



On the complexity of single machine scheduling problems under scenario-based uncertainty

Mohamed Ali Aloulou, Federico Della Croce

► To cite this version:

Mohamed Ali Aloulou, Federico Della Croce. On the complexity of single machine scheduling problems under scenario-based uncertainty. 2006. hal-00115501

HAL Id: hal-00115501

<https://hal.science/hal-00115501>

Preprint submitted on 22 Nov 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the complexity of single machine scheduling problems under scenario-based uncertainty¹

Mohamed Ali Aloulou*, Federico Della Croce[†]

Résumé

Nous considérons des problèmes d’ordonnancement où certains paramètres des tâches sont incertains. Cette incertitude est modélisée au travers d’un ensemble fini de scénarios bien définis. Nous cherchons une solution qui soit acceptable pour l’ensemble des scénarios considérés. Plusieurs critères ont été utilisés dans la littérature pour sélectionner la “meilleure” solution. Nous utilisons ici le critère appelé *robustesse absolue*. Nous présentons des résultats algorithmiques et de complexité pour quelques problèmes classiques d’ordonnancement sur une machine.

Mots-clefs : Ordonnancement ; Données incertaines ; Scénarios ; Robustesse absolue

Abstract

We consider scheduling environments where some job characteristics are uncertain. This uncertainty is modeled through a finite set of well-defined scenarios. In such a context, we search for a solution that is acceptable for any considered scenario. For this purpose, several criteria can be applied to select among solutions. We use here the so-called *absolute robustness* criterion. We present algorithmic and computational complexity results for several single machine scheduling problems.

Key words : Scheduling; Scenario-based uncertainty; Absolute robustness.

¹Research performed while the second author was visiting LAMSADE on a research position funded by the CNRS

[†]LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. aloulou@lamsade.dauphine.fr

[‡]D.A.I., Politecnico di Torino, Italy. federico.dellacroce@polito.it

1 Introduction

This paper deals with single machine scheduling problems where some job characteristics are uncertain. This uncertainty is described through a finite set S of well-defined scenarios. We denote by p_j^s , d_j^s and w_j^s , respectively, the processing time, the release date, the due date and the weight of job j under scenario $s \in S$.

Consider a scheduling problem, denoted by $\alpha|\beta|\gamma$ according to Graham *et al.* notation [5]. Let Π be the set of feasible schedules with respect to the problem constraints. For each scenario $s \in S$, we denote by $OPT(\alpha|\beta|\gamma, s)$ the problem of finding an optimal schedule π_s^* satisfying

$$F(\pi_s^*, s) = \min_{\pi \in \Pi} F(\pi, s). \quad (1)$$

When problem parameters are uncertain, it is appropriate to search for a solution that is acceptable for any considered scenario. For this purpose, several criteria can be applied to select among solutions. In [8], Kouvelis and Yu proposed three different robustness criteria: the absolute robustness or maximal cost, the maximal regret or robust deviation and the relative robustness. In this paper, we focus on the *absolute robustness* criterion.

To the best of our knowledge, the absolute robustness in single machine scheduling problems has only been considered in [3] and [12] where, for the $1||\sum C_j$ problem with uncertain processing times, two distinct proofs of the *NP*-hardness even for $|S| = 2$ were provided (notice that the corresponding deterministic version is well known to be polynomially solvable [11]). The maximal regret criterion was instead much more studied (see for instance, again [12] but also [1, 6, 7]).

The absolute robustness of schedule π over all scenarios $s \in S$ is denoted by $\bar{F}(\pi)$. We have

$$\bar{F}(\pi) = \max_{s \in S} F(\pi, s). \quad (2)$$

We denote by $MinMax(\alpha|\beta|\gamma, \theta)$ the problem of finding a schedule π^A minimizing the absolute robustness $\bar{F}(\pi)$ among all schedules $\pi \in \Pi$. Field θ indicates the set of uncertain problem parameters. For the problems considered here, $\theta \subseteq \{p_j, d_j, w_j\}$. Sequence π^A is called in [8] *absolute robust sequence*. Its cost $\bar{F}(\pi^A)$ satisfies

$$\bar{F}(\pi^A) = \min_{\pi \in \Pi} \bar{F}(\pi) = \min_{\pi \in \Pi} \max_{s \in S} F(\pi, s). \quad (3)$$

Notice that if problem $\alpha|\beta|\gamma$ is NP-hard, then, in the presence of uncertainty, the corresponding problems $MinMax(\alpha|\beta|\gamma, \theta)$ are also NP-hard. However, if problem $\alpha|\beta|\gamma$ is polynomially solvable, then, the corresponding problems $MinMax(\alpha|\beta|\gamma, \theta)$ are not necessarily polynomially solvable. In this work we establish the complexity status for

the absolute robustness versions of the most well known non-preemptive polynomial-time single machine scheduling problems, namely problems $1|prec|f_{\max}$ (with $f_{\max} \in \{C_{\max}, L_{\max}, T_{\max}\}$), $1||\sum w_j C_j$ and $1||\sum U_j$. Notice that all these problems present regular cost functions non-decreasing in the job completion times. In this context any schedule $\pi \in \Pi$ is completely characterized by the corresponding job sequence. Given a schedule $\pi \in \Pi$, the completion time of job j under scenario s , denoted by $C_j(\pi, s)$, $j = 1, \dots, n$, $s \in S$, can easily be determined and the quality of the schedule $\pi \in \Pi$ under scenario s is then evaluated using the regular cost function $F(\pi, s)$. We consider the following cost functions :

- the general maximum cost function $f_{\max} = \max_j \{f_j(C_j)\}$ with $f_{\max} \in \{C_{\max}, L_{\max}, T_{\max}\}$: hence we deal with the maximum completion time (makespan) $C_{\max} = \max_j \{C_j\}$, the maximum lateness $L_{\max} = \max_j \{L_j\}$ with $L_j = C_j - d_j$ and the maximum tardiness $T_{\max} = \max_j \{T_j\}$ with $T_j = \max\{0, L_j\}$;
- the total weighted completion time $\sum_j w_j C_j$;
- the number of late jobs $\sum_j U_j$ with $U_j = 0$, if job j is *on-time* ($C_j \leq d_j$) and $U_j = 1$ if j is *late* ($C_j > d_j$).

Using the set S of scenarios, we construct a scenario s^w in which parameters k_j take their worst case value, denoted by k_j^w . In our case, we have $p_j^w = \max_{s \in S} p_j^s$, $d_j^w = \min_{s \in S} d_j^s$ and $w_j^w = \max_{s \in S} w_j^s$. Notice that in the context of a discrete set of scenarios, the constructed scenario is not necessarily feasible, i.e. we can have $s^w \notin S$: s^w is called *worst-case artificial scenario*.

Remark 1 When parameters are interval-uncertain, s^w is a feasible scenario. In this case, an absolute robust solution π^A of problem $MinMax(1|\beta|\gamma, \theta)$ is such that

$$\bar{F}(\pi^A) = \min_{\pi \in \Pi} \bar{F}(\pi) = \min_{\pi \in \Pi} \max_{s \in S} F(\pi, s) \quad (4)$$

$$= \min_{\pi \in \Pi} F(\pi, s^w). \quad (5)$$

Hence π^A is also optimal for problem $OPT(1|\beta|\gamma, s^w)$. This means that the problem of finding an absolute robust sequence can be solved straightforwardly by the algorithm solving the problem without uncertainty applied to the worst-case artificial scenario. ■

When uncertainty is scenario-based, we can not apply the same reasoning because scenario s^w is not necessarily feasible. Nevertheless, we show in this paper that problems $MinMax(1|prec|L_{\max}, d_j)$ and $MinMax(1||\sum U_j, d_j)$ can be solved by Lawler's algorithm [9] and Moore's algorithm [10] respectively applied to the worst case artificial

scenario. We also prove that an extension of Lawler's algorithm, called here *MinMax-Lawler*, solves problem $MinMax(1|prec|f_{\max}, p_j, d_j)$ in polynomial time. On the other hand, problems $MinMax(1||\sum w_j C_j, w_j)$ and $MinMax(1||\sum U_j, p_j)$ are proved to be NP-hard even when $|S| = 2$.

Table 1 summarizes the above results presenting the complexity status for the absolute robustness versions of the most well known non-preemptive polynomial-time single machine scheduling problems, where an entry "-" indicates that the considered case is not applicable (for instance problem $1||\sum w_j C_j$ cannot have uncertainty on due dates as due dates are not present in the problem).

Table 1: Summary of the obtained results

Uncertain parameter	cost function		
	$1 \sum w_j C_j$	$1 prec f_{\max}$	$1 \sum U_j$
d_j	–	$O(n^2 + n S)$ (Th. 1)	$O(n \log n + n S)$ (Th. 4)
p_j	NP-hard [3, 12]	$O(n^2 S)$ (Th. 2)	NP-hard (Th. 5)
w_j	NP-hard (Th. 3)	–	–
p_j and d_j	–	$O(n^2 S)$ (Coro. 2)	NP-hard (Coro. 3)
p_j and w_j	NP-hard [3, 12]	–	–

2 Problem $MinMax(1|prec|f_{\max}, \theta)$

2.1 Uncertainty on due dates

We consider problem $MinMax(1|prec|f_{\max}, d_j)$ where processing times are deterministic and due dates are uncertain (here $f_{\max} \in \{L_{\max}, T_{\max}\}$ as for C_{\max} no uncertainty holds). In this case the worst-case artificial scenario s^w is such that, for all $j \in N$, $d_j^w = \min_{s \in S} d_j^s$.

We recall that problem $1|prec|f_{\max}$ can be solved in $O(n^2)$ time by Lawler's algorithm [9]. This algorithm constructs an optimal schedule backwards. At the points in time where the unscheduled jobs should complete, starting with point $t = P = \sum_{j \in N} p_j$, Lawler's algorithm chooses among the unscheduled jobs having no successors a job with minimum cost to be completed at t . Notice that as processing times are deterministic, we have

$$\forall \pi \in \Pi, \forall s \in S, \forall j \in N, C_j(\pi, s) = C_j(\pi). \quad (6)$$

The following theorem holds.

Theorem 1. *Problem $MinMax(1|prec|f_{\max}, d_j)$ can be optimally solved in $O(n^2 + n|S|)$ time by means of Lawler's algorithm applied to the worst-case artificial scenario s^w .*

Proof. In the following, for the sake of clarity we consider that $f_{\max} = L_{\max}$ but the same analysis holds for $f_{\max} = T_{\max}$. An absolute robust solution π^A of problem $MinMax(1|prec|L_{\max}, d_j)$ is such that

$$\bar{L}(\pi^A) = \min_{\pi \in \Pi} \max_{s \in S} L_{\max}(\pi, s) \quad (7)$$

$$= \min_{\pi \in \Pi} \max_{s \in S} \max_{j \in N} (C_j(\pi, s) - d_j^s) \quad (8)$$

$$= \min_{\pi \in \Pi} \max_{s \in S} \max_{j \in N} (C_j(\pi) - d_j^s) \quad (9)$$

$$= \min_{\pi \in \Pi} \max_{j \in N} \max_{s \in S} (C_j(\pi) - d_j^s) \quad (10)$$

$$= \min_{\pi \in \Pi} \max_{j \in N} (C_j(\pi) - d_j^w) \quad (11)$$

$$\bar{L}(\pi^A) = \min_{\pi \in \Pi} L_{\max}(\pi, s^w) \quad (12)$$

Hence, π^A is also an optimal solution for problem $OPT(1|prec|L_{\max}, s^w)$. For the complexity, the construction of the worst-case scenario requires $O(n|S|)$ time and the application of Lawler's algorithm requires $O(n^2)$ time, hence the overall complexity is $O(n^2 + n|S|)$. ■

We observe that the proof of Theorem 1 can be applied, as it is, to any scheduling problem $\alpha|\beta|f_{\max}$. Hence, we have the following result.

Corollary 1 *Any algorithm optimally solving problem $\alpha|\beta|f_{\max}$ provides an absolute robust solution for problem $MinMax(\alpha|\beta|f_{\max}, d_j)$, when applied to the worst-case artificial scenario s^w .*

2.2 Uncertainty on processing times and due dates

We consider problem $MinMax(1|prec|f_{\max}, p_j, d_j)$ where we suppose now that both processing times and due dates are uncertain. A robust solution π^A is such that

$$\bar{F}(\pi^A) = \min_{\pi \in \Pi} \max_{s \in S} F_{\max}(\pi, s) \quad (13)$$

$$= \min_{\pi \in \Pi} \max_{s \in S} \max_{j \in N} f_j(C_j(\pi, s)) \quad (14)$$

We propose an algorithm, called *MinMax-Lawler*, which is an extension of Lawler's algorithm. This algorithm constructs a sequence π in reverse order. Let U be the set of unscheduled jobs. Define $p^s(U) = \sum_{j \in U} p_j^s$ for all $s \in S$. The rule is the following : Schedule last the job $j \in U$, which has no successor in U and such that $\max_{s \in S} f_j^s(p^s(U))$ is minimal. It is immediate to see that the complexity of *MinMax-Lawler* is $O(n^2|S|)$.

We have the following result.

Theorem 2. *Problem $MinMax(1|prec|f_{\max}, p_j, d_j)$ is optimally solved by algorithm MinMax-Lawler.*

Proof. The proof is very similar to the proof of Lawler's algorithm optimality for problem $1|prec|f_{\max}$.

Enumerate the jobs in such a way that $(1, 2, \dots, n)$ is the sequence constructed by the proposed algorithm. Let π^A be an absolute robust sequence for problem $MinMax(1|prec|f_{\max}, p_j, d_j)$ with $\pi^A(i) = i$ for $i = n, n-1, \dots, r$ and $\pi^A(r) = j < r$.

Notice that it is possible to schedule $r-1$ immediately before r . Hence, we can construct a sequence π' in which we shift to the left the block between jobs $r-1$ and r and process $r-1$ immediately before r . Clearly,

$$\forall i \in N - \{r-1\}, \forall s \in S, C_i(\pi', s) \leq C_i(\pi^A, s) \quad (15)$$

Hence,

$$\bar{F}(\pi') = \max_{s \in S} F(\pi', s) = \max\{\bar{F}(\pi^A), \max_{s \in S} f_{r-1}^s(C_{r-1}(\pi', s))\} \quad (16)$$

$$= \max\{\bar{F}(\pi^A), \max_{s \in S} f_{r-1}^s(C_j(\pi^A, s))\} \quad (17)$$

$$\leq \max\{\bar{F}(\pi^A), \max_{s \in S} f_j^s(C_j(\pi^A, s))\} \quad (18)$$

$$\leq \bar{F}(\pi^A) \quad (19)$$

Consequently π' is also an absolute robust sequence.

We can reiterate the same reasoning and transform sequence π^A into sequence $(1, 2, \dots, n)$ without increasing the objective function value. ■

Correspondingly, the following corollary also holds.

Corollary 2 *Problem $MinMax(1|prec|f_{\max}, p_j)$ is optimally solved by algorithm MinMax-Lawler.*

3 Problem $MinMax(1||\sum w_j C_j, w_j)$

We consider problem $MinMax(1||\sum w_j C_j, w_j)$ where processing times are deterministic and weights are uncertain. We prove that this problem is *NP*-hard even when $|S| = 2$ and $p_j = 1 \ \forall j$. To this extent we need to prove the following instrumental lemma.

Lemma 1. *The $1||\sum C_j$ problem and the $1|p_j = 1|\sum w_j C_j$ problem are equivalent.*

Proof. Given any instance of the $1||\sum C_j$ problem where each job j has processing time p'_j , generate an instance of the $1|p_j = 1|\sum w_j C_j$ problem where each job j has weight $w''_j = p'_{n-j+1}$. Consider a generic sequence $(1, 2, \dots, n-1, n)$. For the $1||\sum C_j$ problem the corresponding cost function value is $Z_1 = \sum_{j=1}^n (n-j+1)p'_j$. For the $1|p_j = 1|\sum w_j C_j$ problem the corresponding cost function value is $Z_2 = \sum_{j=1}^n jw''_j$. We show that $Z_2 = Z_1$. Indeed, $Z_2 = \sum_{j=1}^n jw''_j = \sum_{j=1}^n jp'_{n-j+1} = \sum_{j=1}^n (n-j+1)p'_j = Z_1$. ■

Theorem 3. *Problem $MinMax(1||\sum w_j C_j, w_j)$ is NP-hard even when $|S| = 2$ $p_j = 1 \forall j$.*

Proof. Due to Lemma 1 and the NP-hardness of problem $MinMax(1||\sum C_j, p_j)$ from [3, 12], the proof immediately holds. ■

4 Problem $MinMax(1||\sum U_j, \theta)$

We consider problem $MinMax(1||\sum U_j, \theta)$ with $\theta \subseteq \{p_j, d_j\}$. We show that problem $MinMax(1||\sum U_j, d_j)$ can be solved by Moore's algorithm [10] applied to the worst-case artificial scenario, whereas problem $MinMax(1||\sum U_j, p_j)$ is NP-hard.

4.1 Uncertainty on due dates

Lemma 2. *There exist an optimal solution of problem $MinMax(1||\sum U_j, d_j)$ in which on-time jobs are scheduled in a non-decreasing order of their worst-case artificial scenario due dates d_j^w .*

Proof. Let π^A be an optimal solution for problem $MinMax(1||\sum U_j, d_j)$. Suppose that in this schedule there exist two consecutive on-time jobs i and j such that $i \prec j$ and $d_i^w \geq d_j^w$. We have

$$\forall s \in S, C_i(\pi^A, s) = C'_i(\pi^A) \leq d_i^s \quad (20)$$

and

$$\forall s \in S, C_j(\pi^A, s) = C'_j(\pi^A) \leq d_j^s \quad (21)$$

By interchanging jobs i and j , we obtain a new feasible schedule denoted by π . In this schedule, only completion time of job i increases and we have

$$C_i(\pi) = C_j(\pi^A) \quad (22)$$

According to equation (21), we obtain

$$C_i(\pi) \leq \max_{s \in S} d_j^s \leq d_j^w \leq d_i^w \leq d_i^s, \forall s \in S \quad (23)$$

This means that job i is on-time in schedule π and consequently π is also optimal for problem $MinMax(1 || \sum U_j, d_j)$.

We can reiterate the same reasoning and obtain finally a solution in which on-time jobs are sequenced in a non-decreasing order of their worst-case artificial scenario due dates d_j^w . ■

Theorem 4. *Problem $MinMax(1 || \sum U_j, d_j)$ can be optimally solved in $O(n \log n + n|S|)$ time by means of Moore's algorithm applied to the worst-case artificial scenario s^w .*

Proof. The proof is very similar to the proof of Moore's algorithm optimality for problem $1 || \sum U_j$.

According to Lemma 2, there exists an optimal schedule $\pi = (\pi_I, \pi_L)$ for problem $MinMax(1 || \sum U_j, d_j)$ in which

- on-time jobs are scheduled before late jobs, and,
- on-time jobs are scheduled according to the non-decreasing order of their worst-case artificial scenario due dates d_j^w .

Without loss of generality, enumerate the jobs such that $d_1^w \leq d_2^w \leq \dots \leq d_n^w$. Let j be the first job deleted, in Moore's algorithm, from the set I of on-time jobs. Suppose that job j is on-time in π . We transform π in another optimal schedule in which j is late. Remark that there exists at least one job k , with $k < j$, which is late in π . Otherwise, Moore's algorithm would not have rejected j . It is immediate to prove that by interchanging jobs j and k we get a new optimal schedule in which job j is late. We can reiterate the same reasoning and transform then sequence π^A into the sequence obtained by Moore's algorithm applied to the worst-case artificial scenario s^w without increasing the number of late jobs.

For the complexity, the construction of the worst-case scenario requires $O(n|S|)$ time and Moore's algorithm requires $O(n \log n)$ time, hence the overall complexity is $O(n \log n, n|S|)$. ■

4.2 Uncertainty on processing times

We prove that problem $MinMax(1||\sum U_j, p_j)$ is NP-hard even when $|S| = 2$ by constructing a reduction from the NP-complete even-odd partition problem [4].

Even-odd partition problem: Given $2n$ positive integers $a_1 < a_2 < \dots < a_{2n}$ where $\sum_{j=1}^{2n} a_j = 2B$, is there a partition of the integers into two subsets A_1 and A_2 such that

$$\sum_{j \in A_1} a_j = \sum_{j \in A_2} a_j = B \quad (24)$$

and A_1 and A_2 each contains exactly one element of each pair (a_{2i-1}, a_{2i}) , $i = 1, \dots, n$?

Given an instance of the even-odd partition problem, we construct the following instance of problem $MinMax(1||\sum U_j, p_j)$.

Instance of problem $MinMax(1||\sum U_j, p_j)$: We have $2n$ jobs and two scenarios 1 and 2. Job processing times and due dates are such that

$$\begin{cases} p_1^1 = p_2^2 = a_1 \\ p_1^2 = p_2^1 = a_2 \\ d_1 = d_2 = a_2 \end{cases} \quad (25)$$

$$\forall i = 2, \dots, n \begin{cases} p_{2i-1}^1 = p_{2i}^2 = a_{2i-1} + a_{2i-2} + \dots + a_2 \\ p_{2i-1}^2 = p_{2i}^1 = a_{2i} + a_{2i-2} + \dots + a_2 \\ d_{2i-1} = d_{2i} = a_{2i} + 2a_{2i-2} + \dots + ia_2, \text{ for } i < n, \text{ and,} \\ d_{2n} = d_{2n-1} = B + d_{2n-2} \end{cases} \quad (26)$$

Jobs $2i-1$ and $2i$ are called $a_{[i]}$ -jobs. Notice that with this construction, the following lemmata hold.

Lemma 3. Any sequence in which each position $i, i = 1, \dots, n$, is occupied by an $a_{[i]}$ -job admits at least $n - 1$ on-time jobs.

Proof. We can prove straightforwardly that in such a sequence the first $n - 1$ jobs are on-time. ■

Lemma 4 If two jobs $2i$ and $2i - 1$ are both on-time, then there exists j , $1 \leq j < i$, such that jobs $2j$ and $2j - 1$ are both late.

Proof. Suppose by contradiction that there exists a sequence in which for all $j, 0 \leq j < i$, at least one job $2j$ or $2j - 1$, denoted $a_{[j]}$ -job, is on time and that jobs $2i$ and $2i - 1$ are both on-time.

It is clear that resequencing these on-time jobs in a non-decreasing order of their due dates does not increase the number of late jobs.

In this new sequence, jobs $2i$ and $2i - 1$ are sequenced last with respect to the considered on-time jobs. The completion time of the last job ($2i$ or $2i - 1$) is greater than or equal to the sum of

1. the sum of the minimal processing times of $a_{[j]}$ -jobs scheduled before jobs $2i$ and $2i - 1, 0 \leq j < i$, denoted by A , and,
2. the processing times of jobs $2i$ and $2i - 1$, denoted by B .

We have

$$A = \begin{cases} a_1 + \\ a_3 + a_2 + \\ a_5 + a_4 + a_2 + \\ \dots \\ a_{2i-3} + a_{2i-4} + \dots + a_4 + a_2 \end{cases} \quad (27)$$

$$= a_1 + a_3 + \dots + a_{2i-3} + a_{2i-4} + \dots + (i - 2) * a_2 \quad (28)$$

and

$$B = p_{2i}^1 + p_{2i-1}^1 = p_{2i}^2 + p_{2i}^2 \quad (29)$$

$$= a_{2i} + a_{2i-1} + 2 * a_{2i-2} + \dots + 2 * a_4 + 2 * a_2. \quad (30)$$

The completion time of the last job is greater than or equal to

$$A + B = a_1 + a_3 + \dots + a_{2i-3} + a_{2i} + 2a_{2i-2} + \dots + ia_2 \quad (31)$$

$$= a_1 + a_3 + \dots + a_{2i-3} + d_{2i} \quad (32)$$

$$> d_{2i}. \quad (33)$$

Hence at least one job among jobs $2i$ and $2i - 1$ is late. This contradicts the considered hypothesis. ■

Lemma 5. *Every schedule admits at most n on-time jobs.*

Proof. Suppose by contradiction that there exists a sequence such that the number of on-time jobs is greater than or equal to $n + 1$. Resequencing the on-time jobs in a non-decreasing order of their due dates and sequencing late jobs last does not increase the number of late jobs. Let π be such a sequence.

Since we have $n + 1$ on-time jobs, then there exist at least an integer i such that jobs $2i$ and $2i - 1$ are both on-time. Let i^* be the smallest i such that $2i$ and $2i - 1$ are both on-time and π_{i^*} be the subsequence of on-time jobs among jobs k , $1 \leq k \leq 2i^*$. We have $\pi = (\pi_{i^*}, \pi_R)$. Notice that in subsequence π_{i^*} there are at most i^* jobs, which are all on-time.

Due to Lemma 4, there exists an integer j , $1 \leq j < i^*$, such that neither job $2j$ nor job $2j - 1$ is on-time. We construct a new subsequence sequence π'_{i^*} in which we exchange one job among $2i^*$ and $2i^* - 1$ (which are on-time in π) with job $2j$ or $2j - 1$ (which are late in π). For example, we exchange jobs $2i^*$ and $2j - 1$. Then, we reorder the jobs of π'_{i^*} in a non decreasing order of their due dates. Due to Lemma 3, all jobs of π'_{i^*} are on-time. Since $\forall s \in S, p_{2j-1}^s < p_{2i^*}^s$, then sequence (π'_{i^*}, π_R) has at least the same number jobs on-time as π .

We can reiterate the same reasoning until we get a sequence in which we have at most $n - 1$ jobs on-time among jobs $1, \dots, 2n - 2$. Since we have at least $n + 1$ jobs on-time, then jobs $2n$ and $2n - 1$ are both on-time but this cannot occur due to Lemma 4. ■

The following theorem hold.

Theorem 5. *Problem $\text{MinMax}(1 || \sum U_j, p_j)$ is NP-hard even when $|S| = 2$.*

Proof. We first prove that, if the even-odd partition problem has a solution, then the constructed instance admits a solution with n on-time jobs.

Let A_1 be a solution of the even-odd partition problem. Consider a sequence $\pi = (\pi_1, \pi_2)$ such that

- for all $j \leq 2n$, if $a_j \in A_1$ then job $j \in \pi_1$;
- the jobs of π_1 are ordered in non-decreasing order of their due dates.

Suppose that $\pi = ([1], [2], \dots, [n], [n+1], [n+2], \dots, [2n])$ and $\pi_1 = ([1], [2], \dots, [n])$

Due to Lemma 3, the first $n - 1$ jobs in π_1 are on-time. We prove that the last job in π_1 is also on-time.

The completion time of the last job $[n]$ in scenario s_1 is

$$C_{[n]}(\pi, s_1) = p_{[1]}^1 + p_{[2]}^1 + \cdots p_{[n]}^1 \quad (34)$$

$$= \begin{cases} a_{[1]} + \\ a_{[2]} + a_2 + \\ \dots \\ a_{[n]} + a_{2n-2} + \cdots + a_4 + a_2 \end{cases} \quad (35)$$

$$= a_{[1]} + a_{[2]} + \cdots + a_{[n]} + a_{2n-2} + \cdots + (n-1) * a_2 \quad (36)$$

$$= B + d_{2n-2} \quad (37)$$

$$= d_{2n} = d_{2n-1} = d_{[n]}. \quad (38)$$

The completion time of the last job $[n]$ in scenario s_2 is

$$C_{[n]}(\pi, s_2) = p_{[1]}^2 + p_{[2]}^2 + \cdots p_{[n]}^2 \quad (39)$$

$$= \begin{cases} a_{[n+1]} + \\ a_{[n+2]} + a_2 + \\ \dots \\ a_{[2n]} + a_{2n-2} + \cdots + a_4 + a_2 \end{cases} \quad (40)$$

$$= a_{[n+1]} + a_{[n+2]} + \cdots + a_{[2n]} + a_{2n-2} + \cdots + (n-1) * a_2 \quad (41)$$

$$= B + d_{2n-2} \quad (42)$$

$$= d_{2n} = d_{2n-1} = d_{[n]}. \quad (43)$$

Consequently, job $[n]$ is on-time for the two scenarios and the number of on-time jobs in π is at least n and cannot be greater than n due to Lemma 5.

We prove now that if the constructed instance admits a solution with n on-time jobs, then the even-odd partition problem admits a solution.

Let π be a sequence in which the number of on-time jobs is equal to n . Suppose that in π , there exist two on-time jobs $2i$ and $2i-1$. Then, due to Lemma 4, there exists an index j , $1 \leq j < i$, such that jobs $2j$ and $2j-1$ are both late. Using the same technique as in Lemma 5, we transform sequence π in a new sequence in which job $2j$ or $2j-1$ is on-time and job $2i$ or $2i-1$ is late. We reiterate the same reasoning until we get a sequence π' , in which for all k , $1 \leq k \leq n$, only one job among jobs $2k$ and $2k-1$ is on-time.

Resequencing the on-time jobs in a non-decreasing order of their due dates and sequencing the late jobs last we do not increase the number of late jobs. We have $\pi' = ([1], [2], \dots, [n], [n+1], [n+2], \dots, [2n])$ and jobs $[i]$, $1 \leq i \leq n$ are on-time.

The completion time of job $[n]$ in scenario s_1 is

$$C_{[n]}(\pi', s_1) = p_{[1]}^1 + p_{[2]}^1 + \cdots p_{[n]}^1 \quad (44)$$

$$= a_{[1]} + a_{[2]} + \cdots + a_{[n]} + a_{2n-2} + \cdots + (n-1) * a_2 \quad (45)$$

$$= a_{[1]} + a_{[2]} + \cdots + a_{[n]} + d_{2n-2} \quad (46)$$

The completion time of job $[n]$ in scenario s_2 is

$$C_{[n]}(\pi', s_2) = p_{[1]}^2 + p_{[2]}^2 + \cdots p_{[n]}^2 \quad (47)$$

$$= a_{[n+1]} + a_{[n+2]} + \cdots + a_{[2n]} + a_{2n-2} + \cdots + (n-1) * a_2 \quad (48)$$

$$= a_{[n+1]} + a_{[n+2]} + \cdots + a_{[2n]} + d_{2n-2}. \quad (49)$$

Since job $[n]$ is on-time then

$$\forall s \in \{s_1, s_2\}, C_{[n]}(\pi', s) \leq d_{[n]} = d_{2n} = d_{2n-1} = B + d_{2n-2}. \quad (50)$$

Consequently, we have

$$\begin{cases} a_{[1]} + a_{[2]} + \cdots + a_{[n]} \leq B, \\ a_{[n+1]} + a_{[n+2]} + \cdots + a_{[2n]} \leq B, \text{ and,} \\ a_{[1]} + a_{[2]} + \cdots + a_{[2n]} = B. \end{cases} \quad (51)$$

This proves that

$$a_{[1]} + a_{[2]} + \cdots + a_{[n]} = a_{[n+1]} + a_{[n+2]} + \cdots + a_{[2n]} = B, \quad (52)$$

which means that the even-odd partition problem has a solution. ■

Correspondingly, the following corollary also holds.

Corollary 3 *Problem MinMax(1|| $\sum U_j, p_j, d_j$) is NP-hard even when $|S| = 2$.*

References

- [1] I. Averbakh, Minmax regret solutions for minimax optimization problems with uncertainty, Operations Research Letters, 27, 57-65, 2000.

- [2] K.R. Baker, Introduction to Sequencing and Scheduling, John Wiley and Sons, New York, NY, 1974.
- [3] R.L. Daniels, P. Kouvelis, Robust scheduling to hedge against processing time uncertainty in single stage production, *Management Science* 41: 363–376, 1995.
- [4] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.
- [5] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, Optimization and approximation in deterministic machine scheduling: a survey, *Annals of Discrete Mathematics* 5: 287–326, 1979.
- [6] A. Kasperski, Minimizing maximal regret in the single machine sequencing problem with maximum lateness criterion, *Operations Research Letters*, 33, 431–436, 2005.
- [7] P. Kouvelis and R.L. Daniels and G. Vairaktarakis, Robust scheduling of a two-machine flow shop with uncertain processing times, *IIE Transactions*, 32: 421–432, 2000.
- [8] P. Kouvelis and G. Yu, Robust Discrete Optimisation and its Applications, Kluwer Academic Publisher, 1997.
- [9] E.L. Lawler, Optimal sequencing of a single machine subject to precedence constraints, *Management Science* 19: 544–546, 1973
- [10] J.M. Moore, A n job, one machine sequencing algorithm for minimizing the number of late jobs, *Management Science* 15: 102–109, 1968.
- [11] W.E. Smith, Various optimizers for single-stage production, *Naval Research Logistics Quarterly* 3 (1956) 59–66.
- [12] J. Yang and G. Yu, On the Robust Single Machine Scheduling Problem, *Journal of Combinatorial Optimization*, 6, 17–33, 2002.