Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Discrete Applied Mathematics 154 (2006) 2167–2177
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Complexity of minimizing the total flow time with interval data and minmax regret criterion

Vasilij Lebedev[a], Igor Averbakh[b]

[a] *Department of Mathematics, Volgograd State University, Volgograd 400062, Russia*
[b] *Division of Management, University of Toronto at Scarborough, 1265 Military Trail, Scarborough, Ont. M1C Canada 1A4*

Received 17 September 2002; received in revised form 19 November 2003; accepted 21 April 2005
Available online 23 March 2006

Abstract

We consider the minmax regret (robust) version of the problem of scheduling n jobs on a machine to minimize the total flow time, where the processing times of the jobs are uncertain and can take on any values from the corresponding intervals of uncertainty. We prove that the problem in NP-hard. For the case where all intervals of uncertainty have the same center, we show that the problem can be solved in $O(n \log n)$ time if the number of jobs is even, and is NP-hard if the number of jobs is odd. We study structural properties of the problem and discuss some polynomially solvable cases.

Keywords: Minmax regret; Robust scheduling; Uncertainty; Complexity

1. Introduction

Minmax regret optimization deals with optimization problems where the objective function is uncertain at the time of solving the problem. Uncertainty is described by a given set of possible realizations of the objective function (scenarios); it is required to find a feasible solution that would minimize the worst-case loss in the objective function value that may occur because the solution is chosen before the actual realized scenario becomes known. The book [6] gives the state-of-art in minmax regret combinatorial optimization (MRCO) up to 1997 and provides a comprehensive discussion of the motivation for the minmax regret approach and various aspects of applying it in practice. Minmax regret solutions are sometimes called *robust* solutions [6], although there are different concepts of robustness in the literature (e.g. [7]). Minmax regret solutions can also be interpreted as *uniformly suboptimal* solutions, that is, solutions that are ε-optimal for all realizations of data, with ε as small as possible.

We consider the minmax regret version of the problem of scheduling n jobs on a machine to minimize the total flow time where the processing times of the jobs are uncertain. We consider the interval-data representation of uncertainty, that is, we assume that the processing time of each job can take on any value from the corresponding interval of uncertainty, regardless of the values taken by the processing times of other jobs. Thus, the set of possible scenarios (possible vectors of processing times of the jobs) is a rectangular box in R^n. The problem was introduced by Daniels and Kouvelis [3]; they studied structural properties of the problem, developed some heuristics and a branch-and-bound exact

E-mail address: averbakh@utsc.utoronto.ca (I. Averbakh).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2005.04.015

algorithm, and proved NP-hardness of a related *discrete-scenario* problem (to be discussed below). The complexity status of the interval-data problem has been unknown so far.

In this paper, we prove that the problem is NP-hard. For the case where all intervals of uncertainty have the same center, it turns out that, surprisingly enough, the complexity status depends on the parity of the number of jobs: we show that the problem can be solved in $O(n \log n)$ time it n is even, and it is NP-hard if n is odd. We also obtain some structural properties of the problem, and discuss some polynomially solvable special cases.

We note that in the case of discrete-scenario representation of uncertainty, where the set of possible scenarios is finite and is represented by listing explicitly all possible scenarios (vectors of processing times) as a part of the input, the problem is known to be NP-hard [3,6] even if there are only two scenarios. It is a general observation that most MRCO problems are NP-hard in the case of discrete-scenario representation of uncertainty [6]. However, Averbakh [1] showed that there is no direct relationship between the complexity of the discrete-scenario and interval data MRCO problems, and there exist MRCO problems that are NP-hard in the discrete-scenario version but are polynomially solvable in the interval data version [1].

A different approach to studying scheduling problems with uncertainty in processing times is based on sensitivity analysis, see [5,8].

2. Notation and definitions

Suppose that there is a set J of jobs that have to be processed on a single machine, $|J| = n$, $n \geqslant 2$. The machine cannot process more than one job at any time. Suppose that for any job $j \in J$, its processing time is uncertain. An assignment of specific values p_j to processing times of jobs $j \in J$ is called a *scenario*. Let S denote a given set of possible scenarios. Let Π be the set of all possible orderings of the jobs from J, $|\Pi| = n!$; elements of Π will be called *permutations*. For any scenario $s = \{p_j^{(s)}, j \in J\} \in S$ and any permutation $x = (j_1, j_2, \ldots, j_n) \in \Pi$, the corresponding *total flow time* is

$$F_s(x) = p_{j_1}^{(s)} \cdot n + p_{j_2}^{(s)} \cdot (n-1) + \cdots + p_{j_{n-1}}^{(s)} \cdot 2 + p_{j_n}^{(s)}$$

and for a job j_i value $p_{j_i}^{(s)}(n - i + 1)$ is called the *flow time contribution* of job j_i for permutation x under scenario s and is denoted $C_{j_i}(x, s)$. For a specific scenario $s \in S$, consider the problem

Problem OPT(s): Minimize $\{F_s(x) | x \in \Pi\}$.

Problem OPT(s) is the problem of finding the order of processing the jobs by the machine with the objective to minimize the total flow time under scenario s (the sum of the completion times of all the jobs, assuming that the machine starts working at time 0). It is well known that Problem OPT(s) can be solved in $O(n \log n)$ time using the following rule: order the jobs according to nondecreasing values of processing times $p_j^{(s)}$ (the shortest processing time (SPT) rule).

Let F_s^* denote the optimum objective function value for Problem OPT(s). For any $x \in \Pi$ and $s \in S$, value $R(x, s) = F_s(x) - F_s^*$ is called the *regret* for x under scenario s. For any $x \in \Pi$, value

$$Z(x) = \max_{s \in S} R(x, s). \tag{1}$$

is called a *worst-case regret* for X. The minmax regret version of Problem OPT is

Problem ROB: Minimize $\{Z(x) | x \in \Pi\}$.

For any $x, y \in \Pi$, let

$$r(x, y) = \max_{s \in S} (F_s(x) - F_s(y)). \tag{2}$$

Then $Z(x)$ can be written as

$$Z(x) = \max_{y \in \Pi} r(x, y) \tag{3}$$

or

$$Z(x) = \max_{s \in S} \max_{y \in \Pi} (F_s(x) - F_s(y)). \tag{4}$$

An optimal permutation to the right-hand side of (3) is called a *worst-case alternative* for x. An optimal scenario to the right-hand side of (1) is called a *worst-case scenario* for x. An optimal scenario to the right-hand side of (2) is called a *worst-case scenario for x with respect to y*. An optimal solution $(\hat{s}, \hat{y})$ to the right-hand side of (4) is called a *worst-case pair for x*. Observe that if $(\hat{s}, \hat{y})$ is a worst-case pair for x, then $\hat{s}$ is a worst-case scenario and $\hat{y}$ is a worst-case alternative for x.

Suppose that for every job $j \in J$, two integer numbers p_j^-, p_j^+ are given, $p_j^- \leqslant p_j^+$. The numbers p_j^-, p_j^+ represent the lower and the upper bounds on the processing time of job j. It is assumed that the processing time of job j can take on any real value from its *interval of uncertainty* $[p_j^-, p_j^+]$, regardless of the values taken by the processing times of other jobs. Thus, the set of scenarios S is the Cartesian product of the intervals of uncertainty $[p_j^-, p_j^+]$, $j \in J$. For any integers $k, t, k \leqslant t$, let $[k : t]$ denote the set of integers between k and t (including k, t).

Observe that if a scenario $s' = \{p_j^{(s')}, j \in J\}$ is obtained from a scenario $s = \{p_j^{(s)}, j \in J\}$ by adding the same constant to processing times of all jobs, then value $F_{s'}(x) - F_s(x)$ does not depend on x. This implies the following:

Lemma 1. *If the same constant is added to all numbers p_j^-, p_j^+, $j \in J$, value $Z(x)$ does not change for any $x \in \Pi$.*

Notice that we do not assume numbers p_j^-, p_j^+, $j \in J$ to be nonnegative; of course, in practice processing times of jobs are always nonnegative, but as can be seen from Lemma 1, for any instance with general values p_j^-, p_j^+, $j \in J$ there is an equivalent instance with nonnegative values p_j^-, p_j^+, $j \in J$, and vice versa. We allow negative endpoints of intervals of uncertainty because this will be convenient for presentation.

A scenario $s = \{p_j^{(s)}, j \in J\}$ such that $p_j^{(s)} \in \{p_j^-, p_j^+\}$ for all $j \in J$ is called an *extreme* scenario. A worst-case scenario for x which is also an extreme scenario will be called a *worst-case extreme scenario for x*. If for some jobs $j', j'', p_{j'}^- \leqslant p_{j''}^-$ and $p_{j'}^+ \leqslant p_{j''}^+$, we say that job j' *dominates* job j''. The following three results were obtained in [3].

Statement 1 (*Daniels and Kouvelis [3]*). For any permutation $x \in \Pi$, there always exists a worst-case extreme scenario.

Statement 2 (*Daniels and Kouvelis [3]*). For any $x \in \Pi$, value $Z(x)$ can be obtained in polynomial time (by matching techniques).

Statement 3 (*Daniels and Kouvelis [3]*). Suppose that job j' dominates job j'', $x \in \Pi$ is an optimal permutation for Problem ROB, and j'' precedes j' in x.Then switching the positions of jobs j' and j'' will result in another optimal permutation for Problem ROB.

Statement 3 implies a weaker statement that if job j' dominates job j'', then there exists an optimal permutation where j' precedes j''.

3. Complexity results

*Problem ROB*1 is the special case of Problem ROB where all intervals of uncertainty have the same center, that is, $(p_j^- + p_j^+)/2$ is the same for all $j \in J$.

For any jobs $j_1, j_2 \in J$, we say that job j_1 is *wider* than job j_2 if the interval of uncertainty for job j_2 is a proper subset of the interval of uncertainty for job j_1, that is, $p_{j_1}^- \leqslant p_{j_2}^-$, $p_{j_1}^+ \geqslant p_{j_2}^+$, and at least one of the inequalities is strict. For any job $j \in J$ and a permutation $x \in \Pi$, let $x(j)$ denote the position of job j in the permutation x (that is, according to permutation x, $x(j) - 1$ jobs are performed before job j is performed). For any $j \in J$ and $x \in \Pi$, let $q(x, j) = \min\{n - x(j), x(j) - 1\}$. (Then jobs with larger values of $q(x, j)$ are closer to the "center" of permutation x, where the "center" of permutation x is the job at the position $(n + 1)/2$ if n is odd, and an imaginary job located between the positions $n/2$, $(n/2) + 1$ if n is even.) A permutation $x \in \Pi$ is called *uniform* if for any $j', j'' \in J$, if j' is wider than j'', then $q(x, j') \geqslant q(x, j'')$.

Theorem 1. *If the number of jobs n is even, then any uniform permutation is an optimal solution to Problem ROB*1 (*and therefore Problem ROB*1 *with even number of jobs is solvable in* $O(n \log n)$ *time*).

Theorem 2. *Problem ROB*1 *with odd number of jobs is NP-hard.*

Proofs of Theorems 1 and 2 will be presented in Section 5 and are based on the structural result obtained in Section 4.

Theorem 3. *Problem ROB is NP-hard*; *it remains NP-hard even if the number of jobs is even.*

Proof. Theorem 2 implies that Problem ROB with odd number of jobs is NP-hard. NP-hardness of Problem ROB with even number of jobs can be proved by a simple reduction from the problem with odd number of jobs. Consider an instance I1 with odd number of jobs, and let p' be an integer number that is smaller than all lower bounds of uncertainty intervals of instance I1. Add a new job with the interval of uncertainty $[p', p']$ obtaining an instance I2 with even number of jobs. It is straightforward to see that this new job is scheduled first in any optimal solution to Problem OPT(s) for any scenario s, and in any optimal solution to Problem ROB (instance I2). Thus, the optimal order of scheduling other jobs in an optimal solution to the instance I2 is optimal for the instance I1, and the optimal objective values are the same for both instances. The theorem is proved. □

The next lemma shows that without loss of generality we can assume that all numbers $p_j^-, p_j^+, j \in J$ are distinct (that is, no two of them are equal).

Lemma 2. *For any instance of Problem ROB*, *it is possible to modify the bounds of intervals of uncertainty so that they become distinct and any optimal permutation for the obtained instance is optimal for the original instance. Such modification can be done in polynomial time and the maximum absolute value of the new* (*integer*) *bounds is bounded by the maximum absolute value of the old* (*integer*) *bounds multiplied by a polynomial function of n. The same holds for Problem ROB*1.

Proof. See Appendix. □

In the remainder of the paper, unless stated otherwise, we make the following assumptions.

Assumption 1. All numbers $p_j^-, p_j^+, j \in J$ are distinct integers.

Assumption 2. When we consider Problem ROB1, we assume that the center of each interval of uncertainty is 0, that is, $p_j^+ = A_j$, $p_j^- = -A_j$, $A_j \geqslant 0$ for all $j \in J$. We denote $A = \sum_{j \in J} A_j$.

According to Lemmas 1 and 2, these assumptions are made without loss of generality.

4. A characterization of worst-case extreme scenarios for uniform permutations for problem ROB1

Consider Problem ROB1 (with Assumption 2).

Theorem 4. *Let α be a uniform permutation*, *and suppose that the jobs from J are identified with their positions in the permutation α*, *that is*, *for any $i \in [1:n]$*, *job i is the ith job in the permutation α. Let $\hat{s} = (p_1, \ldots, p_n)$ be a worst-case extreme scenario for α. Then $\hat{s} = (p_1, \ldots, p_n)$ has the following properties*:

(a) *If n is even*, *$n = 2k$*, *then $p_i = A_i$, $i \in [1:k]$*; *$p_i = -A_i$, $i \in [k+1, 2k]$.*
(b) *If n is odd*, *$n = 2k+1$*, *then $p_i = A_i$, $i \in [1:k]$*; *$p_i = -A_i$, $i \in [k+2:2k]$* (*observe that the value of p_{k+1} is not specified here*).

Proof. Suppose that $\hat{s} = (p_1, \ldots, p_n)$ is a worst-case extreme scenario for α, and suppose that $\hat{s}$ does not satisfy conditions (a),(b) of the theorem. Let t be the smallest value of $q(\alpha, j)$ among all jobs j that do not satisfy conditions (a),(b) under scenario $\hat{s}$. (That is, either job $t+1$ or job $n-t$ is the farthest job from the center of the permutation α that does not satisfy conditions (a),(b)). Clearly $t+1 \leqslant n/2$. We need to consider three cases.

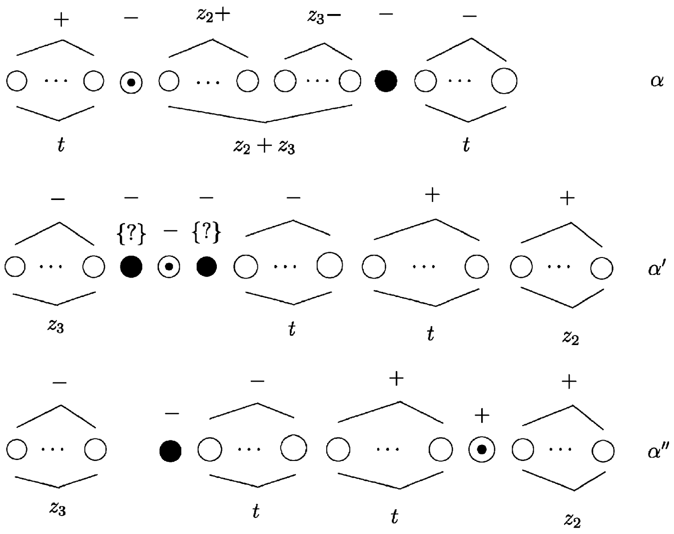

Fig. 1. Illustration for the proof of Theorem 4, Case 1.

Case 1: $p_i = A_i$, $i \in [1:t]$; $p_{t+1} = -A_{t+1}$; $p_{n+1-i} = -A_{n+1-i}$, $i \in [1:t]$; $p_{n-t} = -A_{n-t}$.

The structure of the permutation α under scenario $\hat{s}$ is represented in Fig. 1. Jobs $n-t$ and $t+1$ are symmetrical about the center of α. Let J_1 be the set of the first t jobs in α, J_2 (J_3) be the set of the jobs that are performed between job $t+1$ and job $n-t$ in α and whose processing times under scenario $\hat{s}$ are equal to upper (lower) bounds of the corresponding intervals of uncertainty, J_4 be the set of t last jobs in α. Let $|J_2| = z_2$, $|J_3| = z_3$; clearly $z_2 + z_3 = n - 2t - 2$. (Observe that the jobs from J_2 (J_3) are not necessarily contiguous in α, as may seem from Fig. 1.)

Let α' be the permutation optimal for Problem OPT($\hat{s}$). This permutation under scenario $\hat{s}$ is also represented in Fig. 1. In the permutation α', first the z_3 jobs from J_3 are performed in the order of increasing values $(-A_j)$, then jobs $t+1$ and $n-t$ in one of the two possible orders (depending on which job is wider), then the t jobs from J_4 in the same order as in α, then the t jobs from J_1 in the same order as in α, then the z_2 jobs from J_2 in the order of increasing values A_j.

Let α'' be the permutation optimal for Problem OPT($\hat{s}'$) where scenario $\hat{s}'$ is obtained from scenario $\hat{s}$ by changing the processing time of job $t+1$ from $-A_{t+1}$ to A_{t+1}. This permutation under scenario $\hat{s}'$ is also represented in Fig. 1. In the permutation α'', first we perform the z_3 jobs from J_3 in the same order as in α', then job $n-t$, then the t jobs from J_4 in the same order as in α and α', then the t jobs from J_1 in the same order as in α and α', then job $t+1$, and then the z_2 jobs from J_2 in the same order as in α'.

Thus, when permutation α' is replaced with permutation α'', the jobs from J_2 and J_3 do not change their positions; The jobs from $J_1 \bigcup J_4$ move one position to the left; Job $n-t$ moves one position to the left if $-A_{t+1} < -A_{n-t}$, and does not change its position otherwise; Job $t+1$ moves to the position $n - z_2$.

The flow time contribution of job $t+1$ for permutation α under scenario $\hat{s}$ is

$$C_{t+1}(\alpha, \hat{s}) = -A_{t+1}(1 + z_2 + z_3 + 1 + t).$$

The flow time contribution of job $t+1$ for permutation α' under scenario $\hat{s}$ is

$$C_{t+1}(\alpha', \hat{s}) = -A_{t+1}(1 + b + 2t + z_2)$$

where $b = 1$ if $-A_{t+1} < -A_{n-t}$, and $b = 0$ otherwise; b reflects the potential impact of job $n-t$, which is positioned in α' to the right of job $t+1$ if $-A_{t+1} < -A_{n-t}$.

Now consider scenario $\hat{s}'$. We have

$$C_{t+1}(\alpha, \hat{s}') = A_{t+1}(1 + z_2 + z_3 + 1 + t),$$

$$C_{t+1}(\alpha'', \hat{s}') = A_{t+1}(z_2 + 1).$$

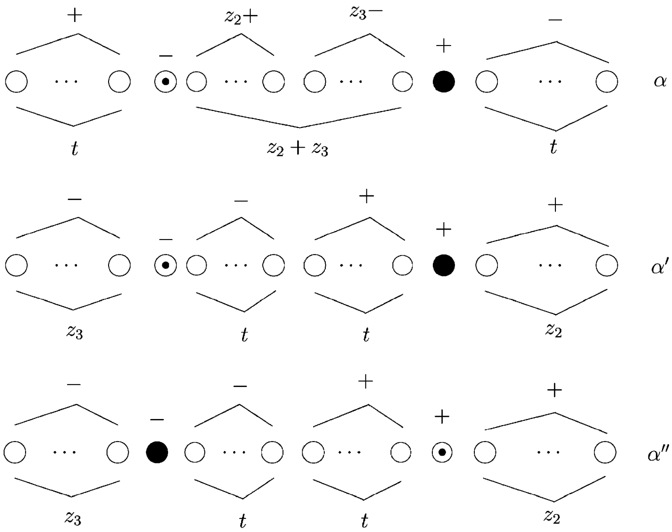

Fig. 2. Illustration for the proof of Theorem 4, Case 2.

Thus, the difference of the flow time contributions of job $t+1$ for permutations α and α' under scenario $\hat{s}$ is

$$C_{t+1}(\alpha, \hat{s}) - C_{t+1}(\alpha', \hat{s}) = -A_{t+1}(1 + z_2 + z_3 + 1 + t - 1 - b - 2t - z_2) = -A_{t+1}(z_3 + 1 - b - t).$$

The difference of flow time contributions of job $t+1$ for permutations α and α'' under scenario $\hat{s}'$ is

$$C_{t+1}(\alpha, \hat{s}') - C_{t+1}(\alpha'', \hat{s}') = A_{t+1}(1 + z_2 + z_3 + 1 + t - z_2 - 1) = A_{t+1}(1 + z_3 + t).$$

So, when scenario $\hat{s}$ is replaced with scenario $\hat{s}'$ and permutation α' is replaced with α'', the difference of flow time contributions of job $t+1$ for permutations α and α' is increased by the quantity $A_{t+1}(1+z_3+t)-(-A_{t+1})(z_3+1-b-t)= A_{t+1}(2z_3+2-b) \geqslant A_{t+1}$.

The difference of flow time contributions of jobs from $J_2 \bigcup J_3$ for permutations α and α' under scenario $\hat{s}$ will not change when scenario $\hat{s}$ is replaced with $\hat{s}'$ and permutation α' is replaced with α'' (the positions of these jobs in α' and α'' are the same).

Value $C_i(\alpha, \hat{s}) - C_i(\alpha', \hat{s})$ for any job i from J_1 (respectively, from J_4) will decrease by A_i (respectively, will increase by A_i) when scenario $\hat{s}$ is replaced with $\hat{s}'$ and permutation α' is replaced with α'' (as the job moves one position to the left when α' is replaced with α'').

Value $C_{n-t}(\alpha, \hat{s}) - C_{n-t}(\alpha', \hat{s})$ for job $n-t$ will increase by A_{n-t} when $\hat{s}$ is replaced with $\hat{s}'$ and α' is replaced with α'' if $-A_{t+1} < -A_{n-t}$ and will not change otherwise; therefore, it will increase by $b \cdot A_{n-t}$.

Thus,

$$R(\alpha, \hat{s}') - R(\alpha, \hat{s}) \geqslant A_{t+1} - \sum_{i=1}^{t} A_i + \sum_{i=n-t+1}^{n} A_i + b \cdot A_{n-t}.$$

Since permutation α is uniform and taking into account Assumption 1,

$$A_{t-1} < A_{n-(t-1)}; \quad A_{t-2} < A_{n-(t-2)}; \cdots; \quad A_1 < A_{n-1}.$$

Thus, $\sum_{i=1}^{t} A_i - \sum_{i=n-t+1}^{n} A_i - b \cdot A_{n-t} \leqslant A_t$, and therefore $R(\alpha, \hat{s}') - R(\alpha, \hat{s}) \geqslant A_{t+1} - A_t > 0$. This is a contradiction with the assumption that $\hat{s}$ is a worst-case scenario for α.

Case 2: $p_i = A_i, i \in [1:t]$; $p_{t+1} = -A_{t+1}$; $p_{n+1-i} = -A_{n+1-i}, i \in [1:t]$; $p_{n-t} = A_{n-t}$.

Let α' be the permutation optimal for Problem OPT($\hat{s}$). The permutations α and α' under scenario $\hat{s}$ are represented in Fig. 2. Let $J_1, J_2, J_3, J_4, z_2, z_3$ have the same meaning as in Case 1. In the permutation α', first the z_3 jobs from J_3 are performed in the order of increasing values $(-A_j)$, then job $t+1$, then the t jobs from J_4 in the same order as in

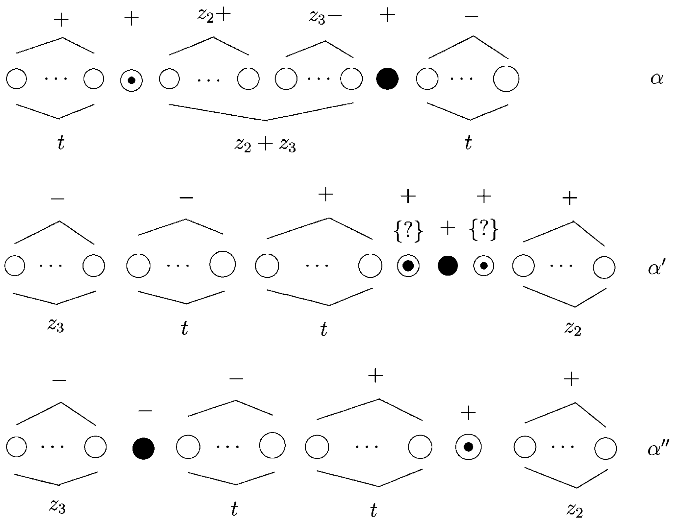

Fig. 3. Illustration for the proof of Theorem 4, Case 3.

α, then the t jobs from J_1 in the same order as in α, then job $n-t$, then the z_2 jobs from J_2 in the order of increasing values A_j.

Let α'' be the permutation optimal for Problem OPT($\hat{s}''$), where scenario $\hat{s}''$ is obtained from scenario $\hat{s}$ by changing the processing time of job $t+1$ from $-A_{t+1}$ to A_{t+1}, and changing the processing time of job $n-t$ from A_{n-t} to $-A_{n-t}$. This permutation under scenario $\hat{s}''$ is also represented in Fig. 2. The description of the permutation α'' is the same as in Case 1.

Thus, when permutation α' is replaced with permutation α'', jobs from J_1, J_2, J_3, J_4 do not change their positions (only jobs $t+1$ and $n-t$ are switched).

We have

$$C_{t+1}(\alpha,\hat{s}) - C_{t+1}(\alpha',\hat{s}) = -A_{t+1}(2+z_2+z_3+t) - (-A_{t+1})(2+2t+z_2) = -A_{t+1}(z_3-t),$$

$$C_{t+1}(\alpha,\hat{s}'') - C_{t+1}(\alpha'',\hat{s}'') = A_{t+1}(2+z_2+z_3+t) - A_{t+1}(z_2+1) = A_{t+1}(z_3+t+1).$$

Thus, value $C_{t+1}(\alpha,\hat{s}) - C_{t+1}(\alpha',\hat{s})$ increases by $A_{t+1}(2z_3+1) \geqslant A_{t+1} > 0$ when $\hat{s}$ is replaced with $\hat{s}''$ and α' is replaced with α''.

Now,

$$C_{n-t}(\alpha,\hat{s}) - C_{n-t}(\alpha',\hat{s}) = A_{n-t}(t+1) - A_{n-t}(z_2+1) = A_{n-t}(t-z_2),$$

$$C_{n-t}(\alpha,\hat{s}'') - C_{n-t}(\alpha'',\hat{s}'') = (-A_{n-t})(t+1) - (-A_{n-t})(1+2t+1+z_2) = A_{n-t}(t+1+z_2).$$

Thus, value $C_{n-t}(\alpha,\hat{s}) - C_{n-t}(\alpha',\hat{s})$ increases by $A_{n-t}(2z_2+1) \geqslant A_{n-t} > 0$ when $\hat{s}$ is replaced with $\hat{s}''$ and α' is replaced with α''. Since only jobs $t+1$ and $n-t$ change their positions when α' is replaced with α'', we have $R(\alpha,\hat{s}'') - R(\alpha,\hat{s}) \geqslant A_{t+1} + A_{n-t} > 0$. This is a contradiction with the assumption that $\hat{s}$ is a worst-case scenario for α.

Case 3: $p_i = A_i$, $i \in [1:t]$; $p_{t+1} = A_{t+1}$; $p_{n+1-i} = -A_{n+1-i}$, $i \in [1:t]$; $p_{n-t} = A_{n-t}$.

This case is completely analogous (symmetric) to Case 1 (see Fig. 3) and is considered in the same way. The theorem is proved. □

5. Proofs of Theorems 1 and 2

In this section, we consider Problem ROB1 (with Assumption 2).

Lemma 3. *Suppose that n is even, $n = 2k$. Then for any uniform permutation x, $Z(x) = kA$.*

Proof. Let x be a uniform permutation. Let x_1 be the sequence of the first k jobs of x and x_2 be the sequence of the last k jobs of x, i.e. $x = x_1x_2$. Let $\hat{s}$ be a worst-case scenario for x. Then according to Theorem 4, $p_j^{(\hat{s})} = p_j^+ = A_j$, $j \in x_1$, and $p_j^{(\hat{s})} = p_j^- = -A_j$, $j \in x_2$. Let $B = \sum_{j\in x_1} A_j$. Then $\sum_{j\in x_1} p_j^{(\hat{s})} = B$, $\sum_{j\in x_2} p_j^{(\hat{s})} = -(A - B)$. Observe that permutation $x' = x_2x_1$ is optimal for Problem OPT($\hat{s}$) (since x is uniform). When $x = x_1x_2$ is replaced with $x' = x_2x_1$, jobs from x_1 are shifted k positions to the right, and jobs from x_2 are shifted k positions tothe left. Therefore, $Z(x) = F_{\hat{s}}(x) - F_{\hat{s}}(x') = kB + (A - B)k = Ak$. □

Proof of Theorem 1. Consider an arbitrary permutation $x = x_1x_2 \in \Pi$, where x_1 and x_2 have the same meaning as in the proof of Lemma 3. Consider also permutation $y = x_2x_1$. Consider the scenario s' that assigns processing times $p_j^+ = A_j$ to all $j \in x_1$ and processing times $p_j^- = -A_j$ to all $j \in x_2$. Let $B = \sum_{j\in x_1} A_j$, then $\sum_{j\in x_1} p_j^{(s')} = B$, $\sum_{j\in x_2} p_j^{(s')} = -(A - B)$. Then $Z(x) = \max_{s\in S} R(x, s) \geqslant R(x, s') = F_{s'}(x) - F^*_{s'} \geqslant F_{s'}(x) - F_{s'}(y) = Bk + (A - B)k = Ak$. Therefore, any permutation x such that $Z(x) = Ak$ is optimal for Problem ROB1. Taking into account Lemma 3, the theorem is proved. □

Proof of Theorem 2. Suppose we are given a set M of $2k$ positive integers $m_1, m_2, \ldots, m_{2k}$ listed in a nondecreasing order. A partition of M into two sets M_1, M_2 is called *balanced* if $\sum_{m_i\in M_1} m_i = \sum_{m_i\in M_2} m_i$, and is called *uniform* if for each $i \in [1 : k]$ exactly one of the numbers m_{2i-1}, m_{2i} belongs to M_1 and the other belongs to M_2. (To explain the terminology, we note that uniform partitions will correspond to uniform permutations in our proof.) Consider the following

Problem UBP (uniform balanced partition). Given a set M of $2k$ positive integers, is there a uniform balanced partition for this set?

Problem UBP is known to be NP-complete [4]. For our proof, we will need a subclass of instances of Problem UBP such that for each instance of this subclass any balanced partition is also uniform, and such that the restriction of Problem UBP to this subclass remains NP-complete. We now describe such a subclass.

We say that the set M has an exponential growth property (EG-property) if

$$m_{2i} > m_1 + m_2 + \cdots + m_{2i-2} + Tk, \quad i \in [1 : k],$$

where $T = \max_{i\in[1:k]}(m_{2i} - m_{2i-1})$.

Claim 1. *If M has the EG-property, then any balanced partition of M is also uniform.*

To prove Claim 1, suppose $\langle M_1, M_2\rangle$ is a balanced partition of M that is not uniform, and let i' be the largest value of i such that m_{2i-1}, m_{2i} both belong to the same part of the partition (say M_1, without loss of generality). Then for any $i > i'$, exactly one of the numbers m_{2i-1}, m_{2i} belongs to M_1. Let $r_1 = \sum\{m_j \,|\, j \geqslant 2i' + 1, m_j \in M_1\}$, $r_2 = \sum\{m_j \,|\, j \geqslant 2i' + 1, m_j \in M_2\}$. Clearly $|r_1 - r_2| \leqslant Tk$. Now it is straightforward to see that the sum of all numbers in M_1 is greater than the sum of all numbers in M_2, which contradicts the assumption that the partition is balanced. Claim 1 is proved.

Let Problem UBPEG denote Problem UBP restricted to instances with the EG-property.

Claim 2. *Problem UBPEG is NP-complete.*

Indeed, given an instance $M = \{m_1, m_2, \ldots, m_{2k}\}$ of Problem UBP, an equivalent instance of Problem UBPEG can be obtained as follows. Let $f_1 = Tk + 1$, and define recursively $f_i = 3f_{i-1}$, $i \in [2 : k]$. Let us obtain $M' = \{m'_1, m'_2, \ldots, m'_{2k}\}$ from M by adding $f_i - m_{2i}$ to m_{2i-1} and m_{2i} for each $i \in [1 : k]$, that is $m'_{2i-1} = m_{2i-1} + f_i - m_{2i}$, $m'_{2i} = f_i$. Using induction, we can see that M' has the EG-property. (Indeed, $m'_2 = f_1 > Tk$; now, if for some i $m'_{2i} > \sum_{j\in[1:2i-2]} m'_j + Tk$, then $m'_{2i+2} = 3m'_{2i} > m'_{2i} + m'_{2i-1} + \sum_{j\in[1:2i-2]} m'_j + Tk = \sum_{j\in[1:2i]} m'_j + Tk$. It is also straightforward to verify that the sequence $m'_1, m'_2, \ldots, m'_{2k}$ is nondecreasing.) Now, since M' is obtained from M by adding the same number $f_i - m_{2i}$ to both m_{2i-1} and m_{2i} for each $i \in [1 : k]$, M' has a balanced uniform partition if and only if M has such a partition. The length of binary encoding of M' is polynomial in the length of binary encoding of M. Claim 2 follows immediately.

Now we prove NP-hardness of Problem ROB1 with odd number of jobs using a reduction from Problem UBPEG.

Consider an instance $M = \{m_1, m_2, \dots, m_{2k}\}$ of Problem UBPEG. Denote $\mu = m_1 + \cdots + m_{2k}$. Elements of M are listed in the order of nondecreasing values. We can assume that all numbers in M are distinct (if $m_{2i-1} = m_{2i}$ for some i, these numbers can be deleted from M without affecting existence of a uniform balanced partition; also recall that M has the EG-property). The corresponding instance of Problem ROB1 has $2k + 1$ jobs with intervals of uncertainty $[-m_1, m_1], \dots, [-m_{2k}, m_{2k}], [-d, d]$, where $d = 4k \cdot \mu + 1$, with obvious correspondence between $2k$ jobs and elements of M.

Claim 3. *The optimal objective function value for the obtained instance of Problem ROB*1 *is greater than* $dk+\mu k+\mu/2$ *if and only if the answer to the original instance of Problem UBPEG is "No".*

Let us prove Claim 3 (this will complete the proof of Theorem 2).

(1) Suppose there exists a uniform balanced partition $\langle M_1, M_2\rangle$ of M. For the obtained instance of Problem ROB1, consider the uniform permutation $x=x_1dx_2$, where jobs from x_1 correspond to elements of M_1, jobs from x_2 correspond to elements of M_2, d is the job with the interval of uncertainty $[-d, d]$ (the requirement that the permutation is uniform uniquely defines the order of jobs in the sequences x_1 and x_2). According to Theorem 4, $Z(x)=\max\{R(x, s'), R(x, s'')\}$, where $s' = s_1^+d^+s_2^-$, $s'' = s_1^+d^-s_2^-$ (the notation $s_1^+d^+s_2^-, s_1^+d^-s_2^-$ is self-explanatory; for example, $s_1^+d^-s_2^-$ denotes the scenario where jobs from x_1 (from x_2) have processing times equal to the upper bounds (lower bounds) of the corresponding intervals of uncertainty, and job d has the processing time equal to the corresponding lower bound $(-d)$).

Observe that permutation x_2x_1d is optimal for Problem OPT(s'), and permutation dx_2x_1 is optimal for Problem OPT(s'').

Permutation x_2x_1d is obtained from permutation x_1dx_2 by shifting the sequence x_1d by k positions to the right and the sequence x_2 by $k + 1$ positions to the left. Let $B = \sum_{m_i\in M_1} m_i$, then $\sum_{m_i\in M_2} m_i = \mu - B$, and

$$R(x, s') = Bk + dk + (\mu - B)(k + 1) = dk + \mu k + \mu - B.$$

Permutation dx_2x_1 is obtained from x_1dx_2 by shifting sequence dx_2 by k positions to the left and sequence x_1 by $k + 1$ positions to the right. Thus,

$$R(x, s'') = dk + (\mu - B)k + B(k + 1) = dk + \mu k + B.$$

Since $\mu - B = B = \mu/2$, we have $Z(x) = dk + \mu k + \mu/2$. Thus, $\min_{x\in\Pi} Z(x) \leqslant dk + \mu k + \mu/2$.

(2) Suppose now that there is no uniform balanced partition for M. Let x be an optimal permutation for the obtained instance of Problem ROB1. Then in x job d must have position $k+1$ (at the center of x), that is, $x=x_1dx_2$, $|x_1|=|x_2|=k$. (Indeed, if y is an arbitrary permutation where job d has position $k+1$ and $\tilde{y}$ is any permutation where job d has another position, then it is straightforward to verify that $Z(y)\leqslant dk+\mu\cdot 2k$, $Z(\tilde{y})\geqslant d(k+1)-\mu\cdot 2k$ and therefore $Z(\tilde{y}) > Z(y)$.) Let jobs from J be identified with their positions in the permutation x, that is, job i is the ith job in x. Consider the following two scenarios: $s' = s_1^+d^+s_2^-$, $s'' = s_1^+d^-s_2^-$ (i.e., scenario $s_1^+d^-s_2^-$ assigns upper (lower) bounds to jobs from x_1 (x_2) and the lower bound to job d). Consider permutation $y' = x_2x_1d$; it is obtained from permutation x_1dx_2 by shifting sequence x_1d by k positions to the right, and sequence x_2 by $k + 1$ positions to the left. Consider also permutation $y'' = dx_2x_1$; it is obtained from permutation x_1dx_2 by shifting sequence dx_2 by k positions to the left, and sequence x_1 by $k + 1$positions to the right. Let $B' = \sum_{i=1}^{k} p_i^+$, then $-\sum_{i=k+2}^{2k+1} p_i^- = \mu - B'$. Then

$$Z(x) = \max_{s\in S} R(x, s) \geqslant \max\{R(x, s'), R(x, s'')\},$$

$$R(x, s') = F_{s'}(x) - F_{s'}^* \geqslant F_{s'}(x) - F_{s'}(y') = B'k + dk + (\mu - B')(k + 1) = dk + \mu k + \mu - B',$$

$$R(x, s'') = F_{s''}(x) - F_{s''}^* \geqslant F_{s''}(x) - F_{s''}(y'') = (\mu - B')k + dk + B'(k + 1) = dk + \mu k + B',$$

$$Z(x) \geqslant dk + \mu k + \max\{\mu - B', B'\} > dk + \mu k + \mu/2.$$

Claim 3 is proved. This completes the proof of Theorem 2. □

6. Some other polynomially solvable cases

Consider Problem ROB without Assumption 1. A job $j \in J$ is called *certain* if $p_j^- = p_j^+$, and is called *uncertain* if $p_j^- < p_j^+$. Let d denote the number of uncertain jobs. Statement 3 implies that we can order the certain jobs according to nondecreasing values $p_j^- = p_j^+$, and consider only permutations where the positions of the certain jobs are consistent with this order. Since there are $O(n^d)$ such permutations, and taking into account Statement 2, we have that if d is fixed, Problem ROB can be solved in polynomial time.

Another special case where polynomial solvability follows immediately from Statement 3 is the case where the dominance relation is complete, that is, for any two jobs one dominates the other. This case can be generalized to the nonpreemptive multimachine setting with *uniform* machines, even if they have different speeds. Suppose that there are p machines; each machine can perform any job, but the machines have different speeds $r_1 \geqslant r_2 \geqslant \cdots \geqslant r_p$. Processing times of jobs (bounds of intervals of uncertainty) are stated in terms of a machine with unit speed, that is, if under scenario s job j has processing time $p_j^{(s)}$, this means that on machine i the processing time is $p_j^{(s)}/r_i$. Consider the set $\{1/r_1, 1/r_2, \ldots, 1/r_p, 2/r_1, 2/r_2, \ldots, 2/r_p, 3/r_1, \ldots\}$; let $t_1, t_2, \ldots$ be the sequence of the numbers in this set in a nondecreasing order. Each element of this sequence can be interpreted as a possible position for a job; assigning a job to a position $t_j = k/r_i$ means that the job is scheduled on machine i as kth last job on this machine. The formulations of Problem OPT(s) and ROB for the multimachine setting are analogous to the single-machine case with the only difference that now feasible solutions are assignments of jobs to positions $t_1, t_2, \ldots$. (e.g. in Problem OPT(s) it is required to minimize the total flow time; the flow time contribution of job j assigned to position k/r_i is $p_j^{(s)}(k/r_i)$.)

Let $T = (t_1, \ldots, t_n)$. (Note that if $p = 1$, then T is the sequence of the n possible positions for the jobs on the single machine in the order opposite to the order of performance.) It is known that for any scenario s, there is an optimal solution to Problem OPT(s) where all jobs are assigned to the positions from T, and an optimal assignment can be obtained as follows: number the jobs according to nonincreasing values of $p_j^{(s)}$, and assign job j to the position t_j, $j \in [1 : n]$ [2]. Using a standard interchange argument, it is straightforward to show that there is an optimal solution to Problem ROB where all jobs are assigned to the positions from T; thus, we can consider only such assignments as feasible solutions. Now, an argument similar to that used in [3] to prove Statement 3 shows that if job j' dominates job j'' and x is an assignment of jobs to positions from T optimal for Problem ROB, and if the position of job j' in T according to the assignment x precedes the position of job j'', then switching the positions of jobs j' and j'' will result in another optimal assignment. Thus, in the special case where the dominance relation is complete, an optimal assignment can be obtained in $O(n \log n)$ time by assigning the jobs to positions from T in the reversed order of dominance.

As a direction for further research, we note that it is not known whether Problem ROB is strongly NP-hard or solvable in pseudopolynomial time (in our NP-hardness proof we used a reduction from Problem UBP which is only weakly NP-hard).

Acknowledgement

Research of Igor Averbakh was supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Appendix

Proof of Lemma 2. Consider an instance of Problem ROB with jobs $j_1, \ldots, j_n$. Introduce new bounds of intervals of uncertainty as follows:

$$\tilde{p}_{j_i}^+ = p_{j_i}^+ + \theta \cdot i, \quad \tilde{p}_{j_i}^- = p_{j_i}^- - \theta \cdot i, \quad i \in [1 : n],$$

where $\theta = 1/(2(n+1)^4)$. All new bounds are distinct (but not integer). Consider an arbitrary extreme scenario s for the original instance, and let $\tilde{s}$ be the corresponding extreme scenario for the obtained instance ("corresponding" means that $\tilde{s}$ and s take upper bounds of intervals of uncertainty on the same jobs). Clearly for any permutation $x \in \Pi$, $|F_s(x) - F_{\tilde{s}}(x)| \leqslant 1/(2(n+1))$; therefore $|R(x,s) - R(x,\tilde{s})| \leqslant 1/(n+1) \leqslant \frac{1}{3}$, and thus any optimal permutation for the obtained instance is also optimal for the original instance (remember that in the original instance all bounds are integer). By multiplying all new bounds by $2(n+1)^4$ we obtain an equivalent instance with distinct integer bounds (values $F_s(x)$ for extreme scenarios s will increase by a factor of $2(n+1)^4$; the same about values $R(x, s)$). Also, if centers of all intervals of uncertainty for the original instance are equal, the same holds for the obtained instance. □

References

[1] I. Averbakh, On the complexity of a class of combinatorial optimization problems with uncertainty, Math. Program. 90 (2001) 263–272.
[2] P. Brucker, Scheduling Algorithms, Springer, Berlin, 2001.
[3] R. Daniels, P. Kouvelis, Robust scheduling to hedge against processing time uncertainty in single-stage production, Management Sci. 41 (2) (1995) 363–376.
[4] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[5] N. Hall, M. Posner, Sensitivity analysis for scheduling problems, J. Sched. 7 (2004) 49–83.
[6] P. Kouvelis, G. Yu, Robust Discrete Optimization and its Applications, Kluwer Academic Publications, Boston, 1997.
[7] J.M. Mulvey, R.J. Vanderbei, S.A. Zenious, Robust optimization of large scale systems, Oper. Res. 43 (2) (1995) 264–281.
[8] B. Penz, C. Rapine, D. Trystram, Sensitivity analysis of scheduling problems, European J. Oper. Res. 134 (2001) 606–615.