Robust single machine scheduling with a flexible maintenance activity

Robust single machine scheduling with a flexible maintenance activity

Paolo Detti * Gaia Nicosia † Andrea Pacifici ‡
Garazi Zabalo Manrique de Lara §

May 6, 2021

Abstract

In this paper, we address a problem arising in a manufacturing environment concerning the joint scheduling of multiple jobs and a maintenance activity on a single machine. Such activity must be processed within a given time window and its non-deterministic duration takes values in a given interval.

We seek job schedules which are *robust* to any possible changes in the maintenance activity duration. We consider makespan and total completion time objectives under four different robustness criteria. We discuss a few properties and the complexity of finding robust schedules for the resulting eight problem scenarios.

For the case of total completion time objective and maximum absolute regret criterion, we design and test exact and heuristic algorithms. The results of an extensive computational campaign, performed for assessing the performance of the proposed solution approaches, are reported.

Keywords: Scheduling, flexible maintenance, robust optimization.

1 Introduction

In this paper, we address a relevant problem arising in a manufacturing environment concerning the joint scheduling of multiple jobs and of a *maintenance activity* on a single machine [18, 27]. The maintenance activity is *flexible*, meaning that it must be performed within a given time window. Furthermore, while the processing times of the jobs are deterministic, the maintenance duration is uncertain and can only be *estimated* at the time when the scheduling of the jobs is planned. Hence, possible deviations from its

*Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena, Via Roma 56, 53100 Italy, e-mail: detti@dii.unisi.it, tel.: +39 0577-234850 (1022), fax: +39 0577-233602

†Dipartimento di Ingegneria, Università degli Studi "Roma Tre", via della Vasca Navale 79, 00146 Rome, Italy, e-mail: nicosia@ing.uniroma3.it

‡Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università degli Studi di Roma "Tor Vergata", Via del Politecnico 1, 00133 Rome, Italy, e-mail: andrea.pacifici@uniroma2.it

§Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena, Via Roma 56, 53100 Italy, e-mail garazizml@gmail.com

nominal value may occur and the resulting schedule depends on the time span attained by the maintenance task.

The above situation may naturally arise, for instance, in manufacturing plants where a machining center requires the jobs to be processed in a given predefined sequence and a maintenance intervention has been planned in advance. Typically, such intervention has to be executed by an external service-company in a certain time-interval. The duration of the maintenance activity is not known until the service will release an estimate (based on monitoring the state of the production resource and/or a site inspection). In this case, the time required by the maintenance activity becomes apparent only *after* the job sequence has been set and, as a consequence, the actual schedule may be established only as a result of the latter time value. In this context, the problem is to determine a *job sequence* which is *robust* to any possible change in the duration of the maintenance activity. Two standard objectives are considered for our scheduling problem: makespan and total completion time of the jobs.

Several different versions of scheduling problems considering maintenance activities, which are usually modelled as machine unavailability, have been addressed in the literature. For a survey see [18]. For the total completion time objective function, deterministic versions of the problem, i.e., in which the maintenance duration is known, have been considered in, e.g., [1], [24] and [27]. In [1], the problem is proven to be $\mathcal{NP}$-hard. In [27], a dynamic programming and a branch-and-bound algorithms are proposed. The authors also show that the SPT (Shortest Processing Time) sequence[1] has a worst-case approximation ratio of 9/7. Recently, the problem solved in [27] has been addressed in [24], assuming a maintenance period related to the machine workload. In [4], the single machine scheduling problem with an operator non-availability period is considered, where the objective is to minimize the total completion time. The operator non-availability period is an open time interval in which a job may neither start nor complete. The problem is proven to be $\mathcal{NP}$-hard, and an algorithm with a tight worst-case ratio of 20/17 is presented.

For the makespan objective function, the deterministic version has been proven to be $\mathcal{NP}$-hard in [16]. Furthermore, Lee [16] showed that the Longest Processing Time rule has a tight worst-case ratio of 4/3, and He *et al.* [10] presented a FPTAS (Fully Polynomial Time Approximation Scheme).

A variant of the deterministic problem has been addressed in [3, 23], in which the machine is assumed to be stopped periodically for maintenance for a constant time during the scheduling period. More precisely, in [3], exact mixed integer programming models and a heuristic algorithm are provided, and in [23], it is shown that the worst-case performance bound of the heuristic algorithm presented in [3] is 2 and that there is no polynomial time approximation algorithm with a worst-case performance bound less than 2 unless $\mathcal{P} = \mathcal{NP}$.

In this paper, we focus on a robust scheduling problem in which the duration of the single maintenance activity is not known in advance. Many authors have addressed robust optimization problems in several fields in the last twenty years, in order to take care of incomplete or erroneous data through a *proactive* approach. In [20] a general framework

[1]In an SPT sequence the jobs are ordered according to non-decreasing values of their processing times.

for explicitly incorporating conflicting objectives and model robustness is proposed, while in [14] several robustness criteria are discussed.

In robust discrete optimization problems, usually one assumes that, due to the variability of some parameters, a set of possible *scenarios* is defined. The robust approach consists of minimizing the worst case "performance" of a solution over all scenarios. The performance, in turn, may be evaluated considering the actual value $f(x, s)$ of a solution x in scenario s or, its *regret* i.e., its deviation from an optimal solution value in that scenario.

On these grounds, different robustness measures have been proposed in the literature. Kouvelis and Yu [14] defined the following three measures to minimize: (1) the maximum absolute cost over all scenarios (*min-max* criterion), (2) the maximum deviation from optimality, i.e., *maximum absolute regret* criterion, and (3) the maximum relative deviation from optimality (*maximum relative regret* criterion). This is the case of [26], where the authors consider those three criteria for robustness in the context of single-machine scheduling. Moreover, in [25], Yager introduced the *Ordered Weighted Averaging* aggregation measure (OWA). Assigning weights to scenarios, OWA generalizes the traditional criteria used in decision making under uncertainty such as the maximum, minimum, average, median, or Hurwicz criterion.

Several works address robustness issues in scheduling problems. In [6], one of the first papers on this topic, the authors study the problem of minimizing the total completion time on a single machine, when processing times may vary in given intervals, and established several properties of robust schedules. Two alternative criteria of schedule robustness are considered focusing on a given schedule worst-case absolute or relative deviation from the optimum over all scenarios. For the same problem, a Mixed Integer Linear (MILP) formulation is proposed in [19] and exact solution methods are analyzed. The problem of finding the schedule minimizing the maximum regret on a single machine, with processing times varying in given intervals, has been shown to be $\mathcal{NP}$-hard in [15], while in [11] approximation algorithms achieving a constant approximation ratio are proposed for the same problem. More recently, the above problem is considered in the more general setting of total weighted completion time metric: the author proposes an exact algorithm for this case [22]. A slightly different single machine scheduling problem is addressed in [17], where job processing times are uncertain, there are sequence-dependent family setup times and the objective is to obtain robust sequences of job families and jobs so as minimize the absolute deviation of total completion time from the optimal solution under the worst-case scenario. The problem of finding the schedule minimizing the maximum regret on unrelated machines is also considered in [5]. The author proposes a polynomial time algorithm for determining the regret given a schedule and uses this result for deriving a MILP formulation for the general problem.

Differently from most of the robust scheduling problems addressed in the literature, in this paper the processing times of the jobs are deterministic and the uncertainty is relative to the duration of the maintenance activity, while the objective functions (i.e., makespan or total completion time minimization) only depend on the completion times of the jobs and not on that of the maintenance activity. Preliminary results of this work have been presented in [7] where the same model is used to describe an analogous problem in a multi-agent scheduling scenario, similar to, e.g., that in [2].

The article is organized as follows. In Section 2, we formally define the addressed robust scheduling problem and give a summary of the results. Sections 3 and 4 are devoted to present properties of robust schedules when minimizing the makespan and the total completion time objectives, respectively. Section 5 presents approximation results for the problem. In Section 6, exact and heuristic algorithms for the robust scheduling problem of minimizing the total completion time with the maximum absolute regret criterion are proposed and evaluated through an extensive computational campaign. Conclusions follow in Section 7.

2 Problem statement

Let $J = \{1, 2, \ldots, n\}$ be a set of n non-preemptive jobs with processing times p_j, $j = 1, \ldots, n$, available at time zero to be processed on a single machine. The machine cannot process any of the jobs during the execution of a maintenance activity, hereafter denoted as M, which must be performed within a predefined time window $[r, d]$. The duration of the maintenance activity M is uncertain and can be regarded as a random variate of a discrete aleatory variable P. A *scenario* s corresponds to one realization of P and we denote by $P(s)$ the realization of P in the scenario s. Clearly, in any feasible schedule of our problem the maintenance activity can be seen as a single task with processing time $P(s)$. In particular, we assume a discrete finite set of scenarios $\mathcal{S} = \{s_{\min} = s_1, s_2, \ldots, s_k = s_{\max}\}$ ordered according to increasing durations of maintenance activity, i.e.,

$$P(s_i) < P(s_{i+1}) \qquad\qquad i = 1, \ldots, k-1. \tag{1}$$

We are interested in scheduling the n jobs together with the maintenance activity so that, taking into account a given objective function, a certain robustness criterion is minimized. Two objective functions are considered, namely makespan and total completion time of the jobs. A formal definition of the four criteria used in this work requires some additional notation presented hereafter.

In our problem, a solution is just a sequence π of jobs in J. However, in order to define the performance of a particular solution π, we look at the realization of the random variable P. More precisely, for all possible scenario $s \in \mathcal{S}$, we evaluate the objective (makespan or total completion time) in the so called *realization schedule* $\sigma(\pi, s)$ which is built as follows. For a certain duration of the maintenance activity, the realization schedule processes the jobs in the order specified by π while M is executed at the latest available time so that: (1) it does not violate the given deadline d and (2) it does not introduce any unnecessary idle time in the schedule after its release time. In the following a formal definition of a realization schedule is given.

Definition 1 *Given a scenario $s \in \mathcal{S}$, the realization $P(s)$ for the maintenance activity duration, and a sequence $\pi = \langle \pi_1, \pi_2, \ldots \pi_n \rangle$ of the jobs in J, let $k(s)$ be the index of the* critical *job in π such that*

$$\sum\nolimits_{i=1}^{k(s)-1} p_{\pi_i} \leq d - P(s) \qquad\qquad \sum\nolimits_{i=1}^{k(s)} p_{\pi_i} > d - P(s),$$

then the realization schedule $\sigma(\pi, s)$ *is obtained by scheduling the maintenance activity at the earliest possible time between jobs* $\pi_{k(s)-1}$ *and* $\pi_{k(s)}$. *Hence, the resulting sequence of jobs in* $\sigma(\pi, s)$ *is* $\langle \pi_1, \dots \pi_{k(s)-1}, M, \pi_{k(s)}, \dots \pi_n \rangle$.

See for instance the schedules depicted in Figure 1 The first and fourth Gantt diagrams represent two realizations schedules corresponding to a sequence $\pi = SPT$. Note that job 2 is processed before or after M depending on the realization value $P(s)$, however the job sequence remains unchanged. (Similar considerations hold for the two schedules illustrated in the same figure corresponding to sequence π^2.)

For a realization schedule $\sigma(\pi, s)$, the completion time $C_i(\pi, s)$ of the i-th job is

$$C_i = \begin{cases} \sum_{\ell=1}^{i} p_{\pi_\ell} & i < k(s) \\ \sum_{\ell=1}^{i} p_{\pi_\ell} + P(S) + \max\left\{0, r - \sum_{\ell=1}^{k(s)-1} p_{\pi_\ell}\right\} & k(s) \le i \le n \end{cases} \tag{2}$$

The term $\max\left\{0, r - \sum_{i=1}^{k(s)-1} p_{\pi_i}\right\}$ measures a possible idle time before the earliest possible start of the maintenance activity.

The values of the makespan $C_{\max}(\pi, s)$ and of the total completion time $\sum C_j(\pi, s)$ are then:

$$C_{\max}(\pi, s) = \max_{i \in J}\{C_i\} = C_{\pi_n} \tag{3}$$

$$\sum_{j \in J} C_j(\pi, s) = \sum_{j=1}^{n} (n - j + 1) p_{\pi_j} + \\ + (n - k(s) + 1) \left[P(s) + \max\{0, r - \sum_{j=1}^{k(s)-1} p_{\pi_j}\} \right]. \tag{4}$$

Observe that, since the realization of P is not known *a priori*, the index of the critical job and therefore the resulting realization schedule $\sigma(\pi, s)$ is not known in advance, too. The problem consists in finding the sequence π in which the jobs of J will be processed so that, whatever the duration of the maintenance activity would be, the resulting schedule is satisfactory in terms of either total completion time or makespan of jobs. In other words, π must be robust with respect to the variations of P.

Robustness criteria. We are now in the position to give a formal definition of the four robustness criteria that have been introduced in Section 1. Given a job sequence π, let us indicate by $f(\pi, s)$ the value, according to objective function f (i.e., $C_{\max}$ or $\sum C_j$), of a realization schedule $\sigma(\pi, s)$. Moreover, let $f^*(s)$ be the optimal solution value in scenario s. We have:

$$ABS(\pi) = \max_{s \in \mathcal{S}}\{f(\pi, s) - f^*(s)\} \quad (\textit{maximum absolute regret}); \tag{5}$$

$$REL(\pi) = \max_{s \in \mathcal{S}} \frac{f(\pi, s)}{f^*(s)} \quad (\textit{maximum relative regret}); \tag{6}$$

$$MM(\pi) = \max_{s \in \mathcal{S}} f(\pi, s) \quad (\textit{min-max}); \tag{7}$$

$$OWA(\pi) = \sum_{i=1}^{k} w_i f(\pi, s[i]) \quad (\textit{ordered weighted averaging}); \tag{8}$$

where, in (8), $s[i] \in \mathcal{S}$ is the scenario producing the i-th largest value of the objective function f, i.e., $f(\pi, s[i]) \geq f(\pi, s[i+1])$, $i = 1, \ldots, k-1$, and w_i is a given weight assigned to scenario $s[i]$.

For the developments to follow, it is also useful to highlight a general property that holds for any fixed solution sequence (independently on the objective function and the objective criterion.)

Observation 2 *For any given solution sequence $\bar{\pi}$ and scenarios $s, s' \in \mathcal{S}$, if $P(s) < P(s')$, then $C_j(\sigma(\bar{\pi}, s)) \leq C_j(\sigma(\bar{\pi}, s'))$, for any job $j \in J$.*

The above observation can be easily shown by considering that, in the scenario s, it is always possible to schedule the jobs in the same positions as in $\sigma(\bar{\pi}, s')$ by introducing a suitable idle time having duration $P(s') - P(s)$. Hence, the jobs completion times cannot be worse in $\sigma(\bar{\pi}, s)$ than in $\sigma(\bar{\pi}, s')$. This in turn implies that $f(\bar{\pi}, s) \leq f(\bar{\pi}, s')$ for any regular objective function f (as, in particular, $C_{\max}$ and $\sum C_j$).

A straightforward consequence of the above arguments is that $s[i] = s_{k-i+1}$ for $i = 1, \ldots, k$ and therefore Equation (8) can be rewritten as follows:

$$OWA(\pi) = \sum_{i=1}^{k} w_i f(\pi, s[i]) = \sum_{s \in \mathcal{S}} w_s f(\pi, s). \tag{9}$$

Given an objective function f depending on the job completion times, a job sequence, i.e., a solution π, and a certain criterion $c \in \{ABS, REL, MM, OWA\}$, among those in Equations (7)–(9), we indicate by $c_f(\pi)$ (or simply, by $c(\pi)$) the value of the selected criterion c corresponding to solution π. As a consequence, we may formally define the addressed robust scheduling problem as follows.

> Robust single-machine Scheduling with Maintenance activity Problem ($RSMP(f, c)$):
> *Given* a set J of n jobs with deterministic processing times p_j, $j = 1, \ldots, n$, and a set $\mathcal{S}$ of discrete scenarios, corresponding to $|\mathcal{S}|$ possible values of the duration of a maintenance activity M;
> *find* a sequence π of the jobs *such that* $c_f(\pi)$ is minimized.

We denote by $\pi^{ROB}(f, c)$ an optimal solution of the above problem $RSMP(f, c)$ with value $c^* = c(\pi^{ROB})$. Whenever it is clear from the context which objective function f and robustness criterion c we are considering, we use π^{ROB} to denote such solution.

Approximation. Also in the context of robust optimization, it is reasonable to investigate the performance of an algorithm that does not guarantee to find the best robust solution compared to optimality. In order to properly define the approximation ratio of an algorithm, we restrict ourselves to consider the set $\mathcal{I}^{>0}$ of instances I of $RSMP(f, c)$ such that the the value $c^*(I)$ of an optimal solution of I is strictly positive. Then the approximation ratio $\varepsilon(A)$ of algorithm A is given by

$$\varepsilon(A) = \sup_{I \in \mathcal{I}^{>0}} \frac{c(\pi, I)}{c^*(I)} \tag{10}$$

in which $c(\pi, I)$ is the value of the sequence determined by algorithm A on instance I. Clearly, $\varepsilon(A) > 1$.

Summary of results. In this work, for each of the two objective functions and each of the four robustness measures, we derive several properties.

Scenario Optimality The first one addresses an issue present in several papers in robust optimization, namely whether the robust solution is also an optimal solution of a given scenario. In this case we say that our problem has the *scenario optimality* property. For makespan minimization, we show that the answer is yes for all four robustness criteria (see Theorems 7, 9), while when the objective is total completion time, this is only true if $c = MM$, i.e., we are looking at a min-max robustness criterion (see Theorems 11, Observation 14, and, also the discussions in Sections 4.2 and 4.4).

Extremality Another typical assumption in several robust optimization problems is that uncertain data take values in a given interval. This is the case, for instance, in [6] where a single machine scheduling problem is addressed with aleatory job processing times that may assume real values in given intervals (no maintenance activity has to be performed). The authors consider the total completion time objective ($f = \sum C_j$) and maximum absolute regret criterion for robustness ($c = ABS$). They show that, for any solution sequence, such a maximum regret occurs in special "extreme point" scenarios. More explicitly, the maximum *value* of the deviation from the optimum, for any solution, takes place when all processing times values are equal either to their minimum or their maximum feasible values.

By slightly extending this concept, in this paper we address the question whether the worst-case value of the quantity measuring the robustness occurs in correspondence of an extreme point scenario (in our case, $s_1 = s_{\min}$ and $s_k = s_{\max}$ where the duration of the maintenance M is at its minimum and maximum values, respectively). If it is the case, we refer to such a situation by saying that the problem has the *extremality* property. Clearly, extremality makes sense for $c = ABS, REL, MM$, while it does not for the OWA robustness criterion, which is a linear combination of values taken by the objective in the various scenarios.

We show that for the absolute and relative regret the extremality property does not hold in general, independently on the makespan or total completion time objective (see Theorems 10 and 12). On the other hand, for the min-max criterion, the maximum value for any objective occurs when $s = s_{\max}$ and hence the property is always verified (see Observations 5 and 14).

Approximation The third issue we address regards approximation. In particular in Section 5, we discuss both the existence of approximation algorithms and whether a given sequence (i.e., a solution to $RSMP(f, c)$) can provide good quality solutions for our problems. We do not address this issue for the case of the maximum absolute regret criterion ($c = ABS$). In this case, inapproximability directly follows from the $\mathcal{NP}$-hardness of the corresponding deterministic problem, as observed in [12].

$C_{\max}$	*ABS*	*REL*	*MM*	*OWA*
Scenario Opt.	yes (Thm. 9)	yes (Thm. 9)	yes (Thm. 7)	yes (Thm. 9)
Extremality	no (Thm. 10)	no (Thm. 10)	yes (Obs. 8)	—
Approximation	not approx.	2 (Cor. 17)	FPTAS (Cor. 19)	2 (Cor. 17)

Table 1: Summary of results for $C_{\max}$ objective.

$\sum C_j$	*ABS*	*REL*	*MM*	*OWA*
Scenario Opt.	no (Thm. 11)	no (Obs. 13)	yes (Obs. 14)	no (Obs. 15)
Extremality	no (Thm. 12)	no (Obs. 13)	yes (Obs. 14)	—
Approximation	not approx.	9/7 (Cor. 20)	9/7 (Cor. 20)	9/7 (Cor. 20)

Table 2: Summary of results for $\sum C_j$ objective.

Tables 1 and 2 summarize the obtained results.

Complexity and general results. The deterministic problem of scheduling a flexible maintenance activity with the objective of minimizing the makespan is trivially $\mathcal{NP}$-hard and this can be easily shown by a reduction from PARTITION [8]. Furthermore, as shown in [1], the deterministic problem of scheduling a flexible maintenance activity with the objective of minimizing total completion time of the jobs is binary $\mathcal{NP}$-hard.

Since the deterministic problem coincides with $RSMP(f, c)$ with a single scenario, $RSMP(f, c)$ is $\mathcal{NP}$-hard, too, for any robustness criterion c given in (5)–(9) and any objective function f (i.e., $C_{\max}$ or $\sum C_j$).

Yet, there are a number of polynomially solvable cases. In particular, it is quite easy to show that the problem of scheduling a flexible maintenance activity within a given due date with the objective of minimizing jobs total completion time can be solved to optimality by a SPT-sequence of the jobs (this is also mentioned by [27]). A straightforward consequence of this fact is the following result.

Observation 3 *If the release date of the maintenance activity is $r = 0$ then, for any robustness criterion $c \in \{ABS, REL, MM, OWA\}$ and objective function $f = \sum C_j$ or $f = C_{\max}$, $RSMP(f, c)$ is polynomially solvable and $\pi^{ROB}(f, c) = SPT$.*

3 Makespan minimization

In the following, we study the robustness properties of $RSMP(C_{\max}, c)$.

We first show that, for any robust criterion c, the robust solution of $RSMP(C_{\max}, c)$ belongs to the set of extreme point scenarios. With this aim, in the following we present some useful observations that allow to characterize the optimal solutions in each scenario.

Recalling that the duration of M increases with the scenario indices (see Equation (1)), it is straightforward to see that the following observation holds for any job sequence π.

Observation 4 *Given a job-sequence π, if the realization schedule $\sigma(\pi, s_j)$ contains no idle time then:* (1) *neither do the realization schedules $\sigma(\pi, s_i)$ in all scenarios s_i, with $i < j$;* (2) *π is an optimal sequence (i.e., minimizes the makespan) in scenarios $s_j, s_{j-1}, \ldots, s_1$.*

An obvious consequence of Observation 4 is that, if the realization schedule $\sigma(\pi, s_{\max})$ does not contain an idle time (in scenario $s_{\max}$), then the realization schedules $\sigma(\pi, s)$ are optimal and do not contain idle times in any scenario $s \in \mathcal{S}$.

Observation 5 *Suppose, in scenario s_i, a realization schedule $\sigma(\pi, s_i)$ contains idle time. Recalling the definition of critical job $k(s)$ for $s \in \mathcal{S}$ (see Definition 1), one has:*

1. *$\sum_{h=1}^{k(s_i)-1} p_{\pi_h} < r$, since otherwise, in $\sigma(\pi, s_i)$, M might start immediately after the jobs $\pi_1, \ldots \pi_{k(s_i)-1}$, without idle time;*
2. *the maintenance activity starts at time r in $\sigma(\pi, s_i)$;*
3. *all realization schedules $\sigma(\pi, s_j)$ for any scenario s_j with $j > i$ contain an idle time of the same duration $r - \sum_{h=1}^{k(s_i)-1} p_{\pi_h}$.*

A consequence of the above observation is Observation 6.

Observation 6 *For any job-sequence π, an index i exists, with $0 \leq i \leq |\mathcal{S}|$, such that $\sigma(\pi, s_j)$ has no idle*[2] *for any scenario s_j with $j \leq i$ and has idle $r - \sum_{h=1}^{k(s_{\max})-1} p_{\pi_h}$ for any scenario s_j with $j > i$.*

Let us consider now $RSMP(C_{\max}, c)$ with the min-max criterion (7), i.e., $c = MM$. When looking at an optimal sequence π_s^* for a given scenario s, we note that any solution minimizing the machine idle minimizes the schedule makespan as well. Let $J^*(v)$ indicate a subset of jobs of maximum total processing time such that $\sum_{j \in J^*(v)} p_j \leq v$ (i.e., $\sum_{j \in J^*(v)} p_i \geq \sum_{j \in J'} p_j$ for all $J' \subseteq J$ such that $\sum_{j \in J'} p_j \leq v$). Let us consider now the set $J^*(d - P(s_{\max}))$, i.e., a subset of jobs of maximum total processing time that can be scheduled before M in scenario $s_{\max}$. When $s = s_{\max}$, let π be a sequence that sequences first a set of jobs of total processing time $\sum_{i \in J^*(d-P(s_{\max}))} p_i$ and let $\tau = r - \sum_{i \in J^*(d-P(s_{\max}))} p_i$, that is the idle time of the realization schedule $\sigma(\pi, s_{\max})$. Since $\sigma(\pi, s_{\max})$ minimizes the machine idle in scenario $s_{\max}$, then it is optimal in this scenario (i.e., minimizes the makespan).

In the following theorem, we show that for $RSMP(C_{\max}, MM)$ there exists a robust solution minimizing the machine idle in scenario $s_{\max}$.

Theorem 7 *A robust solution π exists for $RSMP(C_{\max}, MM)$, that sequences first a set of jobs of total processing time $\sum_{i \in J^*(d-P(s_{\max}))} p_i$, i.e., that is optimal in scenario $s_{\max}$.*

Proof. We first observe that if $\sum_{i \in J^*(d-P(s_{\max}))} p_i \geq r$ the thesis trivially holds. In this case, by Observation 4, in any scenario s, π is optimal since the realization schedule does not contain idle times and $C_{\max}(\pi, s) = \sum_{j \in J} p_j + P(s)$.

Suppose then $\tau = r - \sum_{i \in J^*(d-P(s_{\max}))} p_i = r - \sum_{i \in J^*(r)} p_i > 0$. Then, by definition of $J^*(d - P(s_{\max}))$, an optimal sequence in scenario $s_{\max}$ has an idle time of τ, and by definition, the robust solution π is optimal in this scenario.

[2]If $i = 0$ then idle time exists in all scenarios.

Note that, by Observation 6, π has an idle smaller than or equal to τ in any scenario, and, consequently, the scenario with the maximum makespan is always $s_{\max}$. Since π is optimal in scenario $s_{\max}$ (as it minimizes the idle time), we have that, π minimizes the robustness criterion (7) for the makespan objective. □

Observation 8 *A straightforward consequence of Theorem 7, that the extremality property holds for $RSMP(C_{\max}, MM)$, i.e., the scenario maximizing the absolute deviation from the optimum belongs to the set of extreme point scenarios (in our problem, $s_{\min}$ or $s_{\max}$). In this case, for any solution sequence π, the min-max robustness measure always takes its value from the makespan of the realization schedule in the extreme scenario $s_{\max}$.*

Let us consider now the absolute regret (5), the relative regret (6) and the OWA (9) criteria. In these cases, a solution minimizing the machine idle in scenario $s_{\max}$ could not be a robust solution. On the other hand, the next Theorem 9 shows that a robust solution of $RSMP(C_{\max}, c)$, with $c \in \{ABS, REL, OWA\}$, either minimizes the machine idle in scenario $s_{\max}$, i.e., is optimal in scenario $s_{\max}$, or it is optimal in scenario $s_{\min}$.

Theorem 9 *There is a robust solution π of $RSMP(C_{\max}, c)$ that is optimal in scenario $s_{\max}$ or in scenario $s_{\min}$, for any given criterion $c \in \{ABS, REL, OWA\}$.*

Proof. Given any robust criterion c in (5)–(9), let us suppose, by contradiction, that the robust solution π^{ROB} of problem $RSMP(C_{\max}, c)$ is optimal neither in scenario $s_{\max}$ nor in scenario $s_{\min}$.

As a consequence, π^{ROB} yields a strictly positive idle time $\tau > 0$ in scenario $s_{\min}$ (or, in other words, the realization schedule $\sigma(\pi', s_{\min})$ causes a machine idle before the maintenance activity), otherwise π^{ROB} would be optimal in that scenario. By Observation 6, it follows that π^{ROB} is such that $\sigma(\pi^{ROB}, s)$ has a machine idle equal to τ for any scenario $s \in \mathcal{S}$.

Let $0 \leq \tau^* \leq \tau$ be the minimum machine idle time in scenario $s_{\max}$, that is, the machine idle time in the realization schedule of an optimal sequence $\pi^*_{s_{\max}}$ in scenario $s_{\max}$. Again, by Observation 6, $\pi^*_{s_{\max}}$ has an idle not larger than $\tau^* \leq \tau$ in all the scenarios in $\mathcal{S}$. Hence, the makespan of $\pi^*_{s_{\max}}$ is not worse than those of π^{ROB} in all scenarios, i.e., $C_{\max}(\sigma(\pi^*, s)) \leq C_{\max}(\sigma(\pi^{ROB}, s))$ for all s in $\mathcal{S}$, and strictly better in at least one (otherwise π^{ROB} would be optimal in $s_{\max}$), which contradicts that π^{ROB} is a robust solution. □

In conclusion, Theorems 7 and 9 imply the scenario-optimality property when the objective funtion is the makespan minimization under any robustness criterion (5)–(9).

Let us consider now the extremality property. Recall that such a property holds when the worst-case *value* of the quantity measuring the robustness occurs in correspondence of an extreme scenario, that is $s_{\min}$ or $s_{\max}$. As already stated, extremality makes sense only for the ABS, REL and MM robustness criteria. Note that Theorem 9 does not imply anything about the extremality property for $RSMP(C_{\max}, ABS)$ and $RSMP(C_{\max}, REL)$. In fact, the following theorem shows that the extremality property does not hold when the robustness criterion is ABS or REL.

Theorem 10 *For $RSMP(C_{\max}, ABS)$ and $RSMP(C_{\max}, REL)$, the extremality property does not hold.*

Proof. To prove the thesis, we are showing that there is an instance having a solution[3] in which the maximum value of the (absolute or relative) regret occurs in a scenario s such that the maintenance activity has not its maximum or minimum duration (i.e., $s \notin \{s_{\min}, s_{\max}\}$).

Consider the following instance of $RSMP(C_{\max}, c)$ with $n = 3$ jobs having processing times $p_1 = 7$, $p_2 = 8$ and $p_3 = 11$. Assume there are 3 scenarios with the following maintenance activity durations $P(s_1) = 1$, $P(s_2) = 7$, and $P(s_3) = 10$. Moreover, let $r = 10$ and $d = 20$. The optimal makespan values in the three scenarios s_1, s_2, and s_3 are 27, 33, and 38, respectively. There are two robust solutions for criteria ABS and REL, namely, $\pi^{ROB} \in \{\langle 2, 3, 1\rangle, \langle 2, 1, 3\rangle\}$, which are also optimal in scenarios s_1 and s_3 (with no idle times). The maximum absolute regret value is therefore $ABS = 2$ obtained in scenario s_2. In fact, in this scenario, the makespan of π^{ROB} is $C_{\max}(\sigma(\pi^{ROB}, s_2)) = 35$, while an optimal solution $\pi^*_{s_2}$ schedules job 3 in the first position, in order to generate no idle time, and produce a makespan of 33.

Also for REL criterion, π^{ROB} has its maximum relative regret value in scenario s_2, which is equal to $\frac{35}{33}$. □

Observe that, for the OWA criterion, the distribution of weights may be such that the optimal solution in a single scenario, say s', prevails over any other choice; while the optimal sequence for scenario s' does not minimize the idle times in other scenarios. Hence, the extremality property does not hold for OWA, too.

4 Total Completion time minimization

In this section, we study the robustness properties of $RSMP(\sum C_j, c)$.

4.1 Absolute Regret

In order to characterize a robust solution of $RSMP(\sum C_j, ABS)$, we observe that some natural properties that hold for several robust optimization problems similar to ours do not apply to the present setting. In this regard, we present two negative results in the theorems hereafter.

When there is a limited number of scenarios, it comes natural to use as a robust solution a sequence which is optimal in at least one scenario. This could be pursued by iteratively applying the pseudopolynomial dynamic program proposed in [27]. Unfortunately, this approach, in general, does not yield a solution minimizing the maximum regret as shown below.

Theorem 11 *A robust solution of $RSMP(\sum C_j, ABS)$ may not correspond to the optimal solution of any of the scenarios in $\mathcal{S}$.*

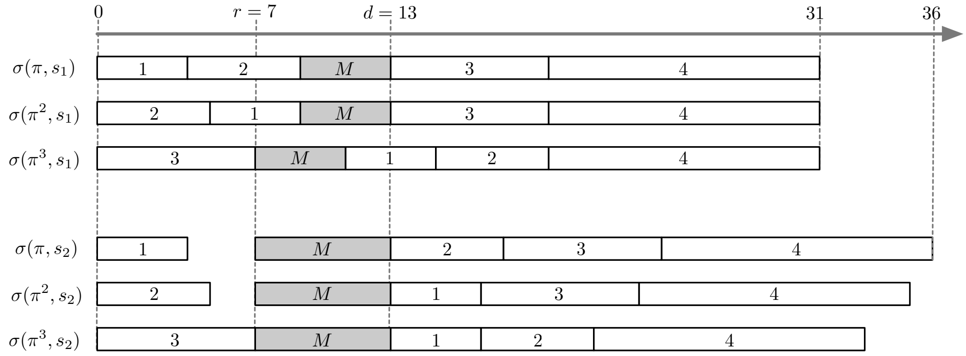


Figure 1: The realization schedules in the proof of Theorem 11.

Proof. The following example proves the thesis. Consider an instance with four jobs $J = \{1, 2, 3, 4\}$ with processing times $p_1 = 4$, $p_2 = 5$, $p_3 = 7$, and $p_4 = Q$ with $Q > 10$. Let the release time of the maintenance activity be $r = 7$ and its deadline $d = 13$. There are two scenarios and the two possible values for $P(s)$ are: $P(s_1) = 4$ and $P(s_2) = 6$.

First note that job 4 cannot be processed before the maintenance activity in both scenarios, so, since it is also the longest job, it is always beneficial to schedule it as the last job. By simple enumeration we may observe that there are only three nondominated sequences which are: $\pi^1 =< 1, 2, 3, 4 >$, $\pi^2 =< 2, 1, 3, 4 >$, and $\pi^3 =< 3, 1, 2, 4 >$.
In scenario s_1 in which $P(s_1) = 4$, the realization schedules for the above three sequences (see Figure 1) have value: $\sum C_j(\pi^1, s_1) = 4 + 9 + 20 + 20 + Q = 53 + Q$, $\sum C_j(\pi^2, s_1) = 5 + 9 + 20 + 20 + Q = 54 + Q$, and $\sum C_j(\pi^3, s_1) = 7 + 15 + 20 + 20 + Q = 62 + Q$. Hence in this scenario, the optimal sequence is π^1 and the absolute regret of π^2 equals 1, while the regret of π^3 is 9.
Consider now the second scenario with $P(s_2) = 6$. Any feasible schedule permits at most one job before the maintenance activity. The realization schedules are depicted in Figure 1 and their total completion time values are: $\sum C_j(\pi^1, s_2) = 4+18+25+25+Q = 72+Q$, $\sum C_j(\pi^2, s_2) = 5 + 17 + 24 + 24 + Q = 70 + Q$, and $\sum C_j(\pi^3, s_2) = 7 + 17 + 22 + 22 + Q = 68 + Q$. So in this case, the optimal sequence is π^3 and the absolute regret of π^1 is 4, while the absolute regret of π^2 is equal to 2.

Summarizing, the maximum regret of π^1 in the two scenarios is 4, that of π^2 is 2, and π^3 is 9. So, in this case, $c^* = 2$ and $\pi^{ROB}(\sum C_j, ABS) = \pi_2$, which is not optimal in any of the two scenarios. ☐

From Section 2, we recall that the extremality property holds for a certain robust problem, if the scenario maximizing the absolute deviation from the optimum belongs to the set of extreme point scenarios, i.e., those with minimum or maximum processing times values, in our case $s_{\min}$ and $s_{\max}$. It is possible to show that for $RSMP(\sum C_j, ABS)$ this property does not hold.

[3] In our example, we are considering solutions which are—incidentally—robust, although this is not required to prove the result.

Theorem 12 *There exist an instance I of $RSMP(\sum C_j, ABS)$ and a sequence π of the jobs such that the maximum absolute regret does not correspond to an extreme value of the maintenance activity duration.*

Proof. Consider an instance of $RSMP(\sum C_j, ABS)$ with 5 jobs having processing times $p_1 = 1$, $p_2 = 3$, $p_3 = 4$, $p_4 = 10$, $p_5 = 12$, and let $r = 0$, $d = 17$ and $P \in [8, 10] \cap \mathbb{Z}$. Since $r = 0$ the optimal solution for any scenario can be obtained by sequencing the jobs in SPT order, that is in the given order. In interval $[8, 10]$ there are three (integer) scenarios: s_1 and s_3 correspond to the extreme point scenario, namely $P(s_1) = 8$ and $P(s_3) = 10$, while s_2 is an intermediate scenario with $P(s_2) = 9$.

Consider now the sequence $\pi = \langle 1, 2, 4, 3, 5 \rangle$. The total completion times for π in the three scenarios are $\sum C_j(\sigma(\pi, s_1)) = 91$, $\sum C_j(\sigma(\pi, s_2)) = 94$, and $\sum C_j(\sigma(\pi, s_3)) = 97$, while the optimal solutions π_1^*, π_2^* and π_3^* in the three scenarios have values $\sum C_j(\sigma(\pi_1^*, s_1)) = 77$, $\sum C_j(\sigma(\pi_2^*, s_2)) = 79$, and $\sum C_j(\sigma(\pi_3^*, s_3)) = 91$.

It is easy to see that the maximum (absolute) regret for π corresponds to the intermediate scenario s_2 thus proving the thesis. $\square$

4.2 Relative Regret

In this section we are looking at $RSMP(\sum C_j, REL)$, the problem in which we are interested in minimizing the maximum relative regret as defined by Equation (6) with total completion objective.

By using the same instances in the proofs of Theorem 11 and Theorem 12, we may easily extend those negative results to the case of the maximum relative regret criterion, i.e, for $RSMP(\sum C_j, REL)$ there is no correspondence between robust and scenario-optimal solutions and the extremality property does not hold.

In fact, as for Theorem 11, considering the relative deviations from the optima, we have that (for any positive M) the maximum relative regrets of π_1, π_2, and π_3 are $REL(\pi_1) = \frac{72+M}{68+M}$, $REL(\pi_2) = \frac{70+M}{68+M}$, and $REL(\pi_3) = \frac{62+M}{53+M}$, respectively. As a consequence, since $REL(\pi_2) = \min\{REL(\pi_1), REL(\pi_2), REL(\pi_3)\}$, $\pi^{ROB} = \pi_2$ is the min-max relative regret solution.

Analogously, in the proof of Theorem 12, it is not hard to to verify that both the maximum values of absolute and relative regret for π correspond to the intermediate scenario s_2 thus proving the thesis. In conclusion, we have the following:

Observation 13 *Theorems 11 and 12 extend to problem $RSMP(\sum C_j, REL)$.*

4.3 Min-max criterion

Consider now criterion (7) and problem $RSMP(\sum C_j, MM)$. By Observation 2, we have that the maximum value of the objective function is attained when the flexible maintenance activity M has a duration equal to $P(s_{\max})$.

Hence, the following observation is straightforward.

Observation 14 *For $RSMP(\sum C_j, MM)$:*

1. *For any fixed job sequence π, if $i < h$ (and hence $s_i < s_h$) then $\sum_{j\in J} C_j(\sigma(\pi, s_i)) \leq \sum_{j\in J} C_j(\sigma(\pi, s_h))$, i.e., the maximum value of the total completion time is obtained in the extreme scenario $s_{\max}$.*

2. *Hence, a robust solution π^{ROB} corresponds to an optimum in scenario $s_{\max}$, i.e., $c^* = \sum_{j\in J} C_j(\sigma(\pi^{ROB}, s_{\max}))$.*

In other words, the extremality property holds for $RSMP(\sum C_j, MM)$ (and this can be done in pseudopolynomial time as described in [27]).

4.4 Ordered Weighted Average

Let us consider now $RSMP(\sum C_j, OWA)$. First, recall that, with this robustness criterion, there is no point in considering the extremality property.

Furthermore, by using the same instance of the proof of Theorem 11 and by weighting the two scenarios with the same weight, i.e. $w_1 = w_2 = 1/2$ it is quite easy to show that the robust solution may not correspond to the optimal solution of one of the scenarios. Summarizing, we have the following:

Observation 15 *Theorems 11 and 12 extend to problem $RSMP(\sum C_j, OWA)$.*

5 Approximation Properties

In this section, approximation results for $RSMP(f, c)$ are reported. Note that, here, as observed in Section 2, we do not consider approximation properties for $RSMP(f, ABS)$, since—as observed in [12]—when the deterministic (single-scenario) problem is $\mathcal{NP}$-hard the corresponding robust version with $c = ABS$ is not at all approximable unless $\mathcal{P} = \mathcal{NP}$.

We first prove a general approximation result holding for $RSMP(f, c)$ for any regular objective function f. It states that if we are able to find a fixed sequence $\bar{\pi}$ which is a ε-approximate solution for the (deterministic) single-scenario version of the problem, in all scenarios $s \in \mathcal{S}$, then $\bar{\pi}$ is also a ε-approximate solution of the robust problem $RSMP(f, c)$. This is the case when the algorithm provides a sequence based on a priority rule independent on the duration of the maintenance activity, such as, for instance, the SPT rule.

Theorem 16 *Let f be a regular objective function and $c \in \{REL, MM, OWA\}$ a robustness criterion. If algorithm $\mathcal{A}$ is an ε-approximation algorithm for the (deterministic) single-scenario version of $RSMP(f, c)$ that returns the same sequence $\bar{\pi}$ for all scenarios $s \in \mathcal{S}$, then $\bar{\pi}$ is also an ε-approximate solution of the robust problem $RSMP(f, c)$.*

Proof. Let us consider the robust problem $RSMP(f, c)$, and let $\bar{c}$ denote the value of the solution returned by $\bar{\pi}$ in the worst scenario, while π^{ROB} and c^* are, respectively, the robust solution and its value according to the considered robustness criterion c. Hereafter, we show that, for each $c \in \{REL, MM, OWA\}$, $\bar{c} \leq \varepsilon c^*$ and hence the thesis holds.

Case (1): $c = MM$. Let us first consider the min-max robustness criterion. Then,

$$\pi^{ROB} = \arg\min_{\pi} \max_{s\in\mathcal{S}} \{f(\pi, s)\}$$

and, since in the case of minimax criterion, c^* is given by the measure of the objective in the worst possible scenario we have: $c^* = \max_{s\in\mathcal{S}} \left\{f(\pi^{ROB}, s)\right\}$. Hence

$$\bar{c} = \max_{s\in\mathcal{S}} \{f(\bar{\pi}, s)\} = f(\bar{\pi}, s') \leq \varepsilon f^*_{s'} \leq \varepsilon f(\pi^{ROB}, s') \leq \varepsilon \max_{s\in\mathcal{S}} \left\{f(\pi^{ROB}, s)\right\} = \varepsilon c^*.$$

Case (2): $c = REL$. For the relative regret robustness criterion we have

$$\pi^{ROB} = \arg\min_{\pi} \max_{s\in\mathcal{S}} \left\{\frac{f(\pi, s)}{f^*_s}\right\} \text{ and } c^* = \max_{s\in\mathcal{S}} \left\{\frac{f(\pi^{ROB}, s)}{f^*_s}\right\}.$$

Clearly $c^* \geq 1$ and therefore, if $\hat{s} \in \mathcal{S}$ is a scenario corresponding to the worst case objective ratio, the following inequalities hold:

$$\bar{z} = \max_{s\in\mathcal{S}} \left\{\frac{f(\bar{\pi}, s)}{f^*_s}\right\} = \frac{f(\bar{\pi}, \hat{s})}{f^*_{\hat{s}}} \leq \varepsilon \leq \varepsilon c^*.$$

Case (3): $c = OWA$. In this case, we have

$$\pi^{ROB} = \arg\min_{\pi} \left\{\sum_{s\in\mathcal{S}} \lambda_s f(\pi, s)\right\} \text{ and } c^* = \sum_{s\in\mathcal{S}} \lambda_s f(\pi^{ROB}, s).$$

Therefore

$$\bar{z} = \sum_{s\in\mathcal{S}} \lambda_s f(\bar{\pi}, s) \leq \sum_{s\in\mathcal{S}} \lambda_s \varepsilon f^*_s \leq \varepsilon \sum_{s\in\mathcal{S}} \lambda_s f(\pi^{ROB}, s) = \varepsilon c^*.$$

□

5.1 Makespan Minimization

Recall that, in $RSMP(C_{\max}, c)$, whatever the solution sequence π, the maximum among the makespan values of the realization schedules corresponds to $C_{\max}(\sigma(\pi, s_{\max}))$, i.e., the solution value for the scenario $s_{\max}$, when the maintenance activity has the largest duration. As a consequence, under the min-max robustness criterion, any sequence $\tilde{\pi}$ within a constant ratio $\alpha > 1$ from an optimal solution for scenario $s_{\max}$ is also an α-approximating solution of $RSMP(C_{\max}, MM)$, i.e.,

$$MM(\tilde{\pi}) = C_{\max}(\sigma(\tilde{\pi}, s_{\max})) \leq \alpha\, C_{\max}(\sigma(\pi^*_{s_{\max}}, s_{\max})) = \alpha c^* \tag{11}$$

It is easy to observe that an arbitrary sequence always produces a makespan not larger than twice the optimal solution. As a consequence, a corollary of Theorem 16 is that:

Corollary 17 *For $c \in \{REL, MM, OWA\}$, $RSMP(C_{\max}, c)$ admits a 2-approximation algorithm.*

For $RSMP(C_{\max}, MM)$, we are able to obtain a better approximation ratio by using any ϵ-approximate algorithm[4] A for SUBSET SUM [9]. Indeed, in the proof of the following theorem, we show that, by exploiting an ϵ-approximate algorithm A for the Subset Sum problem it is possible to obtain an ε approximation for our problem, where $\varepsilon \leq \min\{1+\theta, 2-\epsilon\}$ and $\theta \leq 1$, defined below, is roughly the ratio between the total processing time of the jobs processed before the maintenance activity and the maximum makespan in an optimal solution.

Theorem 18 *For problem $RSMP(C_{\max}, MM)$ there is a ε-approximate algorithm, where $\varepsilon = 1 + (1-\epsilon)\theta\}$, $\theta = (d - P(s_{\max})/(P(s_{\max}) + \sum_{j\in J} p_j)$, and ϵ is the best (largest) approximation ratio for* SUBSET SUM.

Proof. In its optimization format, a generic instance SUBSET SUM is defined as follows: Given a set N of n positive integers $a_1, a_2, \ldots, a_n$ and a bound $B > 0$, define a subset $N^* \subseteq N$ whose sum is as large as possible but not greater than B. Suppose there is an algorithm A that, for any instance of SUBSET SUM, always returns a subset $N^A \subseteq N$ such that

$$\sum_{j\in N^A} a_j \geq \epsilon \sum_{j\in N^*} a_j \qquad (12)$$

for some fixed $\epsilon \leq 1$.

Consider now an optimal solution π^* for the problem of minimizing $C_{\max}$ in the scenario $s_{\max}$. Denote by $C^*_{\max}$ the value of such a minimum makespan. The associated optimal realization schedule $\sigma(\pi^*, s_{\max})$, schedules a set $N^* \subseteq J$ of jobs *before* M and N^* corresponds to an optimal solution set of the above mentioned SUBSET SUM problem, with $a_j = p_j$ and $B = d - P(s_{\max})$. Let $p(N^*)$ be the total processing time of the jobs in N^*. Then $C^*_{\max} = \max\{r, p(N^*)\} + P(s_{\max}) + (p(J) - p(N^*))$, where $p(J) = \sum_{j\in J} p_j$ is the total processing time of all jobs.

Consider now another solution π^A obtained by sequencing first the jobs in a set $N^A \subseteq J$ returned by Algorithm A, where again $B = d - P(s_{\max})$. Let $p(N^A)$ be the total processing time of the jobs in N^A and let $C^A_{\max}$ be the makespan $C_{\max}(\sigma(\pi^A, s_{\max}))$ of the realization schedule obtained by π^A. Clearly, $p(N^A) \geq \epsilon\, p(N^*)$ and $C^A_{\max} = \max\{r, p(N^A)\} + P(s_{\max}) + (p(J) - p(N^A))$.

Assume first $p(N^*) < r$, as $p(N^A) \leq p(N^*)$ then

$$C^A_{\max} - C^*_{\max} = p(N^*) - p(N^A) \leq (1-\epsilon)p(N^*). \qquad (13)$$

As $p(N^*) \leq C^*_{\max}$, there exists $\theta \leq 1$ such that $p(N^*) \leq \theta\, C^*_{\max}$. Letting $\varepsilon = 1 + \theta - \theta\epsilon$, we have

$$C^A_{\max} \leq (1-\epsilon)p(N^*) + C^*_{\max} \leq \varepsilon\, C^*_{\max}. \qquad (14)$$

[4]Different from Equation (10), since we are dealing with a maximization problem, here $\epsilon < 1$ indicates a lower bound on the ratio between the value of the solution obtained by an approximation algorithm and the optimum.

If $p(N^A) \leq r < p(N^*)(\leq r')$, we have that Equation (13) becomes $C^A_{\max} - C^*_{\max} = r - p(N^A) \leq p(N^*) - p(N^A)$ and Equation (14) still holds. The remaining case $p(N^A) > r$ is trivial: the sequence $p(N^A)$ is in fact an optimal one.

In conclusion, due to (11), there is an ε-approximation algorithm for $RSMP(C_{\max}, MM)$. □

Note that $\varepsilon \leq \min\{1+\theta, 2-\epsilon\}$ so, if θ is small (for instance when $r << p(J)+P(s_{\max})$) or if ϵ is close to 1 (e.g., A is good approximation algorithm for Subset Sum), then there is a good approximation ratio for the robust problem.

So, due to the above theorem and the fact that Subset Sum problem admits a Fully Polynomial Time Approximation Scheme (FPTAS) [9], the following corollary is immediate.

Corollary 19 *Problem $RSMP(C_{\max}, MM)$ admits a FPTAS.*

Note that Corollary 19 is also implied by the results provided in the paper by He *et al.* [10] (dealing with the deterministic version of the problem). However we point out that the computational complexity $O(kn^2/\epsilon)$ derived by using their approach is larger than the one we can obtain running the fastest FPTAS for Subset Sum as illustrated in the proof of Theorem 18, with final complexity smaller than $O(n/\epsilon)$ [13].

5.2 Total Completion Time Minimization

When $r > 0$, as shown in [27], SPT is a 9/7-approximation algorithm for the deterministic problem of scheduling a maintenance activity of a fixed duration, with the objective of minimizing the total completion time. Clearly, in the robust version of the same problem, SPT produces a sequence not larger than 9/7 of the scenario optimum, for all the scenarios. Hence, by Theorem 16, the following corollary holds:

Corollary 20 *SPT is a 9/7-approximation algorithm for $RSMP(\sum C_j, c)$ when $c \in \{REL, MM, OWA\}$.*

6 Solution algorithms for $RSMP(\sum C_j, \boldsymbol{ABS})$

In this section, we are dealing with computing a robust solution for $RSMP(\sum C_j, ABS)$. To this purpose we designed exact and heurisic algorithms, whose performance are assessed through a computational study. We focus on this special problem since, as discussed in the preceding sections, it is apparently the hardest one among those illustrated above: Indeed, when considering other robustness criteria, namely, relative regret, min-max, and OWA, there are simple algorithms guaranteeing *a-priori* small errors (and achieving outstanding results in the average case).

Hereafter, we first present a Mixed Integer Linear Programming formulation of the problem and simple heuristic algorithms. Then, in order to compare the exact and the heuristic approaches a computational campaign is performed.

6.1 A Mixed Integer Linear Programming formulation

A Mixed Integer Linear Programming formulations of the problem can be written by *explicitly* considering all possible scenarios in the finite set $\mathcal{S}$, i.e., all possible realizations $P(s)$ and the values $\sum C_j^*(s)$ of the corresponding optimal solutions (i.e., minimum total completion time of the jobs in the scenario s).

Hereafter, we present a MILP model for $RSMP(\sum C_j, ABS)$ in which the $\sum C_j^*(s)$ are considered as input data. A second MILP, that can be found in [7], is here omitted since it is outperformed by the one described below.

Binary variable x_{jh}, defined for all jobs $j \in J$, indicates whether job j is the h-th job in the sequence π. Note that, if such a job is preceded by the maintenance activity in a scenario s, then it is actually in the $(h+1)$-th position of the realization schedule $\sigma(\pi, s)$. We use an additional set of binary variables, defined for all $s \in \mathcal{S}$, to control the position of the maintenance activity M: In particular, $y_h(s)$ indicates whether M is scheduled between the $(h-1)$-th and h-th jobs in $\sigma(\pi, s)$, for all $h = 2, \ldots, n$, while $y_1(s)$ and $y_{n+1}(s)$ are equal to 1 if M is scheduled before and after all the jobs, respectively. Note that, differently from $x \in \{0,1\}^{n^2}$, these variables depend on the realization of P and therefore, in general, they take different values depending on the particular scenario s. Variables $x \in \{0,1\}^{n^2}$, together with the position of the maintenance activity, actually define our solution schedule.

Variable $C_h(s)$ is defined for all positions $h = 1, \ldots, n$ and all scenarios $s \in \mathcal{S}$, and denotes (an upper bound on) the completion times of the job in position h in π. Similarly, variable $C_M(s)$, defined for all scenarios, indicates the completion time of the maintenance activity, in $\sigma(\pi, s)$.

In the following model we also use

$$\theta_h(s) = \sum_{j \in J} p_j x_{jh} \qquad h = 1, \ldots, n, \ s \in \mathcal{S} \tag{15}$$

$$\alpha_h(s) = \sum_{i=1}^{h} y_i(s) \qquad h = 1, \ldots, n, \ s \in \mathcal{S} \tag{16}$$

$\theta_h(s)$ refers to the processing time of the job in position h of π while, $\alpha_h(s)$ takes value 1 if the maintenance activity is placed *before* the h-th job in $\sigma(\pi, s)$. The absolute regret value corresponding to the solution given by sequence π in the scenario $s \in \mathcal{S}$ is then $\sum_{h=1}^{n} C_h(s) - \bar{C}^*(s)$, where $\bar{C}^*(s) = \sum_{j \in J} C_j^*(s)$ is the optimal solution value of the problem in the scenario $s \in \mathcal{S}$.

Hence, the proposed MILP model is the following:

$$\min \max_{s \in \mathcal{S}} \left\{ \sum_{h=1}^{n} C_h(s) - \bar{C}^*(s) \right\} \tag{17}$$

$$\sum_{h=1}^{n} x_{jh} = 1 \qquad j \in J \tag{18}$$

$$\sum_{j \in J} x_{jh} = 1 \qquad h = 1, \ldots, n \tag{19}$$

$$\sum_{h=1}^{n+1} y_h(s) = 1 \qquad s \in \mathcal{S} \tag{20}$$

$$C_h(s) \geq C_{h-1}(s) + \theta_h(s) \qquad h = 1, \ldots, n, \; s \in \mathcal{S} \tag{21}$$

$$C_h(s) \geq C_M(s) + \theta_h(s) - Q(1 - \alpha_h(s)) \qquad h = 1, \ldots, n, \; s \in \mathcal{S} \tag{22}$$

$$C_M(s) \geq C_h(s) + P(s) - Q\alpha_h(s) \qquad h = 1, \ldots, n, \; s \in \mathcal{S} \tag{23}$$

$$C_M(s) \geq r + P(s) \qquad s \in \mathcal{S} \tag{24}$$

$$C_M(s) \leq d \qquad s \in \mathcal{S} \tag{25}$$

$$x_{jh} \in \{0, 1\} \qquad j \in J, \; h = 1, \ldots, n \tag{26}$$

$$y_h(s) \in \{0, 1\} \text{ and } C_h(s) \geq 0 \qquad h = 1, \ldots, n, \; s \in \mathcal{S} \tag{27}$$

$$C_M(s) \geq 0 \qquad s \in \mathcal{S} \tag{28}$$

Being $\theta_h(s)$ and $\alpha_h(s)$ quantities dependent on x_{jh} and $y_h(s)$ according to (15) and (16), to improve readability we omit their definition as variables in program (17)–(28).

The first three constraints are standard assignment constraints. Additional constraints rule the values $C_h(s)$ of the completion times, in the different scenarios. Constraints (21) and (22) define a lower bound on the completion times of jobs (C_0 is set equal to zero). In constraints (22), Q is a suitable large number that can be set equal to the total processing time of the jobs (including the processing time of the maintenance activity). Constraints (23) and (24) define a lower bound on the completion time of the maintenance activity, and constraints (25) set the maximum completion time of the maintenance activity. The objective function, i.e. the maximum regret, can be easily linearized.
The number of variables and constraints in the above MILP formulation is $O(n^2 + n|\mathcal{S}|)$ and $O(n|\mathcal{S}|)$, respectively.

6.2 Heuristic algorithms

In this section, we present some simple heuristic algorithms for $RSMP(\sum C_j, ABS)$, which have been implemented and tested as illustrated in Section 6.3. We know from [7] that the worst case approximation ratio of the SPT algorithm may be arbitrarily large, however it is expected that SPT performs well in the average case and therefore it is worth to assess its performance in practice. Moreover, two of the proposed procedures are based on a local search mechanism which takes the SPT ordering as an initial solution. Together with SPT and the local search procedures, we also considered the additional solution provided by π^A, i.e., the optimal sequence for one scenario which minimizes the worst-case absolute regret. More precisely, let π_s^* be an optimal solution sequence for the (deterministic) problem associated to scenario $s \in \mathcal{S}$, i.e. $\sum C_j(\sigma(\pi_s^*, s)) = \bar{C}^*(s)$. Then we use the sequence $\pi^A = \pi_\varsigma^*$, where

$$\varsigma = \arg\min_{s \in \mathcal{S}} \left\{ \max_{s' \in \mathcal{S}} \left\{ \sum_{j \in J} C_j(\sigma(\pi_s^*, s')) - \bar{C}^*(s') \right\} \right\}. \tag{29}$$

For the sake of shortness, in Section 6.3, the solution π^A is referred to as BS (*Best Scenario-optimum*).

Hereafter, the two simple local search algorithms, called LS_1 and LS_2, are described.

6.2.1 Heuristic LS_1

Starting from an initial solution, Algorithm LS_1 is a local search algorithm that changes, if it is beneficial, the positions of two jobs scheduled after and before the maintenance activity M.

More precisely, for a given scenario $s \in \mathcal{S}$ and the corresponding realization schedule $\sigma(SPT, s)$, we evaluate the following solution sequences: For each job j, scheduled before M in $\sigma(SPT, s)$, we consider the sequence obtained by exchanging the positions of j and the first job k placed immediately after the maintenance activity M.

Eventually, the local search algorithm outputs the best sequence π (i.e., the one minimizing $ABS(\pi)$) among those obtained by applying the above local search phase to all possible realization schedules $\sigma(SPT, s)$, for each scenario $s \in \mathcal{S}$.

6.2.2 Heuristic LS_2

Starting from an initial solution, Algorithm LS_2 also tries to change the positions of two jobs, but in a "symmetric way" with respect to LS_1.

For a given scenario $s \in \mathcal{S}$ and the corresponding realization schedule $\sigma(SPT, s)$, the last job k placed before the maintenance activity M is moved right after M, and a job j that was placed after M in $\sigma(SPT, s)$ is moved to the original position of k. As for LS_1, the actual job j which is eventually assigned to the former position of k is selected by a similar local search phase which evaluates, one by one, the sequences obtained exchanging j (scheduled after the maintenance activity M in $\sigma(SPT, s)$) and k.

The heuristic returns the best sequence π (i.e., the one minimizing $ABS(\pi)$) obtained by applying the above local search phase to the realization schedules $\sigma(SPT, s)$ of all possible scenarios $s \in \mathcal{S}$.

6.3 Computational results

In order to assess the quality of the proposed MILP formulation and the heuristic algorithms, we performed a number of tests on several pseudo-randomly generated instances. We generated 96 classes of 20 instances each, varying the following parameters:

- The number of jobs: $n \in \{25, 50, 75, 100\}$.

$s \in \mathcal{S}^1$:	s_1^1	s_2^1	s_3^1
$P(s)$:	12	25	50

$s \in \mathcal{S}^2$:	s_1^2	s_2^2	s_3^2
$P(s)$:	100	125	150

Table 3: Test instances: description of the scenarios.

- The processing times of the jobs: uniformly distributed in interval $[1, 25]$.
- The scenario sets $\mathcal{S}$: two sets of different scenarios have been considered for the duration of the maintenance activity, denoted as $\mathcal{S}^1 = \{s_1^1, s_2^1, s_3^1\}$ and $\mathcal{S}^2 = \{s_1^2, s_2^2, s_3^2\}$. The corresponding durations for the maintenance activity M are given in Table 3. Those values have been chosen in such a way that, in $\mathcal{S}^1$ the durations of M have similar order of magnitude to the jobs processing times, while in $\mathcal{S}^2$ M is much longer than the jobs.
- The left boundary r of the allotted time window for M: In each instance, it is computed as follows,
$$r = \left\lfloor \alpha \sum_{i=1}^{n} p(i) \right\rfloor,$$
with four different values for α, namely $\alpha \in \{0.1, 0.25, 0.5, 0.75\}$. Therefore the values taken by α indicate the location of the maintenance time window within the schedule: at the very beginning, in the first quarter, in the middle of the schedule, or in the last quarter.
- The right boundary d of the maintenance time window is set as follows
$$d = r + P(s_{\max}) + \lfloor \beta P(s_{\max}) \rfloor,$$
with four different values for β: $\beta \in \{0, 0.01, 0.025, 0.05\}$. Hence, a larger value of β indicates a greater allowance of the maintenance time window, i.e., larger intervals available for scheduling M. The *slack* time $\lfloor \beta P(s_{\max}) \rfloor$ for M, is therefore proportional to β.

All the heuristic algorithms described above were run on a laptop equipped with 2.2 Ghz processor and 8 Gb of RAM. The MILP formulation has been solved by Cplex 12.5 running on a PC, equipped with Intel i7 processor and 64 Gb of RAM.

Tables 4–6 illustrate the experimental results on the two sets of scenarios $\mathcal{S}^1$ and $\mathcal{S}^2$. In Tables 4 and Tables 5, the objective function values obtained by the different algorithms are reported. There, the first three columns refer to the instance parameters; more precisely, column 1 indicates the number of jobs n, while the information on the maintenance activity time window are in columns 2 (the position of the window in the overall schedule depends on α) and 3 (the slack with respect to the value of Mdepends on β). Column 4 reports the optimal solution obtained by the MILP formulation presented in Section 6.1, columns 5–6 respectively are the solution attained by the SPT rule and the Best Scenario-optimum (as defined in (29)), and columns 7–8 show the worst case absolute regret attained by the heuristic algorithms, LS_1 and LS_2, introduced in Section

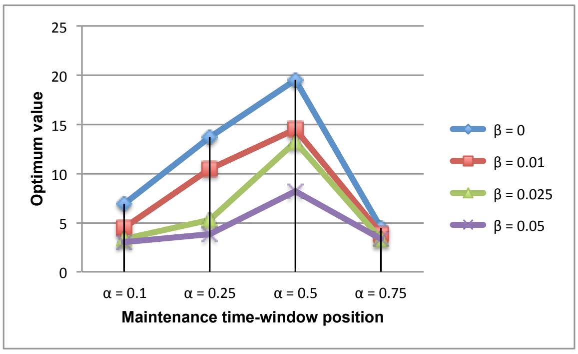
n	α	β	Opt	SPT	BS	LS_1	LS_2
25	0.1	0	6.90	52.95	**9.45**	10.20	14.35
25	0.1	0.01	4.55	34.35	9.75	**6.20**	7.60
25	0.1	0.025	3.30	27.25	7.85	**4.45**	5.95
25	0.1	0.05	2.95	23.70	7.50	**3.85**	5.05
25	0.25	0	13.70	71.60	**18.55**	24.45	24.15
25	0.25	0.01	10.50	57.85	**16.10**	18.20	18.45
25	0.25	0.025	5.20	32.80	9.40	7.90	**7.75**
25	0.25	0.05	3.80	24.40	**4.25**	5.45	5.50
25	0.5	0	19.50	54.20	43.80	**28.35**	33.40
25	0.5	0.01	14.55	44.05	34.20	**22.8**	25.90
25	0.5	0.025	13.25	39.00	30.75	**20.15**	22.70
25	0.5	0.05	8.15	24.00	15.50	**11.75**	13.05
25	0.75	0	4.40	9.65	9.65	**4.80**	**4.80**
25	0.75	0.01	3.90	8.70	8.30	**4.20**	**4.20**
25	0.75	0.025	3.35	7.90	7.85	3.65	**3.50**
25	0.75	0.05	3.35	7.90	7.45	3.65	**3.50**
50	0.1	0	11.70	129.50	36.65	**24.30**	30.65
50	0.1	0.01	10.70	120.30	32.55	**19.70**	25.25
50	0.1	0.025	8.85	98.75	24.85	**14.50**	21.50
50	0.1	0.05	4.45	57.45	16.70	**7.65**	11.35
50	0.25	0	25.40	164.80	**44.80**	54.20	63.60
50	0.25	0.01	16.20	106.45	33.50	**30.75**	37.30
50	0.25	0.025	14.10	95.90	27.15	**27.05**	32.90
50	0.25	0.05	11.00	76.15	**19.10**	20.90	27.15
50	0.5	0	22.35	81.90	**41.55**	43.00	42.65
50	0.5	0.01	19.75	72.15	**33.50**	35.70	36.70
50	0.5	0.025	19.05	72.15	**35.40**	35.65	36.55
50	0.5	0.05	17.20	65.20	**30.95**	31.30	33.45
50	0.75	0	12.05	28.40	27.95	**18.00**	19.15
50	0.75	0.01	11.70	28.45	27.40	**18.00**	19.45
50	0.75	0.025	11.65	28.45	27.40	**18.00**	19.00
50	0.75	0.05	10.85	26.40	25.60	**16.45**	17.75
75	0.1	0	14.75	186.85	67.75	**31.10**	43.00
75	0.1	0.01	9.50	134.75	37.85	**18.55**	27.90
75	0.1	0.025	5.55	91.65	27.20	**11.05**	15.15
75	0.1	0.05	3.45	63.50	23.55	**6.85**	9.45
75	0.25	0	25.90	202.30	77.25	**61.45**	65.20
75	0.25	0.01	15.90	142.30	43.40	36.00	**35.45**
75	0.25	0.025	15.85	142.30	43.15	36.00	**34.20**
75	0.25	0.05	13.70	125.30	40.45	30.45	**28.30**
75	0.5	0	22.95	107.30	**43.55**	47.00	50.30
75	0.5	0.01	23.90	107.30	**43.30**	47.00	49.75
75	0.5	0.025	25.95	107.30	49.70	**47.00**	50.75
75	0.5	0.05	26.00	107.30	49.05	**47.00**	51.75
75	0.75	0	23.65	57.05	53.05	40.10	**39.80**
75	0.75	0.01	23.35	57.05	52.30	40.10	**39.90**
75	0.75	0.025	21.15	53.30	48.75	37.25	**36.90**
75	0.75	0.05	18.20	45.20	40.65	31.00	**30.65**
100	0.1	0	19.70	281.70	110.65	**45.90**	61.65
100	0.1	0.01	15.75	230.85	85.40	**33.90**	47.30
100	0.1	0.025	12.50	195.80	76.00	**27.20**	37.45
100	0.1	0.05	3.20	66.90	21.10	**6.40**	7.95
100	0.25	0	18.55	189.40	72.65	**51.60**	53.00
100	0.25	0.01	18.70	189.40	63.90	**51.60**	52.35
100	0.25	0.025	13.55	144.60	45.05	**34.45**	35.30
100	0.25	0.05	10.35	123.50	40.40	**24.40**	26.30
100	0.5	0	46.15	197.00	**80.55**	119.10	110.85
100	0.5	0.01	35.65	161.25	**59.30**	92.50	87.40
100	0.5	0.025	20.75	103.10	**31.95**	51.80	49.10
100	0.5	0.05	22.10	103.10	**34.45**	51.80	50.00
100	0.75	0	27.75	70.75	55.50	50.70	**50.25**
100	0.75	0.01	21.60	59.85	41.90	**41.80**	41.85
100	0.75	0.025	21.70	59.85	**41.55**	41.80	42.60
100	0.75	0.05	20.05	54.50	**37.25**	37.50	38.05
Averages			15.10	89.61	36.94	30.09	32.41

Table 4: Maximum regret values for the different algorithms on instances with scenario set $\mathcal{S}^1$.

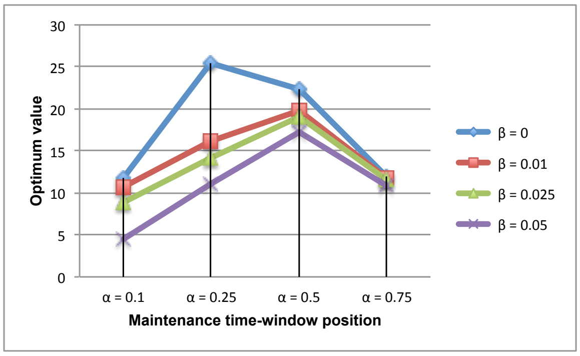
n	α	β	Opt	SPT	BS	LS_1	LS_2
25	0.1	0	6.90	52.95	12.75	**10.20**	14.80
25	0.1	0.01	3.30	27.25	5.10	**4.45**	5.95
25	0.1	0.025	2.35	20.70	**3.00**	3.10	3.45
25	0.1	0.05	0.20	0.65	**0.20**	**0.20**	**0.20**
25	0.25	0	13.70	71.60	**21.35**	24.45	26.25
25	0.25	0.01	5.20	32.80	8.80	7.90	**7.15**
25	0.25	0.025	3.00	19.80	**3.25**	3.95	3.55
25	0.25	0.05	1.70	13.10	**1.80**	2.15	2.00
25	0.5	0	19.35	54.20	46.45	**28.35**	35.10
25	0.5	0.01	13.25	39.00	33.10	**20.15**	23.20
25	0.5	0.025	8.30	24.00	17.45	**11.75**	13.55
25	0.5	0.05	5.25	16.30	12.50	**7.15**	9.55
25	0.75	0	4.40	9.65	9.50	**4.80**	**4.80**
25	0.75	0.01	3.35	7.90	7.90	3.65	**3.50**
25	0.75	0.025	2.80	6.35	6.35	3.00	**2.95**
25	0.75	0.05	2.05	5.10	5.05	**2.15**	2.20
50	0.1	0	11.70	129.50	34.30	**24.3**	47.45
50	0.1	0.01	8.85	98.75	25.75	**14.5**	23.80
50	0.1	0.025	3.60	49.05	9.70	**6.15**	11.00
50	0.1	0.05	0.00	**0.00**	**0.00**	**0.00**	**0.00**
50	0.25	0	25.45	164.80	**51.90**	54.20	76.40
50	0.25	0.01	14.10	95.90	30.85	**27.05**	39.85
50	0.25	0.025	7.75	54.65	15.70	**14.80**	19.00
50	0.25	0.05	2.20	19.25	5.70	**3.70**	3.95
50	0.5	0	21.95	81.95	63.25	**43.00**	48.10
50	0.5	0.01	18.90	72.15	43.95	**35.65**	39.60
50	0.5	0.025	16.65	65.20	49.50	**31.30**	38.70
50	0.5	0.05	10.65	44.55	23.35	**19.20**	23.55
50	0.75	0	12.05	28.45	27.85	**18.00**	19.50
50	0.75	0.01	11.65	28.40	27.65	**17.95**	19.35
50	0.75	0.025	10.60	24.20	23.35	**15.00**	17.00
50	0.75	0.05	7.65	17.70	16.95	**10.70**	12.95
75	0.1	0	14.60	186.85	63.20	**31.10**	69.65
75	0.1	0.01	5.55	91.50	27.30	**11.05**	22.80
75	0.1	0.025	2.50	47.65	10.95	**4.60**	8.20
75	0.1	0.05	0.05	2.60	**0.05**	**0.05**	**0.05**
75	0.25	0	25.90	202.30	87.95	**61.45**	77.25
75	0.25	0.01	15.85	142.30	45.90	**36.00**	42.15
75	0.25	0.025	12.30	109.90	42.30	**26.20**	38.25
75	0.25	0.05	2.30	22.80	7.55	**3.70**	4.00
75	0.5	0	23.10	107.30	71.60	**47.00**	61.35
75	0.5	0.01	25.15	107.30	76.95	**47.00**	63.30
75	0.5	0.025	24.10	107.30	79.50	**47.00**	62.15
75	0.5	0.05	13.40	70.25	32.25	**26.50**	34.05
75	0.75	0	24.00	57.05	51.65	**40.10**	41.55
75	0.75	0.01	21.50	53.35	47.40	**37.30**	37.35
75	0.75	0.025	19.25	45.20	40.25	**31.00**	33.65
75	0.75	0.05	13.00	30.00	25.35	**20.40**	21.70
100	0.1	0	19.65	281.70	94.55	**45.90**	76.20
100	0.1	0.01	12.50	195.80	76.70	**27.20**	68.15
100	0.1	0.025	2.35	53.95	14.50	**4.90**	6.55
100	0.1	0.05	0.00	**0.00**	**0.00**	**0.00**	**0.00**
100	0.25	0	18.60	189.40	85.95	**51.60**	67.25
100	0.25	0.01	13.35	144.60	54.15	**34.45**	46.55
100	0.25	0.025	11.20	123.50	44.80	**24.40**	35.95
100	0.25	0.05	3.30	51.95	20.85	**6.10**	9.20
100	0.5	0	49.85	197.00	**112.40**	119.10	121.30
100	0.5	0.01	19.75	103.10	52.75	**51.80**	52.20
100	0.5	0.025	17.05	86.85	48.90	**41.20**	48.65
100	0.5	0.05	10.00	60.95	31.05	**23.65**	31.05
100	0.75	0	27.90	70.75	52.30	**50.70**	51.65
100	0.75	0.01	21.80	59.85	44.85	**41.80**	42.25
100	0.75	0.025	21.85	54.50	41.45	**37.50**	39.85
100	0.75	0.05	14.30	37.10	27.85	**24.15**	27.05
Averages			12.26	69.85	33.71	23.87	30.31

Table 5: Maximum regret values for the different algorithms on instances with scenario set $\mathcal{S}^2$.

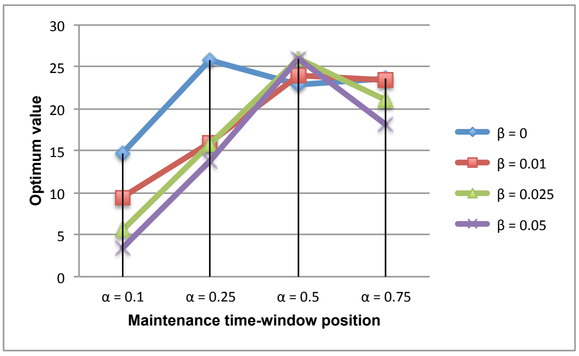
6.2. The results of each row are average values over the 20 instances. The values in bold correspond to the best average results obtained by one of the heuristics. The last row of each table reports the overall average results. The results of Table 4 are also illustrated through Figure 2, where we present four charts showing how the optimal regret values vary depending on the maintenance window position.



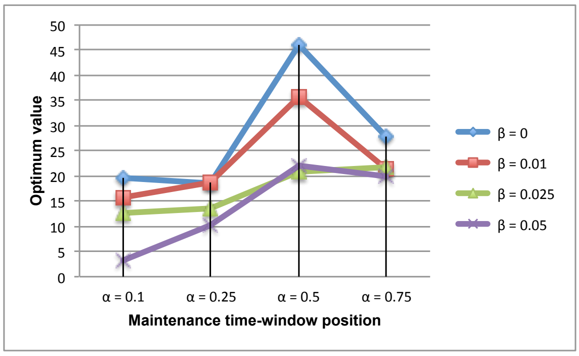
(a) $n = 25$



(b) $n = 50$



(c) $n = 75$



(d) $n = 100$

Figure 2: Variations of the optimal regret values depending on the maintenance time-window position (α) and slack (β).

A few comments are in order. A first observation on the presented results is that the solution value (i.e., the minmax regret) decreases as β increases, i.e. when the $[r, d]$ window becomes larger. This is quite natural, since a larger slack allows to better schedule M and therefore avoid possible idle times. For the same reason, the regret values of the instances of the scenario set $\mathcal{S}^2$ are, most of the times, smaller than those of scenarios in $\mathcal{S}^1$, since the $[r, d]$ window allowance is proportional to the M length which is larger in those instances. However, when $\beta = 0$ the maintenance window equals the duration of M in the scenario $s_{\max}$. For this reason, in this case, the maximum regret values for the scenario sets $\mathcal{S}^1$ and $\mathcal{S}^2$ are often identical.

Looking at Figure 2, it is immediate to note that when β and n are fixed, the highest regret values—most of the times—correspond to $\alpha = 0.5$ (and in some cases, when $\beta = 0$, the maximum regret is attained for $\alpha = 0.25$).

As one may notice by looking at Table 4, the optimal absolute regret values are slightly sensitive to the number of jobs. Indeed, those values are larger when the number of jobs increases (the highest regret values correspond, depending on the β values, to the instances

n	α	β	CPU times on set $\mathcal{S}^1$	CPU times on set $\mathcal{S}^2$
25	0.1	0	532	482
25	0.1	0.01	470	460
25	0.1	0.025	464	485
25	0.1	0.05	459	411
25	0.25	0	466	431
25	0.25	0.01	508	429
25	0.25	0.025	443	435
25	0.25	0.05	428	448
25	0.5	0	464	589
25	0.5	0.01	491	587
25	0.5	0.025	454	497
25	0.5	0.05	452	444
25	0.75	0	460	510
25	0.75	0.01	461	460
25	0.75	0.025	470	470
25	0.75	0.05	460	440
50	0.1	0	1931	1941
50	0.1	0.01	1535	2008
50	0.1	0.025	1681	2187
50	0.1	0.05	1498	1421
50	0.25	0	1930	2329
50	0.25	0.01	2028	2086
50	0.25	0.025	1673	1899
50	0.25	0.05	1580	2417
50	0.5	0	1616	1835
50	0.5	0.01	1635	1872
50	0.5	0.025	1712	1880
50	0.5	0.05	2308	1632
50	0.75	0	1150	1220
50	0.75	0.01	1191	1291
50	0.75	0.025	1180	1331
50	0.75	0.05	1230	1390
75	0.1	0	6491	8898
75	0.1	0.01	5475	5420
75	0.1	0.025	5003	4922
75	0.1	0.05	3467	3993
75	0.25	0	5211	7480
75	0.25	0.01	5668	8115
75	0.25	0.025	4139	7106
75	0.25	0.05	4448	4395
75	0.5	0	5384	6023
75	0.5	0.01	4919	7180
75	0.5	0.025	4568	5601
75	0.5	0.05	4140	5966
75	0.75	0	3680	2690
75	0.75	0.01	3481	3370
75	0.75	0.025	2921	2880
75	0.75	0.05	3030	2740
100	0.1	0	13427	14180
100	0.1	0.01	16265	13965
100	0.1	0.025	11292	12480
100	0.1	0.05	8878	10801
100	0.25	0	10037	17134
100	0.25	0.01	10296	15896
100	0.25	0.025	10815	16288
100	0.25	0.05	9391	10758
100	0.5	0	11637	17432
100	0.5	0.01	9263	12320
100	0.5	0.025	11085	11205
100	0.5	0.05	11060	9709
100	0.75	0	8650	9030
100	0.75	0.01	6610	5970
100	0.75	0.025	5910	7551
100	0.75	0.05	7790	6370

Table 6: Computation times required by the MILP formulation on both instance sets (in ms).

with 100 and 75 jobs). However, comparing the optimal regret values to the average values of total completion times, it is clear that the regrets grow at a lower rate. For instance, for $\beta = 0$, the largest average value ($ABS = 46.15$), attained at $n = 100$ is around an order of magnitude larger than the smallest value ($ABS = 4.4$) at $n = 25$, while the actual total completion time values range between $6 \cdot 10^3$ and $3.5 \cdot 10^5$. As a consequence, the ratio between the values maximum regret and actual objective (total completion time) is decreasing when the number of jobs becomes larger. The latter ratio is related to the so called maximum *relative* regret which is shortly discussed in Section 4.2.

As for the instances in the second scenario set $\mathcal{S}^2$ (Table 5) the trends of the optimal regret values are quite similar to those described above for the first set $\mathcal{S}^1$.

Since LS_1 and LS_2 are local search procedures starting from the SPT solution, then—as expected—the latter one always attains the worst results among the heuristic algorithms. From Tables 4 and 5, we note that heuristics LS_1 and LS_2 perform better than BS. Furthermore, in most of the cases, LS_1 attains the best performance (best average values in 35 out of 64 instance classes for scenarios in $\mathcal{S}^1$, 55 out of 64 instance classes for scenarios in $\mathcal{S}^2$). On the other hand, BS has the best performances on the instances with scenario set $\mathcal{S}^1$, for $n \in \{50, 100\}$ and $\alpha = 0.5$ (see Table 4).

In scenario set $\mathcal{S}^1$, the relative error of heuristic LS_1 with respect to the optimal regret $\varepsilon = \frac{ABS_{LS_1} - ABS^*}{ABS^*}$ (where ABS^* and ABS_{LS_1} are, for a given instance, the optimal regret and the value provided by the heuristic algorithm) ranges between 7.7% and 178.2%. Such an error remains below 10% in around 6% of the instances (when $n = 25$ and $\alpha = 0.75$). The improvement of the local search algorithms, with respect to the value provided by SPT, can be quantified comparing the entries of columns 5 and 7. Such an improvement rises up to more than 900% (see instances of scenario set $\mathcal{S}^1$, $n = 100$, $\alpha = 0.1$ and $\beta = 0.05$) and it is more evident for those classes with $\alpha = 0.1$, i.e., when the maintenance interval is positioned early in the schedule. It is also worth to note that the values of the absolute regret obtained by the algorithms (including SPT), are always two or more orders of magnitude smaller than the values of the actual objective, i.e., total jobs completion time.

In scenario set $\mathcal{S}^2$, (see Table 5) the performance of the heuristics shows a slight improvement: the local search algorithms find an optimal solution for all instances with $\alpha = 0.1$ and $\beta = 0.05$. Furthermore, in 13% of the instances the relative error is not larger than 10%.

All the heuristics are very fast, requiring less than 1 millisecond per instance, on a standard laptop. Table 6 reports the average computational times in milliseconds required by Cplex on the all instances. Observe that, Cplex requires less than 20 seconds for all instances.

7 Conclusions

In this paper, a problem arising in a manufacturing environment concerning the joint scheduling of multiple jobs and of a maintenance activity on a single machine has been addressed. The maintenance activity must be processed within a given time window and its duration is not known *a-priori*, though it takes values in a given interval.

In this context, the problem of finding jobs schedules which are *robust* to any possible change in the maintenance activity duration has been addressed. We investigate properties of robust schedules when minimizing makespan and total completion time of the jobs, under four different standard robustness criteria. In addition, for the seemingly hardest problem in which the objective is the total completion time and the maximum absolute regret criterion is adopted, a mixed integer linear program and a number of heuristic algorithms have been evaluated. The results of an extensive computational campaign show the efficiency and effectiveness of the proposed solution approaches.

Future research directions include a theoretical study to characterize properties of the robust schedules, an extension of the results to the case in which precedence among jobs are given (see, for instance [21]), possible development of solution algorithms based on a dynamic programming approach similar to that of [27], the design of new heuristic algorithms based on metaheuristic techniques.

Acknowledgements

The authors thank the associate editor and one anonymous reviewer for their valuable suggestions and comments.

References

[1] Adiri I., J. Bruno, E. Frostig, A. H. G. Rinnooy Kan (1989). Single machine flow-time scheduling with a single breakdown, Acta Informatica, 26(7), 679–696.

[2] Agnetis, A., G. Nicosia, A. Pacifici, U. Pferschy (2015). Scheduling two agent task chains with a central selection mechanism, Journal of Scheduling, 18(3), 243–261.

[3] Chen J.-S. (2008). Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan, European Journal of Operational Research, 190, 90–102.

[4] Chen Y., A. Zhang, Z. Tan (2013). Complexity and approximation of single machine scheduling with an operator non-availability period to minimize total completion time, Information Sciences 251, 150– 163.

[5] Conde E. (2014). A MILP formulation for the minmax regret total completion time in scheduling with unrelated parallel machines, Optimization Letters 8(4), 1577–1589.

[6] Daniels R.L., P. Kouvelis (1995). Robust Scheduling to Hedge Against Processing Time Uncertainty in Single-stage Production, Management Science, 41(2), 363–376.

[7] Detti P., G. Nicosia, A. Pacifici, G. Zabalo Manrique de Lara (2016). Robust single machine scheduling with external-party jobs, proceedings of IFAC conference MIM 2016, IFAC-PapersOnLine, 49(12), 1731–1736.

[8] Garey, M. R., D. S. Johnson, (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. A Series of Books in the Mathematical Sciences. W. H. Freeman and Co.

[9] Ibarra O. H., C. E. Kim (1975). Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems, Journal of the ACM, 22(4), 463–468.

[10] He Y., Ji M., Cheng T.C.E. (2005). Single machine scheduling with a restricted rate-modifying activity, Naval Research Logistics, 52, 361-369.

[11] Kasperski A. and P. Zieliński (2006). An approximation algorithm for interval data minmax regret combinatorial optimization problems, Information Processing Letters, 97(5), 177–180.

[12] Kasperski A. and P. Zielinski (2014). Minmax (regret) scheduling problems, In: Sotskov Y., Werner F. (Eds.), Sequencing and Scheduling with Inaccurate Data, 159–210.

[13] Kellerer H., R. Mansini, U. Pferschy, and M.G. Speranza (2003). An efficient fully polynomial approximation scheme for the Subset-Sum Problem, Journal of Computer and System Sciences, 66(2), 349–370

[14] Kouvelis P., Yu G. (1997). Robust discrete optimization and its applications, Kluwer Academic Publishers, Boston.

[15] Lebedev V., I. Averbakh (2006). Complexity of minimizing the total flow time with interval data and minmax regret criterion, Discrete Applied Mathematics, 154(15), 2167–2177.

[16] Lee C.-Y. (1996). Machine scheduling with an availability constraint, Journal of Global Optimization, 9(3–4), 395–416.

[17] Lu C.C., S.W. Lin, K.C. Ying (2012). Robust scheduling on a single machine to minimize total flow time, Computers & Operations Research, 39(7), 1682–1691.

[18] Ma Y., C. Chu, C. Zuo (2010). A survey of scheduling with deterministic machine availability constraints, Computers & Industrial Engineering, 58(2), 199–211.

[19] Montemanni R. (2007). A Mixed Integer Programming Formulation for the Total Flow Time Single Machine Robust Scheduling Problem with Interval Data, Journal of Mathematical Modelling and Algorithms, 6(2), 287–296.

[20] Mulvey J.M., R.J. Vanderbei, S.A. Zenios (1995). Robust Optimization of Large-Scale Systems, Informs Journal of Computing, 43(2), 264–281.

[21] Nicosia G., A. Pacifici (2017). Scheduling assembly tasks with caterpillar precedence constraints on dedicated machines, International Journal of Production Research, 55(6), 1680–1691.

[22] Pereira J. (2016). The robust (minmax regret) single machine scheduling with interval processing times and total weighted completion time objective, Computers & Operations Research, 66, 141–152.

[23] Xu D., Yin Y., Li H. (2009). A note on "scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan", European Journal of Operational Research 197, 825–827.

[24] Xu D., L. Wan, A. Liu, D. Yang (2015). Single machine total completion time scheduling problem with workload-dependent maintenance duration, Omega 52, 101–106.

[25] Yager R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems, Man and Cybernetics, 18, 183–190.

[26] Yang J., G. Yu (2002). On the Robust Single Machine Scheduling Problem, Journal of Combinatorial Optimization, 6, 17–33.

[27] Yang S., Y. Maa, D. Xu, J. Yang (2011). Minimizing total completion time on a single machine with a flexible maintenance activity, Computers & Operations Research, 38, 755–770.