



HAL
open science

Single machine robust scheduling with budgeted uncertainty

Marin Bougeret, Artur Alves Pessoa, Michael Poss

► **To cite this version:**

Marin Bougeret, Artur Alves Pessoa, Michael Poss. Single machine robust scheduling with budgeted uncertainty. *Operations Research Letters*, 2023, 51 (2), pp.137-141. 10.1016/j.orl.2023.01.007 . hal-03926286

HAL Id: hal-03926286

<https://hal.science/hal-03926286>

Submitted on 6 Jan 2023

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.

Highlights

Single machine robust scheduling with budgeted uncertainty

Marin Bougeret, Artur Pessoa, Michael Poss

- Minimizing the number of tardy jobs with uncertain durations is \mathcal{NP} -hard.
- Minimizing the weighted number of tardy jobs is pseudo-polynomial.
- Worst-case scenario is better than average scenario for budget uncertainty.

Single machine robust scheduling with budgeted uncertainty

Marin Bougeret

LIRMM, University of Montpellier, CNRS, rue Ada 156, 34095, Montpellier, France

Artur Pessoa

Production Engineering Department, Universidade Federal Fluminense, Rua Passo da Patria, 156, 309-D, Sao Domingos, Niteroi, 24.210-240, Brazil

Michael Poss

LIRMM, University of Montpellier, CNRS, rue Ada 156, 34095, Montpellier, France

Abstract

We consider robust single machine scheduling problems. First, we prove that with uncertain processing times, minimizing the number of tardy jobs is \mathcal{NP} -hard. Second, we show that the weighted variant of the problem has the same complexity as the nominal counterpart whenever only the weights are uncertain. Last, we provide approximation algorithms for the problems minimizing the weighted sum of completion times. Noticeably, our algorithms extend to more general robust combinatorial optimization problems with cost uncertainty, such as max-cut.

Keywords: robust scheduling, \mathcal{NP} -hardness, approximation algorithms

1. Introduction

Classical optimization models suppose perfect information over all parameters. This can lead to optimal solutions having poor performance when the actual parameters deviate, even by a small amount, from the predictions used in the optimization model. Different frameworks have been proposed to overcome this issue, among which Robust Optimization which tackles the uncertainty by providing a set of possible values for these parameters, and considering the worst outcome over that set. In this paper, we consider optimization problems where only the objective function is affected by the uncertainty, so that the robust counterpart amounts to find min-max solutions.

Initially, Robust Optimization considered discrete uncertainty sets without particular structure [11], in particular, in the context of robust scheduling problems [1, 6, 10, 13]. A major step forward has been undertaken by Bertsimas and Sim [2], who studied the robust counterparts of combinatorial optimization problems with linear cost functions. Their model considers that in any scenario, at most Γ coefficients of the cost function deviate, the other ones taking their nominal values. Letting \bar{u}_i denoting the nominal value of uncertain parameter u_i and \hat{u}_i being its deviation, and $\Gamma \in \mathbb{Z}_+$, they define

$$\mathcal{U}^\Gamma \equiv \left\{ u \in \mathbb{R}^n : u_j = \bar{u}_j + \delta_j \hat{u}_j, j \in \{1, \dots, n\}, \delta \in \{0, 1\}^n, \sum_{j=1}^n \delta_j \leq \Gamma \right\}.$$

One useful property of \mathcal{U}^Γ is that exponentially many scenarios are described in a compact manner. More importantly, Bertsimas and Sim show that the solvability and approximability of the deterministic problem are transferred to its robust counterpart, as the latter amounts to solve a linear number of deterministic problems. Their result has later been generalized to some extent to problems with integer variables [8] and to more general uncertainty sets [15].

In what follows, we consider robust scheduling problems. The objective functions of scheduling problems are usually not compatible with the result from [2]. For instance, the robust counterpart of the problem that minimizes the weighted sum of completion times is \mathcal{NP} -hard in the strong sense [5], contrasting with the polynomial deterministic counterpart. The objective of this paper is to pursue the work initiated in [5]: studying the complexity and approximability of the robust counterpart of basic single machine scheduling problems under budgeted uncertainty. Next, we define more precisely the problems we study in this paper.

Problem definitions. Consider a set of n jobs, each of which has non-negative processing time p_j , weight w_j , and due-date d_j . A schedule σ is a permutation of $\{1, \dots, n\}$, where $\sigma(j)$ denotes the position of job j , without idle time between jobs. As this paper will rely on mathematical programming formulations, we alternatively characterize any schedule σ by the binary vector $x \in \{0, 1\}^{n^2}$ such that $\sigma(i) \leq \sigma(j) \Leftrightarrow x_{ij} = 1$. In particular, we set $x_{ii} = 1$ for each $i = 1, \dots, n$. Denoting by $X^\sigma \subseteq \{0, 1\}^{n^2}$ the set of all schedules, and by $g : X^\sigma \rightarrow \mathbb{R}_+$ the objective function, the robust scheduling problems under study can be cast as combinatorial optimization problems of the form

$$\min_{x \in X^\sigma} g(x).$$

We consider herein problems with four different objective functions g . Let $C_j(x, p) = \sum_{i=1}^n p_i x_{ij}$ denote the completion time of job j for the schedule corresponding to x and processing times p , and let the function $U_j(x, p)$ be equal to 1 if job j is tardy ($C_j(x, p) > d_j$), 0 otherwise. We consider here the problems minimizing either the sum of completion times $C_j(x, p)$ or the number of tardy jobs, possibly considering weights w . Borrowing the three-field notation $\alpha|\beta|\gamma$ from [9], we define more precisely:

- the robust counterpart of $1||\sum U_j$ with processing time uncertainty, formally stated as $\min_{x \in X^\sigma} \max_{p \in \mathcal{U}^\Gamma} \sum_{j \in \mathcal{J}} U_j(x, p)$;
- the robust counterpart of $1||\sum w_j U_j$ with weight uncertainty, formally stated as $\min_{x \in X^\sigma} \max_{w \in \mathcal{U}^\Gamma} \sum_{j \in \mathcal{J}} w_j U_j(x, p)$;
- the robust counterparts of $1||\sum w_j C_j$ with either processing time uncertainty or weight uncertainty, formally stated as $\min_{x \in X^\sigma} \max_{p \in \mathcal{U}^\Gamma} \sum_{j \in \mathcal{J}} w_j C_j(x, p)$ and $\min_{x \in X^\sigma} \max_{w \in \mathcal{U}^\Gamma} \sum_{j \in \mathcal{J}} w_j C_j(x, p)$, respectively.

Related works. Tadayon and Smith [16] considered the robust counterparts of problems $1||\sum C_j$, $1||\sum w_j C_j$, and $1||\sum U_j$ for uncertain processing times, using uncertainty set \mathcal{U}^Γ as well as related sets. They provided mixed-integer linear programming formulations for some of these problems, and proved the polynomiality of the robust counterpart of $1||\sum C_j$ for one of these related sets. Sticking to \mathcal{U}^Γ , the authors of [5] prove that the robust counterpart of $1||\sum C_j$ with uncertain processing times is polynomially solvable. They further provide constant factor approximation algorithms for scheduling problems with parallel machines, later improved in [4]. The related bin-packing problem has also been investigated in [3], where the authors provide constant factor approximation algorithms.

Contributions and structure of the paper. We first prove in Section 2 that the robust counterpart of $1||\sum U_j$ with uncertain processing times is \mathcal{NP} -hard in the ordinary sense by using a reduction from the partition problem. The result contrasts with the polynomiality of the deterministic counterpart [14]. Then, in Section 3 we prove that the robust counterpart of $1||\sum w_j U_j$ with uncertain weights can be solved in pseudo-polynomial time by employing the deterministic algorithm as a subroutine [12]. For the robust counterparts of $1||\sum w_j C_j$ with either uncertain processing times or uncertain weights, we present in Section 4 an approximation algorithm that amounts to solve exactly the nominal problem with a vector of processing times (resp. weights) equal to $\bar{p} + \hat{p}$ (resp. $\bar{w} + \hat{w}$). Interestingly, the results presented in Sections 3 and 4 extend to robust combinatorial optimization problems more general than scheduling.

2. Hardness of the robust counterpart of $1||\sum U_j$

We study in this section the complexity of optimization problem

$$\min_{x \in X^\sigma} \max_{p \in \mathcal{U}^\Gamma} \sum_{j \in \mathcal{J}} U_j(x, p). \quad (1)$$

The hardness proof of this section relies on a reduction from the partition problem.

Problem 1. *PARTITION PROBLEM (PP): given N positive integer values a_1, \dots, a_N such that $\sum_{i=1}^N a_i = 2A$, does there exist a subset S of $\{1, \dots, N\}$ satisfying $\sum_{i \in S} a_i = A$?*

Let us introduce first a small result that will be instrumental in connecting subsets S of $\{1, \dots, N\}$ to certain types of schedules for (1).

Lemma 1. *Consider an instance of problem (1) of size $n \geq 2N$ where $d_{2j-1} = d_{2j}$, for $j = 1, \dots, N$,*

$$\bar{p}_{2j-1} + \bar{p}_{2j} + \sum_{j' \in \{1, \dots, j-1\} \setminus \{k\}} \min\{\bar{p}_{2j'-1}, \bar{p}_{2j'}\} > d_{2j}, \quad (2)$$

for $j, k = 1, \dots, N$ ($k < j$), and

$$\bar{p}_1 + \bar{p}_2 > d_1. \quad (3)$$

Given a feasible schedule x for this instance where at most N jobs are late, we have that, for each $j \in \{1, \dots, N\}$ and each $p \in \mathcal{U}^\Gamma$, exactly one of the two jobs $2j-1$ and $2j$ is late.

Proof. First note that, since at most n jobs are late in x , it is enough to prove that at least one job is late for each pair of jobs $\{2j-1, 2j\}$, $j = 1, \dots, N$. Given N , we prove the above by induction on the value of each $n \leq N/2$. For the base case ($n = 1$), (3) implies immediately that job 1 or job 2 is late (or both), regardless from the value of $p \in \mathcal{U}^\Gamma$. Now, assume by the inductive hypothesis that the result is valid for any value of n smaller than a given n' and consider $n = n' + 1$ together with a schedule x . Let x' be the schedule x restricted to jobs $\{1, \dots, n\} \setminus \{2N-1, 2N\}$. The hypothesis implies that at least $n' = n-1$ jobs are late in x' . Then, schedule x is obtained from x' by inserting jobs $2N-1$ and $2N$. Constraints (2) imply that there is no subset of N jobs that include both jobs $2N-1$ and $2N$ and that can all be scheduled on time. This is again true for any $p \in \mathcal{U}^\Gamma$, proving the result. \square

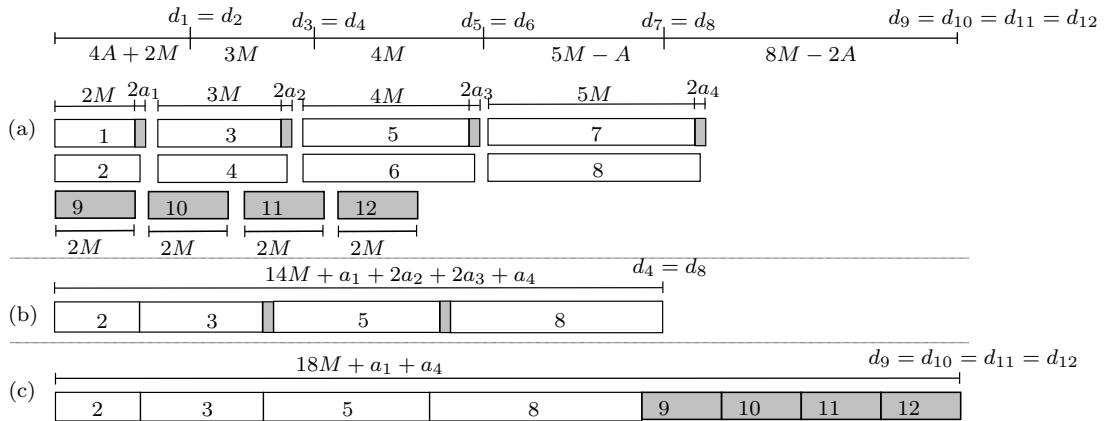


Figure 1: The instance of problem (1) obtained via reduction from a PARTITION instance with $N = 4$.

Theorem 1. *Problem (1) is \mathcal{NP} -hard.*

Proof. Let (N, a) be an instance of PP. Let $2A = \sum_{i=1}^N a_i$, and M be a sufficiently large number that we define later. We set $n = 3N$, $\Gamma = N$, and, for $j = 1, \dots, N$, $\bar{p}_{2j-1} = (j+1)M$, $\hat{p}_{2j-1} = 2a_j$, $\bar{p}_{2j} = (j+1)M + a_j$, $\hat{p}_{2j} = 0$, $\bar{p}_{2N+j} = 0$, and $\hat{p}_{2N+j} = 2M$. Further, we define the due dates

$$d_{2j-1} = d_{2j} = 4A + \sum_{j'=1}^j (j'+1)M$$

for $j = 1, \dots, N-1$,

$$d_{2N-1} = d_{2N} = 3A + \sum_{j=1}^N (j+1)M$$

and,

$$d_{2N+1} = \dots = d_{3N} = A + \sum_{j=1}^N (j+1)M + NM$$

Figure 1(a) illustrates this construction for $N = 4$. Each job is represented by one or two rectangles, where the white ones represent non-zero mean processing times and the gray ones the non-zero deviations. The rectangle widths are proportional to the time values they represent. Job indices are indicated inside the rectangles.

We show next that the answer to PP is *yes* if and only if there is a feasible schedule for (1) with cost at most N . Given a solution S to PP, we show how to construct a feasible schedule for (1) with cost at most N . For each $j = 1, \dots, N$, if $j \in S$ we schedule job $2j$ in position j ; otherwise, if $j \notin S$, we schedule job $2j-1$ in position j . Jobs $2N+1, \dots, 3N$ are scheduled in positions $N+1, \dots, 2N$, and the remaining jobs from $\{1, \dots, 2N\}$ are scheduled at the very end. This solution is depicted in Figures 1(b) and 1(c) for two different scenarios $p \in \mathcal{U}^\Gamma$. Figure 1(b) illustrates a worst-case scenario for meeting the due date d_8 (since all jobs scheduled before that deadline deviate). We see that the deadline constraint on $d_{2N} = d_{2N-1}$ for such a worst-case scenario is

$$\sum_{j=1}^N (j+1)M + \sum_{j \in S} a_j + \sum_{j \notin S} 2a_j = \sum_{j=1}^N (j+1)M + 2A + \sum_{j \notin S} a_j \leq \sum_{j=1}^N (j+1)M + 3A,$$

which is satisfied because S is a solution to PP (so $\sum_{j \notin S} a_j = A$). Notice that deadlines d_2, d_3, d_5 and $d_9 = d_{10} = d_{11} = d_{12}$ are also satisfied for that scenario, so at most 4 jobs are late for that scenario. Figure 1(c) illustrates a worst-case scenario for meeting the due dates $d_9 = d_{10} = d_{11} = d_{12}$. We see that the deadline constraints on $d_{2N+1} = \dots = d_{3N}$ for such a worst-case scenario is

$$\sum_{j=1}^N (j+1)M + \sum_{j \in S} a_j + \sum_{j=1}^N M \leq \sum_{j=1}^N (j+1)M + A + NM,$$

which is again satisfied because S is a solution to PP. For that scenario, one readily verifies that only the jobs scheduled at the very end are late, so this solution satisfies exactly $2N$ deadlines.

Conversely, we show next that any feasible schedule for (1) with cost at most N leads to a solution S to PP. This argument relies on Lemma 1, so let us verify that (2) and (3) are satisfied. We see that (3) is satisfied if and only if $M > 2A - a_1/2$. Moreover, for $j = 2, \dots, N$, the value of k that minimizes the left-hand side of (2) (among the considered ones) is $j-1$. In this case, (2) is satisfied if and only if $M > 4A - a_j$, for $j = 1, \dots, N-1$, and if and only if $M > 3A - a_j$, for $j = N$. Thus, taking $M = 4A$, Lemma 1 holds. As at most N job may be late, for each $j = 1, \dots, N$, the lemma implies that either job $2j-1$ or job $2j$ will be scheduled. Let S be the set of indices j such that job $2j$ is scheduled. We are left to prove that $\sum_{i \in S} a_i = A$. By the above discussion, this is required to meet the due dates d_{2N-1}, d_{2N} , and d_{2N+1}, \dots, d_{3N} , concluding the proof. \square

3. Solving the robust counterpart of $1 \parallel \sum w_j U_j$ with uncertain weights

The main purpose of this section is to provide efficient algorithms for solving problem

$$\min_{x \in X^\sigma} \max_{w \in \mathcal{U}^\Gamma} \sum_{j \in \mathcal{J}} w_j U_j(x, p). \quad (4)$$

We do so by considering a slightly more general class of robust combinatorial optimization problem

$$\min_{x \in X} \max_{u \in \mathcal{U}^\Gamma} f(x, u), \quad (5)$$

where $X \subseteq \{0, 1\}^m$, and the objective function f is assumed to be linear in u , that is, there exists functions f_1, \dots, f_n such that $f(x, u) = \sum_{i=1}^n f_i(x) u_i$. We further assume in this section more specifically that there are positive numbers $F_i > 0$ such that, for each $i \in 1, \dots, n$,

$$f_i : X \rightarrow \{0, F_i\}. \quad (6)$$

We see that problem (4) is a special case of problem (5) under condition (6) by choosing $X = X^\sigma$, $f_i(x) = U_j(x, p)$ (recall p is a given and fixed tuple in this problem), and identifying u with w . Another example of problem (5) under condition (6) is the problem that searches for a maximum cut in a weighted graph $G = (V, E, w)$, where the weights w are uncertain and belong to \mathcal{U}^Γ , which can be formulated as

$$\min_{x \in \{0, 1\}^{|V|}} \max_{u \in \mathcal{U}^\Gamma} \sum_{e=\{i, j\} \in E} w_e (x_i(1 - x_j) + x_j(1 - x_i)),$$

where binary variables x_i , $i \in V$, describe the of partition V in two sets according to the chosen cut.

The following result is an extension of Theorem 3 from [2] to the setting introduced above.

Proposition 1. *Consider an instance I of problem (5) and suppose that (6) holds. Let us introduce $\hat{u}_{n+1} = 0$ and $F_{n+1} = 0$, and for $\ell = 1, \dots, n + 1$, define $\tilde{u}_i^\ell = \max(0, \hat{u}_i - \hat{u}_\ell \frac{F_i}{F_\ell})$. The optimal solution to I can be obtained by solving*

$$\min_{\ell=1, \dots, n+1} \Gamma \hat{u}_\ell F_\ell + \min_{x \in X} f(x, \bar{u} + \tilde{u}^\ell).$$

Proof. The inner maximization problem in (5) can be developed as

$$\sum_{i=1}^n \bar{u}_i f_i(x) + \max_{\delta} \left\{ \sum_{i=1}^n \delta_i \hat{u}_i f_i(x) \mid \sum_{i=1}^n \delta_i \leq \Gamma, 0 \leq \delta_i \leq 1 \right\},$$

where the constraints on δ lead to an integral polytope so $\delta \in \{0, 1\}^n$ in any maximizer. Then, we dualize the maximization over δ and obtain

$$\begin{aligned} & \min_{y, \theta} \left\{ \Gamma \theta + \sum_{i=1}^n y_i \mid \theta + y_i \geq \hat{u}_i f_i(x), i = 1, \dots, n, y, \theta \geq 0 \right\} \\ &= \min_{y, \theta} \left\{ \Gamma \theta + \sum_{i=1}^n \max(0, \hat{u}_i f_i(x) - \theta) \mid \theta \geq 0 \right\} \quad [y = \max(0, \hat{u}_i f_i(x) - \theta)] \\ &= \min_{\theta} \left\{ \Gamma \theta + \sum_{i=1}^n f_i(x) \max(0, \hat{u}_i - \theta / F_i) \mid \theta \geq 0 \right\} \quad [\text{using (6)}] \\ &= \min_{\theta \in \{0, \hat{u}_1 F_1, \dots, \hat{u}_n F_n\}} \Gamma \theta + \sum_{i=1}^n f_i(x) \max(0, \hat{u}_i - \theta / F_i). \end{aligned}$$

where the last inequality holds because for any $x \in X$, the function being minimized is convex and piece-wise linear, so its minimum is always reached at one of its knickpoints. We obtain

$$\begin{aligned}
\min_{x \in X} \max_{u \in \mathcal{U}^\Gamma} f(x, u) &= \min_{x \in X} \sum_{i=1}^n \bar{u}_i f_i(x) + \min_{\theta \in \{0, \hat{u}_1 F_1, \dots, \hat{u}_n F_n\}} \Gamma \theta + \sum_{i=1}^n f_i(x) \max(0, \hat{u}_i - \theta / F_i) \\
&= \min_{\theta \in \{0, \hat{u}_1 F_1, \dots, \hat{u}_n F_n\}} \Gamma \theta + \min_{x \in X} \sum_{i=1}^n \bar{u}_i f_i(x) + \sum_{i=1}^n f_i(x) \max(0, \hat{u}_i - \theta / F_i) \\
&= \min_{\ell=1, \dots, n+1} \Gamma \hat{u}_\ell F_\ell + \min_{x \in X} \sum_{i=1}^n f_i(x) \left(\bar{u}_i + \max \left(0, \hat{u}_i - \hat{u}_\ell \frac{F_\ell}{F_i} \right) \right) \\
&= \min_{\ell=1, \dots, n+1} \Gamma \hat{u}_\ell F_\ell + \min_{x \in X} \sum_{i=1}^n f_i(x) (\bar{u}_i + \tilde{u}_i^\ell) \\
&= \min_{\ell=1, \dots, n+1} \Gamma \hat{u}_\ell F_\ell + \min_{x \in X} f(x, \bar{u} + \tilde{u}^\ell).
\end{aligned}$$

□

Corollary 1. *Problem (4) can be solved in $O(n^2 \max_j d_j)$.*

Proof. Given the vector of processing times p , we define $f(x, u) = \sum_{j=1}^n u_j U_j(x, p)$, and $X = X^\sigma$. We have $f_j(x) = U_j(x, p)$, which satisfies (6). Applying Proposition 1 to this instance of problem (5), we end up solving $n + 1$ problems $1 \parallel \sum w_j U_j$. Each of these problems $1 \parallel \sum w_j U_j$ corresponds to a different vector of weights $w = \bar{u} + \tilde{u}^\ell$ and can be solved in $O(n \max_j d_j)$ [12]. □

Algorithm 1: Approximation algorithm for (5) with assumption (6).

```

1 for  $\ell = 1, \dots, n + 1$  do compute an  $\alpha$ -approximate solution  $x_A^\ell$  to  $\min_{x \in X} f(x, \bar{u} + \tilde{u}^\ell)$  using  $A$ ;
2 for  $\ell = 1, \dots, n + 1$  do  $Z_A^\ell = \max_{u \in \mathcal{U}^\Gamma} f(x_A^\ell, u)$ ;
3 let  $\ell^* = \arg \min_{\ell=1, \dots, n+1} Z_A^\ell$ ;
4  $x^* = x_A^{\ell^*}$  and  $Z^* = Z^{\ell^*}$ ;
return  $(x^*, Z^*)$ 

```

Given $u \geq 0$, we define the nominal counterpart of (5) as

$$\min_{x \in X} f(x, u), \tag{7}$$

and we define the optimal objective values of (7) and (5) as opt^{nom} and opt^{rob} , respectively. We explain next how we can also extend the approximation result from [2] and leverage Proposition 1 to transfer approximation ratios of opt^{nom} to opt^{rob} . Given $\alpha \geq 1$, suppose that we have an algorithm A that returns a solution $x_A \in X$ with objective Z_A that satisfies $Z_A \leq \alpha \times opt^{nom}$. Then, we can use Algorithm 1 to obtain an α -approximation algorithm for (5).

Observation 1. *Recall that $\max_{u \in \mathcal{U}^\Gamma} f(x_A^\ell, u) = \max_{u \in \mathcal{U}^\Gamma} \sum_{i=1}^n f_i(x_A^\ell) u_i$, so step 2 of Algorithm 1 can be solved in $O(n \log(n))$ by sorting $(f_1(x_A^\ell) u_1, \dots, f_n(x_A^\ell) u_n)$ and taking the largest Γ indices. Thus, Algorithm 1 runs in polynomial time if A runs in polynomial time.*

The details of the proof of the following result are omitted for brevity, we refer the interested reader to [2].

Proposition 2. *Algorithm 1 returns a solution $x^* \in X$ with objective Z^* that satisfies $Z^* \leq \alpha \times opt^{rob}$.*

Proof. Follows closely the lines of the proof of Proposition 4 from [2], replacing Theorem 3 with Proposition 1. \square

Corollary 2. *There is a Fully Polynomial Time Approximation Scheme (FPTAS) for problem (4).*

Proof. Let A be an FPTAS for $1||\sum w_j U_j$ (e.g. [7]). Observation 1 and Proposition 2 imply that Algorithm 1 using A is an FPTAS for problem (4). \square

4. A general $\frac{n}{\Gamma}$ -approximation algorithm

The main purpose of this section is to provide approximation algorithms for problems

$$\min_{x \in X^\sigma} \max_{w \in \mathcal{U}^\Gamma} \sum_{j \in \mathcal{J}} w_j C_j(x, p) \text{ and } \min_{x \in X^\sigma} \max_{p \in \mathcal{U}^\Gamma} \sum_{j \in \mathcal{J}} w_j C_j(x, p). \quad (8)$$

As in the previous section, we reach our goal by considering the more general problem $\min_{x \in X} \max_{u \in \mathcal{U}^\Gamma} f(x, u)$, with f linear in u , but without the supplementary condition (6). We denote again the optimal objective values of (7) and (5) as opt^{nom} and opt^{rob} , respectively. We also assume that we have an algorithm A that returns a solution $x_A \in X$ with objective Z_A that satisfies $Z_A \leq \alpha \times opt^{nom}$.

Algorithm 2: Approximation algorithm for (5).

- 1 let $u^{max} = (\bar{u}_1 + \hat{u}_1, \dots, \bar{u}_n + \hat{u}_n)$;
 - 2 compute an α -approximate solution x^* to $\min_{x \in X} f(x, u^{max})$ using A ;
 - 3 $Z^* = \max_{u \in \mathcal{U}^\Gamma} f(x^*, u)$;
- return** (x^*, Z^*)
-

Proposition 3. *Algorithm 2 returns a solution $x^* \in X$ with objective Z^* that satisfies $Z^* \leq \alpha \times \frac{n}{\Gamma} \times opt^{rob}$.*

Proof. The proof is done in two steps. First, let us define $F(x) = f(x, u^{max})$ and $f^\Gamma(x) = \max_{u \in \mathcal{U}^\Gamma} f(x, u)$. The following holds:

$$\frac{f^\Gamma(x)}{F(x)} = \frac{\sum_{i=1}^n f_i(x) \bar{u}_i + \max_{\substack{0 \leq \delta \leq 1 \\ \sum \delta \leq \Gamma}} \sum_{i=1}^n f_i(x) \hat{u}_i \delta_i}{\sum_{i=1}^n f_i(x) \bar{u}_i + \sum_{i=1}^n f_i(x) \hat{u}_i} \geq \frac{\max_{\substack{0 \leq \delta \leq 1 \\ \sum \delta \leq \Gamma}} \sum_{i=1}^n f_i(x) \hat{u}_i \delta_i}{\sum_{i=1}^n f_i(x) \hat{u}_i} \geq \frac{\Gamma}{n}. \quad (9)$$

Let us now prove that x^* is an $\alpha \times \frac{n}{\Gamma}$ -approximate solution for (5). Let x^F and x^{rob} be optimal solutions of $\min_{x \in X} F(x)$ and problem (5), respectively. We have $Z^* = f^\Gamma(x^*) \leq F(x^*) \leq \alpha \times F(x^F) \leq \alpha \times F(x^{rob}) \leq \alpha \times \frac{n}{\Gamma} \times f^\Gamma(x^{rob}) = \alpha \times \frac{n}{\Gamma} \times opt^{rob}$, where the last inequality follows from (9). \square

Notice that the ratio proposed in Proposition 3 compares favorably to the well-known ratio $|\mathcal{U}^\Gamma|$ that is achieved by the average scenario [11], obtained by using $u^{mean} = \frac{1}{|\mathcal{U}^\Gamma|} \sum_{u \in \mathcal{U}^\Gamma} u$ instead of u^{max} in Algorithm 2.

Corollary 3. *There is a polynomial time $\frac{n}{\Gamma}$ -approximation algorithm for problems (8).*

Proof. The nominal problem $1||\sum w_j C_j$ can be formulated as $\min_{x \in X^\sigma} \sum_{j=1}^n \sum_{i=1}^n p_i w_j x_{ij}$. Thus, the objective function is bilinear in w and p , and both robust counterparts are special cases of problem (5). As in Observation 1, we can find in polynomial time an exact solution of the deterministic problem for the single scenario u^{max} (which corresponds to a scenario where every job deviates). \square

We present below an example in which the above approach yields a ratio of at least \sqrt{n} for $\Gamma = 1$. Hence, the ratio α_{Γ}^n , which is equal to n in the case of problems (8) with $\Gamma = 1$, is not far from being tight.

Example 1. Let $n \in \mathbb{N}$. Let $\bar{p}_1 = \bar{p}_2 = \dots = \bar{p}_n = 0$. Let $\frac{\hat{p}_1}{w_1} = \frac{\hat{p}_2}{w_2} = \dots = \frac{\hat{p}_n}{w_n} = 1$. Hence, for all j , $\hat{p}_j = w_j$. Let $w_i \sum_{j=i}^n w_j = 1$ for all i . This determines exactly all w_i , hence all \hat{p}_i (for instance, $w_n = 1$). On this instance, our algorithm can output any schedule. Let us compare two of them: $\sigma_1 = (1, 2, \dots, n)$ and $\sigma_2 = (n, n-1, \dots, 1)$, and let x_1 and x_2 be the corresponding binary vectors. We denote by $\delta_x(i)$ the effect on objective function of the deviation of job i . We have $\delta_{x_1}(i) = \hat{p}_i \sum_{j=i}^n w_j = w_i \sum_{j=i}^n w_j = 1$ for all i . Hence, $F(x_1) = 1$. Let us now consider $F(x_2)$ and notice that $F(x_2) = \max_i(\delta_{x_2}(i)) \geq \delta_{x_2}(1)$. We will show that $\delta_{x_2}(1) \geq \sqrt{n}$. We note $\Omega_i = \sum_{j=i}^n w_j$. From $w_i \sum_{j=i}^n w_j = 1$ we get $w_i^2 + w_i \sum_{j=i+1}^n w_j - 1 = 0$, hence $w_i = \frac{1}{2} \left(-\Omega_{i+1} + \sqrt{\Omega_{i+1}^2 + 4} \right)$. Since $\Omega_i = w_i + \Omega_{i+1}$, we get $\Omega_i = \frac{1}{2} \left(\Omega_{i+1} + \sqrt{\Omega_{i+1}^2 + 4} \right)$. Let $u_i = \Omega_{n-i}^2$. Then $u_i = \left(\frac{1}{2} \left(\Omega_{n-i+1} + \sqrt{\Omega_{n-i+1}^2 + 4} \right) \right)^2 = \frac{1}{4} \left(\sqrt{u_{i-1}} + \sqrt{u_{i-1} + 4} \right)^2 = \frac{1}{4} \left(4 + 2u_{i-1} + 2\sqrt{u_{i-1}}\sqrt{u_{i-1} + 4} \right) \geq \frac{1}{4} (4 + 4u_{i-1}) = u_{i-1} + 1$. Since $u_0 = \Omega_n^2 = 1$, we have $u_i \geq i + 1$ for all i . Hence $\Omega_{n-i} \geq \sqrt{i + 1}$, so $\Omega_1 \geq \sqrt{n}$. We obtain $\delta_{x_2}(1) = \hat{p}_n \Omega_1 \geq \sqrt{n}$, so that $F(x_2) \geq \sqrt{n}$.

Notice that an alternative polynomial time 2-approximation algorithm can be obtained for problems (8) by following the approach depicted in Theorem 5 from [13], dualizing the robust constraint to avoid enumerating all $|\mathcal{U}^\Gamma|$ elements of \mathcal{U}^Γ . Thus, Algorithm 2 is better than the one adapted from [13] when $\Gamma > \frac{n}{2}$. Furthermore, Algorithm 2 is much more general than the ad-hoc procedure adapted from [13] and can be applied to any robust combinatorial optimization problem of the form (5).

5. Conclusion

The results presented here raise two natural questions. First, given the hardness of the robust counterpart of $1||\sum U_j$, one could dig into approximability of the problem. Second, one could dig into approximability results when the due-dates are uncertain, for the robust counterpart of both $1||\sum U_j$ and $1||\sum w_j U_j$.

Acknowledgement

The authors are grateful to the referees, the comments of which improved the readability of the paper. This research benefited from the support of the ANR project ROBUST [ANR-16-CE40-0018].

References

- [1] M. A. Aloulou and F. D. Croce. Complexity of single machine scheduling problems under scenario-based uncertainty. *Operations Research Letters*, 36(3):338 – 342, 2008.
- [2] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Math. Program.*, 98(1-3):49–71, 2003.
- [3] M. Bougeret, G. Dósa, N. Goldberg, and M. Poss. Constant-ratio approximation for robust bin packing with budgeted uncertainty. *SIAM J. Discret. Math.*, 36(4):2534–2552, 2022.
- [4] M. Bougeret, K. Jansen, M. Poss, and L. Rohwedder. Approximation results for makespan minimization with budgeted uncertainty. *Theory Comput. Syst.*, 65(6):903–915, 2021.
- [5] M. Bougeret, A. A. Pessoa, and M. Poss. Robust scheduling with budgeted uncertainty. *Discret. Appl. Math.*, 261:93–107, 2019.
- [6] R. L. Daniels and P. Kouvelis. Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science*, 41(2):pp. 363–376, 1995.
- [7] G. V. Gens and E. V. Levner. Fast approximation algorithm for job sequencing with deadlines. *Discrete Applied Mathematics*, 3(4):313–318, 1981.
- [8] K. Goetzmann, S. Stiller, and C. Telha. Optimization over integers with robustness in cost and few constraints. In *Approximation and Online Algorithms - 9th International Workshop, WAOA 2011, Saarbrücken, Germany, September 8-9, 2011, Revised Selected Papers*, pages 89–101, 2011.
- [9] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. R. Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5:287–326, 1979.
- [10] A. Kasperski, A. Kurpisz, and P. Zielinski. Approximating a two-machine flow shop scheduling under discrete scenario uncertainty. *European Journal of Operational Research*, 217(1):36–43, 2012.

- [11] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 1997.
- [12] E. L. Lawler and J. M. Moore. A functional equation and its application to resource allocation and sequencing problems. *Management Science*, 16:77–84, 1969.
- [13] M. Mastrolilli, N. Mutsanas, and O. Svensson. Approximating single machine scheduling with scenarios. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 153–164. Springer Berlin Heidelberg, 2008.
- [14] J. M. Moore. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management science*, 15(1):102–109, 1968.
- [15] M. Poss. Robust combinatorial optimization with knapsack uncertainty. *Discret. Optim.*, 27:88–102, 2018.
- [16] B. Tadayon and J. C. Smith. Algorithms and complexity analysis for robust single-machine scheduling problems. *J. Scheduling*, 18(6):575–592, 2015.