easier than estimating the probability distributions. Namely, a fuzzy parameter can be modeled as a trapezoidal fuzzy interval whose support contains all the possible parameter values and the core contains the most likely ones. Furthermore, the fuzzy problems are generally computationally easier than the stochastic ones. Finally, the solution concept used in this paper is consistent with the popular and widely accepted robust approach to optimization, where decision makers are interested in minimizing a solution cost in the worst case. Apart from showing the general framework, we have also described some methods to compute a solution. It is very important that determining an optimal sequence in the fuzzy problem (12) is not much harder than in the interval case (5). On the other hand, the NP-hardness of the interval problem implies that the fuzzy problem is NP-hard as well.

Our approach has also a drawback which, however, appears in most approaches that model uncertainty. Namely, the underlying minmax regret sequencing problems with interval parameters are mostly hard to solve. Furthermore, they do not possess such nice and general properties which hold true for the combinatorial problems discussed in [34]. So analyzing and solving the sequencing problems is more challenging. In this paper we have applied the fuzzy framework to several basic problems. There is, however, a large number of different problems to which this framework can also be applied. Each of these problems may have its own properties, which should be explored and this is a subject of further research.

REFERENCES

[1] P. Brucker, *Scheduling Algorithms, 5th edition*. Hedielberg: Springer Verlag, 2007.

[2] M. Rothkopf, "Scheduling with random service times," *Management Science*, vol. 12, pp. 707–713, 1966.

[3] M. Uetz, "Algorithms for Deterministic and Stochastic Scheduling," Ph.D. dissertation, Institut für Mathematik, Technische Universität Berlin, 2001.

[4] R. Weber, "Stochastic Scheduling Bibliography," http://www.statslab.cam.ac.uk/~rrw1/stoc_sched/, 1995.

[5] X. Wu and X. Zhou, "Stochastic scheduling to minimize expected maximum lateness," *European Journal of Operational Research*, vol. 190, no. 1, pp. 103–115, 2008.

[6] J. Kleinberg, Y. Rabani, and É. Tardos, "Allocating bandwidth for bursty connections," *SIAM Journal on Computing*, vol. 30, pp. 191—217, 2000.

[7] J. N. Hagstrom, "Computational complexity of pert problems," *Networks*, vol. 18, pp. 139–147, 1988.

[8] D. Dubois, H. Prade, and P. Smets, "Representing partial ignorance," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 26, pp. 361–377, 1996.

[9] S. Ferson and L. Ginzburg, "Different methods are needed to propagate ignorance and variability," *Reliability Engineering and Systems Safety*, vol. 54, pp. 133–144, 1996.

[10] P. Kouvelis and G. Yu, *Robust discrete optimization and its applications*. Boston: Kluwer Academic Publishers, 1997.

[11] R. D. Luce and H. Raiffa, *Games and decisions. Introduction and critical survey*. New York: Dover Publications, INC., 1957.

[12] H. Aissi, C. Bazgan, and D. Vanderpooten, "Min–max and min–max regret versions of combinatorial optimization problems: A survey," *European Journal of Operational Research*, vol. 197, pp. 427–438, 2009.

[13] R. Loulou and A. Kanudia, "Minimax regret strategies for greenhouse gas abatement: methodology and application," *Operation Research Letters*, vol. 25, pp. 219–230, 1999.

[14] Y. P. Li, G. H. Huang, and S. L. Nie, "A robust interval-based minimax-regret analysis approach for the identification of optimal water-resources-allocation strategies under uncertainty," *Resources, Conservation and Recycling*, vol. 54, pp. 86–96, 2009.

[15] Y. P. Li and G. H. Huang, "Minimax Regret Analysis for Municipal Solid Waste Management: An Interval-Stochastic Programming Approach," *Journal of the Air & Waste Management Association*, vol. 56, pp. 931–944, 2006.

[16] N. B. Chang and E. Davila, "Siting and Routing Assessment for Solid Waste Management Under Uncertainty Using the Grey Mini-Max Regret Criterion," *Environmental Management*, vol. 38, pp. 654–672, 2006.

[17] R. L. Daniels and P. Kouvelis, "Robust scheduling to hedge against processing time uncertainty in single stage production," *Management Science*, vol. 41, pp. 363–376, 1995.

[18] A. Kasperski, "Minimizing maximal regret in the single machine sequencing problem with maximum lateness criterion," *Operation Research Letters*, vol. 33, no. 4, pp. 431–436, 2005.

[19] P. Kouvelis, R. L. Daniels, and G. L. Vairaktarakis, "Robust scheduling of a two-machine flow shop with uncertain processing times," *IIE Transactions*, vol. 32, pp. 421–432, 2000.

[20] I. Averbakh, "Minmax regret solutions for minimax optimization problems with uncertainty," *Operation Research Letters*, vol. 27, no. 2, pp. 57–65, 2001.

[21] ——, "The minmax regret permutation flow-shop problem with two jobs," *European Journal of Operational Research*, vol. 169, pp. 761–766, 2006.

[22] V. Lebedev and I. Averbakh, "Complexity of minimizing the total flow time with interval data and minmax regret criterion," *Discrete Applied Mathematics*, vol. 154, no. 15, pp. 2167–2177, 2006.

[23] R. Montemanni, "A mixed integer programming formulation for the total flow time single machine robust scheduling problem with interval data," *Journal of Mathematical Modelling and Algorithms*, vol. 6, no. 2, pp. 287–296, 2007.

[24] A. Kasperski and P. Zieliński, "A 2-approximation algorithm for interval data minmax regret sequencing problems with the total flow time criterion," *Operation Research Letters*, vol. 36, pp. 343–344, 2008.

[25] D. Dubois and H. Prade, *Possibility theory: an approach to computerized processing of uncertainty*. New York: Plenum Press, 1988.

[26] M. Inuiguchi and J. Ramik, "Possibilistic linear programming: a brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem," *Fuzzy Sets and Systems*, vol. 111, no. 11, pp. 3–28, 2000.

[27] M. Vlach, "Single machine scheduling under fuzziness," in *Scheduling under Fuzziness*, ser. Studies in Fuzziness and Soft Computing, R. Słowiński and M. Hapke, Eds. Physica–Verlag, 2000, vol. 37, pp. 223–245.

[28] H. Ishii, M. Tada, and T. Masuda, "Scheduling problem with fuzzy due dates," *Fuzzy Sets and Systems*, vol. 46, pp. 339–347, 1992.

[29] S. Han, H. Ishii, and S. Fuji, "One machine scheduling problem with fuzzy due dates," *European Journal of Operational Research*, vol. 79, pp. 1–12, 1994.

[30] A. Kasperski, "A possibilistic approach to sequencing problems with fuzzy parameters," *Fuzzy Sets and Systems*, vol. 150, no. 1, pp. 77–86, 2005.

[31] S. Chanas and A. Kasperski, "Possible and necessary optimality of solutions in the single machine scheduling problem with fuzzy parameters," *Fuzzy Sets and Systems*, vol. 142, pp. 359–371, 2004.

[32] M. Inuiguchi and M. Sakawa, "Robust optimization under softness in a fuzzy linear programming problem," *International Journal of Approximate Reasonning*, vol. 18, pp. 21–34, 1998.

[33] M. Inuiguchi, "On possibilistic/fuzzy optimization," in *Foundations of Fuzzy Logic and Soft Computing, IFSA 2007*, ser. Lecture Notes in Artificial Intelligence 4529. Springer-Verlag, 2007, pp. 351–360.

[34] A. Kasperski and P. Zieliński, "On possibilistic combinatorial optimization problems," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, Hong Kong, 2008, pp. 324–329.

[35] E. Lawler, "Optimal sequencing of a single machine subject to precedence constraints," *Management Science*, vol. 19, pp. 544–546, 1973.

[36] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. MIT Press, 1990.

[37] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[38] T. Lai, Y. Sotskov, N. Sotskova, and F. Werner, "Optimal makespan scheduling with given bounds of processing times," *Mathematical and Computer Modelling*, vol. 26, pp. 67–86, 1997.

[39] T. Lai and Y. Sotskov, "Sequencing with uncertain numerical data for makespan minimization," *Journal of the Operational Research Society*, vol. 50, pp. 230–234, 1999.

[40] A. Kasperski, *Discrete optimization with interval data. Minmax regret and fuzzy approach*, ser. Studies in Fuzziness and Soft Computing. Berlin Heidelberg: Springer, 2008, vol. 228.

[41] A. Volgenant and C. W. Duin, "Improved polynomial algorithms for robust bottleneck problems with interval data," *Computers and Operations Research*, vol. 37, pp. 909–915, 2010.

[42] E. Conde, "An improved algorithm for selecting p items with uncertain returns according to the minmax regret criterion," *Mathematical Programming*, vol. 100, pp. 345–353, 2004.
[43] D. Dubois and H. Prade, "When upper probabilities are possibility measures," *Fuzzy Sets and Systems*, vol. 49, pp. 65–74, 1992.
[44] ——, "Possibility theory and data fusion in poorly informed environments," *Control Engineering Practice*, vol. 2, pp. 811–823, 1994.
[45] D. Dubois, H. Fargier, and P. Fortemps, "Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge." *European Journal of Operational Research*, vol. 147, pp. 231–252, 2003.
[46] D. Dubois, H. Fargier, and V. Galvagnon, "On latest starting times and floats in activity networks with ill-known durations," *European Journal of Operational Research*, vol. 147, pp. 266–280, 2003.
[47] M. Sakawa, *Fuzzy Sets and Interactive Multiobjective Optimization*. Plenum Press, New York, 1993.
[48] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
[49] E. Pistikopoulos, M. Georgiadis, and V. Dua, Eds., *Multi-parametric Programming, Process Systems Engineering*. Wiley-VCH, 2007, vol. 1.
[50] Z. Li and M. G. Ierapetritou, "A New Methodology for the General Multiparametric Mixed-Integer Linear Programming (MILP) Problems," *Industrial & Engineering Chemistry Research*, vol. 46, pp. 5141–5151, 2007.
[51] ——, "Process scheduling under uncertainty: Review and challenges," *Computers and Chemical Engineering*, vol. 32, pp. 715–727, 2008.
[52] J.-H. Ryu and E. Pistikopoulos, "A novel approach to scheduling of zero-wait batch processes under processing time variations," *Computers and Chemical Engineering*, vol. 31, pp. 101–106, 2007.
[53] S. T. McCormick, "Fast Algorithms for Parametric Scheduling Come From Extensions to Parametric Maximum Flow," *Operations Research*, vol. 47, pp. 744–756, 1999.

Adam Kasperski received the M.Sc. degree in management from the Wrocław University of Technology, Poland, in 1998, Ph. D. degree in computer science from the Wrocław University of Technology, Poland, in 2001 and Habilitation in computer science from the Systems Research Institute Polish Academy of Science. He is currently an Assistant Professor at the Institute of Industrial Engineering and Management, Wrocław University of Technology. His research interests include discrete optimization with uncertain parameters and robust optimization. He is the author of a book on minmax regret and fuzzy discrete optimization published by Springer in 2008.

Paweł Zieliński received the M.Sc. degree in computer science from the University of Wrocław, Poland, in 1993, Ph.D. degree in computer science from the Wrocław University of Technology, Poland, in 1997 and Habilitation in computer science from the Adam Mickiewicz University, Poznań, Poland, in 2009. He is currently an Assistant Professor at the Institute of Mathematics and Computer Science, Wrocław University of Technology. His research interests include discrete optimization with uncertain parameters, robust optimization, scheduling and planning in uncertain environment.