A Robust Optimization Approach for the Unrelated Parallel Machine Scheduling Problem

Jonathan De La Vega, Alfredo Moreno, Reinaldo Morabito and Pedro Munari
Production Engineering Department, Federal University of São Carlos, São Carlos-SP, Brazil,
jdlvmartinez@gmail.com, alfredomorenoarteaga@gmail.com, morabito@ufscar.br, munari@dep.ufscar.br

Abstract

In this paper, we address the Unrelated Parallel Machine Scheduling Problem (UPMSP) with sequence- and machine-dependent setup times and job due-date constraints. Different uncertainties are typically involved in real-world production planning and scheduling problems. If ignored, they can lead to suboptimal or even infeasible schedules. To avoid this, we present two new robust optimization models for this UPMSP variant, considering stochastic job processing and machine setup times. To the best of our knowledge, this is the first time that a robust optimization approach is used to address uncertain processing and setup times in the UPMSP with sequence- and machine-dependent setup times and job due-date constraints. We carried out computational experiments to compare the performance of the robust models and verify the impact of uncertainties to the problem solutions when minimizing the production makespan. The results of computational experiments indicate that the robust models incorporate uncertainties appropriately into the problem and produce effective and robust schedules. Furthermore, the results show that the models are useful for analyzing the impact of uncertainties in the cost and risk of the scheduling solutions.

Keywords: Production scheduling; Unrelated parallel machines; Sequence-dependent setups; Due date constraints; Uncertain processing and setup times and Robust optimization.

1 Introduction

The parallel machine scheduling problem (PMSP) consists of determining a production schedule of jobs in parallel machines, with each job requiring a single operation in one of the machines, while optimizing a given objective such as the minimization of the maximum job completion time (makespan), total tardiness, mean completion time, and maximum tardiness. Three classes of PMSPs have been considered in the literature (Cheng and Sin, 1990), namely identical, uniform and unrelated PMSP. The unrelated PMSP (UPMSP) is the most general of the three classes (Torabi et al., 2013) and assumes that the machines perform the same functions, but have different processing velocities, resources or capacities. The UPMSP is often found in real manufacturing and service settings, such as in the textile, chemical, electronic, mechanical and service industries (Ying et al., 2012).

Machine setup times and job due date constraints are relevant aspects when addressing the UPMSP in practice. As pointed out in the literature, the consideration of sequence- and machine-dependent setup times and due date constraints is important in different industrial and service contexts to derive feasible and effective production schedules (Arnaout et al., 2014). Due date constraints have broad industrial implications in many manufacturing systems (Lin et al., 2011), as they impose that the due dates of certain jobs from primary customers must not be violated.

In this paper, we consider the UPMSP with due date constraints and sequence- and machine-dependent setup times. In practice, various parameters involved in this variant, such as job processing and machine setup times, are usually uncertain due to machine and/or human factors (Chyu and Chang, 2011; Liao and Su, 2017). These uncertainties are natural in production planning and scheduling problems and they can lead to suboptimal or even infeasible schedules if ignored. In spite of this, few studies presented thus far in the literature have considered such uncertainties.

1.1 Related work

Different approaches have been used to address uncertain data in scheduling environments, such as sensitivity analysis, fuzzy programing, stochastic programming, and robust optimization (Macedo et al., 2016; González-Neira et al., 2017). Most of them considered job due dates, but without explicitly considering due date constraints. Considering this, a fuzzy model for the UPMSP with sequence-dependent setup times was proposed in Gharehgozli et al. (2009). The authors considered uncertain processing times and minimized the total weighted flow time and the total weighted tardiness simultaneously. In Chyu and Chang (2011), the authors used the fuzzy theory to address the UPMSP with sequence- and machine-dependent setup times to minimize the makespan and the average tardiness under fuzzy processing times, fuzzy due dates and fuzzy deadlines for the completion of all jobs. A multi-objective UPMSP with sequence and machine-dependent setup times considering uncertainty in processing times and due dates was introduced in Torabi et al. (2013). The authors proposed a fuzzy approach to minimize the total weighted flow time, the total weighted tardiness, and the machine load variation. Recently, the UPMSP was studied in Liao and Su (2017) taking into account the fuzzy processing, release and sequence-dependent setup times when minimizing the makespan.

Stochastic programing and robust optimization have been used to address variants of the PMSP with identical machines. In Xu et al. (2013), the authors addressed a makespan minimization scheduling problem in identical parallel machines, in which uncertain processing times are represented as intervals by using a min-max regret scheduling model. Robust optimization approaches were proposed in Seo and Chung (2014) for the identical PMSP with uncertain processing times and minimizing the makespan. Neither of these two papers have considered production setup times or due date constraints. A PMSP variant with sequence-dependent setup times was also addressed in Hu et al. (2016), but without any consideration of due dates. The authors developed a scenario based mixed-integer linear programming (MIP) formulation to consider uncertainty in processing and arrival times when minimizing the makespan. A scenario based approach was developed in Feng et al. (2016) to take into account uncertainty in processing times, for the makespan minimization scheduling problem in a two-stage hybrid flow shop. The first production stage has only one machine, while the second stage has identical parallel machines. These two papers identified robust schedules by using min?max regret models based on the worst case scenario.

The UPMSP with sequence and machine-dependent setup times and due date constraints has been studied by a few authors Chen (2009); Lin et al. (2011); Zeidi and MohammadHosseini (2015);

Chen (2015), but without considering uncertainties in the problem parameters. Different objectives have been addressed for this scheduling problem, such as minimizing the total tardiness Chen (2009); Lin et al. (2011), the total cost of tardiness and earliness Zeidi and MohammadHosseini (2015) and the total weighted completion time Chen (2015). Other recent papers deal with the UPMSP with sequence and machine-dependent setup times, but without taking into account due date constraints or uncertainties Ravetti et al. (2007); Arnaout et al. (2014); Nogueira et al. (2014); Avalos-Rosales et al. (2015); Afzalirad and Rezaeian (2016); Salehi Mir and Rezaeian (2016); Sadati et al. (2017); Gomes and Mateus (2017). To the best of our knowledge, there are no studies so far in the literature considering uncertainties in the UPMSP with sequence- and machine-dependent setup times and explicit due date constraints.

1.2 Contributions

In this paper, we contribute to the literature by considering for the first time the UPMSP with due date constraints and sequence- and machine-dependent setup times under the hypothesis of uncertain processing and setup times. Uncertainty is incorporated into the problem via a robust optimization (RO) approach using budgeted uncertainty sets (Ben-Tal and Nemirovski, 2000; Bertsimas and Sim, 2004; Bertsimas et al., 2011). RO has some advantages over other stochastic approaches, such as it does not require probability distribution information to describe parameter uncertainty, the robust counterpart of an uncertain MIP model can be reformulated as a deterministic MIP depending on the uncertainty set, and it provides a feasible solution (uncertainty-immunized solution) for any parameter realization belonging to a predetermined uncertainty set (Seo and Chung, 2014). We propose two RO models for the addressed UPMSP variant, considering the minimization of the makespan as the objective function. We are not aware of other RO models in the literature for any variant of the UPMSP considering sequence and machine-dependent setup times.

The difficulty of considering the classical RO approaches for the addressed problem is that all the uncertain parameters (processing and/or setup times) related to a given machine need to appear in the same model constraint. Hence, we first present two deterministic models for the addressed UMPSP variant that are convenient for using RO and hence overcome the mentioned difficulty. Computational experiments using problem instances from the literature were carried out to compare the performance of the proposed RO models. We analyze the impact of the uncertainties to the solutions and verify the corresponding trade-off between cost and risk (robustness) of the solutions. Furthermore, a risk analysis using a Monte-Carlo simulation indicates the potential of the RO models to address the cost-risk trade-off in the addressed UPMSP variant. It is worth mentioning that the developments presented in this paper are not restricted only to the addressed UPMSP variant and hence can also be used for related variants of the PMSP.

1.3 Organization of the paper

The remainder of this paper is organized as follows. Section 2 describes the UPMSP with sequence- and machine-dependent setup times and due-date constraints and presents two MIP formulations for its deterministic (nominal) variant. These formulations are convenient for developing the RO models proposed in Section 3. In Section 4, we report the results of the computational experiments with the proposed RO approach. Finally, Section 5 presents final remarks and discusses perspectives for future research.

2 Problem description and deterministic formulations

The UPMSP with sequence- and machine-dependent setup times and due-date constraints consists of determining a schedule for a set of jobs $\mathcal{N}$ in a set of unrelated parallel machines $\mathcal{M}$. All machines have the same functions, but with different resources and capacities. Each job requires a single operation in one of these machines. There is a given due date for finishing the processing of each job in the subset $\mathcal{B} \subseteq \mathcal{N}$. These jobs are related to primary customers and their due dates cannot be violated.

In the deterministic (nominal) variant of the problem, all job processing and setup times are assumed to be known in advance. As the machines are unrelated, the (nominal) processing time to perform the operation of a given job depends on the machine. The same happens for the (nominal) setup time to perform the operation of a given job, which further depends on the production sequence of the machine. For the sake of simplicity, we assume that these setup times satisfy the triangular inequality. To take into account the setup time of the first job in the production sequence of a given machine, we define the dummy job 0 and the node set $\mathcal{N}_0 = \mathcal{N} \cup \{0\}$ (Gharehgozli et al., 2009). This dummy job has to be the first job in the sequence of any machine, with null (nominal) processing time and null setup times from each job to it.

Using the sets defined, we state the following input parameters of the problem:

$\bar{p}_{ik}$ (nominal) processing time of job $i \in \mathcal{N}$ in machine $k \in \mathcal{M}$;

$\bar{s}_{ijk}$ (nominal) setup time from job $i \in \mathcal{N}_0$ to job $j \in \mathcal{N}$ in machine $k \in \mathcal{M}$;

b_i due date of job $i \in \mathcal{B}$;

V sufficiently large number.

The objective of the problem is to determine the sequences of jobs in the machines that minimize the production makespan (i.e., the total time required to finish all jobs). Different formulations are available in the literature for the addressed UPMSP variant (Avalos-Rosales et al., 2015; Gomes and Mateus, 2017). We adapt these formulations in the remainder of this section, to make them more convenient for incorporating uncertainties via RO.

2.1 Deterministic precedence-based formulation

The addressed UPMSP variant can be formulated as a MIP model using decision variables that describe job precedences. Consider the following decision variables:

C_{max} completion time of the last processed job;

C_{ik} completion time of job $i \in \mathcal{N}_0$ in machine $k \in \mathcal{M}$;

R_{ik} position of job $i \in \mathcal{N}_0$ in the production sequence of machine $k \in \mathcal{M}$;

x_{ijk} 1, if job $j \in \mathcal{N}$ is processed immediately after job $i \in \mathcal{N}_0$ in machine $k \in \mathcal{M}$; 0, otherwise;

z_{ijk} 1, if the processing of job $i \in \mathcal{N}_0$ precedes the processing of job $j \in \mathcal{N}$ in machine $k \in \mathcal{M}$; 0, otherwise.

Notice that we have defined two types of variables to define the production sequence in the machine, namely x_{ijk} and z_{ijk}. The difference between them is that $x_{ijk} = 1$ means that job i precedes job j immediately in the sequence of machine k, while $z_{ijk} = 1$ means that job i precedes job j, but not necessarily immediately. These two types of variables are necessary to guarantee that all uncertain parameters related to the jobs assigned to a given machine are available in the same constraint, which makes the model convenient for using RO.

Using the decision variables and parameters previously defined, the deterministic precedence-based formulation is:

Minimize
$$C_{max} \tag{2.1}$$

Subject to:
$$\sum_{\substack{i \in \mathcal{N}_0 \\ i \neq j}} \sum_{k \in \mathcal{M}} x_{ijk} = 1, \forall j \in \mathcal{N}, \tag{2.2}$$
$$\sum_{\substack{i \in \mathcal{N}_0 \\ i \neq h}} x_{ihk} - \sum_{\substack{j \in \mathcal{N}_0 \\ j \neq h}} x_{hjk} = 0, \forall h \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.3}$$
$$\sum_{j \in \mathcal{N}_0} x_{0jk} = 1, \forall k \in \mathcal{M}, \tag{2.4}$$
$$\sum_{i \in \mathcal{N}} \bar{p}_{ik} z_{ijk} + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \bar{s}_{lik} x_{lik} z_{ljk} \leq C_{jk}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.5}$$
$$C_{ik} \leq b_i, \forall i \in \mathcal{B}, \forall k \in \mathcal{M}, \tag{2.6}$$
$$C_{max} \geq C_{ik}, \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.7}$$
$$R_{jk} \geq R_{ik} + 1 - |\mathcal{N}|(1 - x_{ijk}), \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \tag{2.8}$$
$$V z_{ijk} \geq R_{jk} - R_{ik} - V \left[2 - \sum_{l \in \mathcal{N}_0} x_{lik} - \sum_{l \in \mathcal{N}_0} x_{ljk} \right], \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \tag{2.9}$$
$$2x_{ijk} \leq z_{iik} + z_{jjk}, \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \tag{2.10}$$
$$\sum_{k \in \mathcal{M}} z_{iik} = 1, \forall i \in \mathcal{N}, \tag{2.11}$$
$$C_{0k} = 0, \forall k \in \mathcal{M}, \tag{2.12}$$
$$C_{ik} \geq 0, \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.13}$$
$$0 \leq R_{ik} \leq |\mathcal{N}|, \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.14}$$
$$x_{ijk}, z_{ijk} \in \{0, 1\}, \forall i, j \in \mathcal{N}_0, \forall k \in \mathcal{M}. \tag{2.15}$$

The objective function (2.1) consists of minimizing the total time required to process all jobs (makespan). Constraints (2.2)-(2.4) are the flow constraints: (2.2) ensure that each job j must be processed in exactly one machine; (2.3) impose that each job h processed in a machine has an immediate predecessor and an immediate successor; and (2.4) ensure that the dummy job 0 is the immediate predecessor of the first actual job processed in each machine. Note that in accordance with these constraints, the dummy job 0 must immediately precede the first and immediately succeed the last actual jobs in each of the $|\mathcal{M}|$ machines.

Constraints (2.5) determine the completion time of the jobs assigned to each machine. For a given job $j \in \mathcal{M}$ and machine $k \in \mathcal{M}$, the first term on the left-hand side of (2.5) computes the processing times of all jobs that are processed before j in the sequence of machine k, while the second term computes all the setup times corresponding to changing from job l to job i, such that l immediately precedes i in the sequence, and l and i are processed before j. This computation requires the product of variables $x_{lik}z_{ijk}$, as the setup times depend on the immediate precedence of jobs in the machine (expressed by x_{lik}) and we are interested only in those jobs that are processed before j and in the same machine (expressed by z_{ljk}). In fact, we have $z_{ljk} = 1$ for any job l that is processed before j in machine k; thus, $x_{lik}z_{ljk} = 1$ if and only if l is the job that immediately precedes i in the same machine (notice that constraints (2.2) guarantee that there must be exactly one job in $l \in \mathcal{N}$ that immediately precedes node i). If job l does not precede j in machine k, then $z_{ljk} = 0$ and the product of variables is equal to zero.

Constraints (2.6) ensure that the due dates of jobs of primary customers are met. The makespan is computed by constraints (2.7), together with the objective function. Note that C_{max} corresponds to the completion time of the last processed job. Constraints (2.8) link the variables R_i and x_{ijk} based on two jobs i and j such that i immediately precedes j. Constraints (2.9) links variables z_{ijk} to R_i and x_{ijk} based on the jobs that are processed before job j in machine k. To determine if job i precedes job j, job i needs to be processed before job j (i.e., $R_{jk} > R_{ik}$) and both jobs need to be processed in the same machine k. This second requirement is guaranteed by the expression inside the brackets on the right-hand side of the constraints, which is zero only if i and j have an immediate predecessor in machine k (and hence both are processed in the same machine). If in addition $R_{jk} > R_{ik}$, then variable z_{ijk} assumes the value of 1 (recall that it is a binary variable).

Constraints (2.10) are added to the formulation to impose that two different jobs i and j are predecessors of themselves if they are processed in the same machine. Constraints (2.11) impose that job i is predecessor of itself in a unique machine. A null value for the completion time of job 0 in each machine is set by constraints (2.12). Finally, constraints (2.13), (2.14) and (2.15) determine the type and domain of the decision variables.

The formulation (2.1)-(2.15) is a non-linear model because of the product of variables in constraints (2.5). Nevertheless, this product of binary variables can be linearized by adding binary variables to the model, together with a set of linear constraints. Let y_{lijk} be a new binary variable that assumes the value of 1 if and only if job l immediately precedes job i and both jobs (l and i) are processed before job j in machine k. The following set of constraints corresponds to the linearization of constraints (2.5):

$$\sum_{i \in \mathcal{N}} \bar{p}_{ik} z_{ijk} + \sum_{i \in \mathcal{N}} \sum_{l \in \mathcal{N}_0} \bar{s}_{lik} y_{lijk} \leq C_{jk}, \; \forall j \in \mathcal{N}, \; \forall k \in \mathcal{M}, \tag{2.16}$$

$$y_{lijk} \leq x_{lik}, \; \forall i, j \in \mathcal{N}, \; \forall l \in \mathcal{N}_0, \; \forall k \in \mathcal{M}, \tag{2.17}$$

$$y_{lijk} \leq z_{ljk}, \; \forall i, j \in \mathcal{N}, \; \forall l \in \mathcal{N}_0, \; \forall k \in \mathcal{M}, \tag{2.18}$$

$$y_{lijk} \geq x_{lik} + z_{ljk} - 1, \; \forall i, j \in \mathcal{N}, \; \forall l \in \mathcal{N}_0, \; \forall k \in \mathcal{M}. \tag{2.19}$$

Note that if $x_{lik} = 1$ and $z_{ljk} = 1$, then constraints (2.19) imply that variable y_{lijk} is equal to 1. Otherwise, variable y_{lijk} is equal to 0. Thus, by replacing constraints (2.5) with the set of constraints (2.16)-(2.19), we obtain a MIP formulation based on decision variables that describe the job precedences.

One clear disadvantage of the proposed linearization would be the addition of a large number of binary variables. Fortunately, because of constraints (2.19), the set of constraints (2.16)-(2.19) is satisfied for the binary variables y_{lijk} if this set is also satisfied for the continuous variables y_{lijk} in the interval $[0,1]$. Therefore, the same number of binary variables in the non-linear formulation is maintained in the MIP model, although there is an increase in the number of continuous variables.

2.2 Deterministic position-based formulation

We can also model the UPMSP variant addressed in this paper as a MIP model using decision variables that describe the position of the jobs in the production sequence of the machines. The following formulation assumes that each machine k has a vector of fixed size, hereafter referred to as *permutation*, that assigns jobs to positions of the production sequence of the machine. Let $\mathcal{P}$ be the set of positions in the permutation of a machine. In addition to variables C_{max} and C_{ik} defined in Section 2.1, we further define the following decision variables:

x_{hik} 1, if job $i \in \mathcal{N}$ is assigned to position $h \in \mathcal{P}$ of the permutation used in machine $k \in \mathcal{M}$; 0, otherwise;

Θ_{hk} completion time of the job assigned to position $h \in \mathcal{P}$ of the permutation used in machine $k \in \mathcal{M}$.

The position-based formulation is posed as follows:

Minimize
$$C_{max} \tag{2.20}$$
Subject to:
$$\sum_{h \in \mathcal{P}} \sum_{k \in \mathcal{M}} x_{hjk} = 1, \forall j \in \mathcal{N}, \tag{2.21}$$
$$\sum_{h \in \mathcal{P}} \sum_{k \in \mathcal{M}} x_{h0k} = |\mathcal{M}|, \tag{2.22}$$
$$\sum_{j \in \mathcal{N}} x_{hjk} \leq 1, \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \tag{2.23}$$
$$\Theta_{hk} \geq \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{j \in \mathcal{N}} \bar{p}_{jk} x_{mjk} + \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \bar{s}_{ijk} x_{(m-1)ik} x_{mjk}, \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \tag{2.24}$$
$$C_{jk} \geq \Theta_{hk} - V(1 - x_{hjk}), \forall h \in \mathcal{P}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.25}$$
$$C_{jk} \leq b_j, \forall j \in \mathcal{B}, \forall k \in \mathcal{M}, \tag{2.26}$$
$$\sum_{i \in \mathcal{N}_0} x_{h+1,ik} \leq \sum_{j \in \mathcal{N}_0} x_{hjk}, \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \tag{2.27}$$
$$x_{0jk} = 0, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.28}$$
$$x_{00k} = 1, \forall k \in \mathcal{M}, \tag{2.29}$$
$$\Theta_{0k} = 0, \forall k \in \mathcal{M}, \tag{2.30}$$
$$C_{0k} = 0, \forall k \in \mathcal{M}, \tag{2.31}$$
$$C_{max} \geq C_{ik}, \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.32}$$
$$C_{ik}, \Theta_{ik} \geq 0, \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.33}$$
$$x_{hjk} \in \{0,1\}, \forall h \in \mathcal{P}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}. \tag{2.34}$$

The objective function (2.20) consists of minimizing the production makespan C_{max}. Constraints (2.21) ensure that each job j is assigned to a unique position of exactly one machine.

Constraints (2.22) guarantee the assignment of the dummy job 0 to a single position of the permutations used in the machines. Constraints (2.23) ensure that at most one job can be assigned to position h of the permutation used in machine k, hence avoiding the overlapping of jobs. Constraints (2.24) determine the completion time of the job assigned to position h of the permutation used in machine k. The first term on the right-hand side of these constraints accumulates the processing times of all jobs assigned to the first h positions of the permutation used in machine k. The second term accumulates the setup times related to the jobs in these first h positions. Notice that we use the product $x_{(m-1)ik}x_{mjk}$ to verify the immediate precedence of jobs. Given two jobs i and j, we have that i immediately precedes j only if $x_{(m-1)ik}x_{mjk} = 1$ for a given $m \leq h$.

Constraints (2.25) determine the completion time of job j in the permutation of machine k. Note that if job j is assigned to position h of the permutation of machine k, then the second term on the right-hand side of the constraints is equal to zero and thus the completion time of job j becomes the completion time of the job assigned to this position. Constraints (2.26) ensure that the due dates of the jobs from the primary customers are met. Constraints (2.27) allow the assignment of a job in position $h+1$ of the permutation of machine k only if there is a job assigned to position h of this permutation. They are used to break symmetry in the solution space. Constraints (2.28) avoid assigning a job at position 0 of the permutations as this position is uniquely used by the dummy job according to constraint (2.29). Constraints (2.30) and (2.31) set initial values to variables Θ_{0k} and C_{0k}, respectively. Finally, constraints (2.32) impose lower bounds to the makespan and constraints (2.33) and (2.34) define the type and domain of the decision variables.

Constraints (2.24) can be linearized similarly to constraints (2.5). Let y_{hijk} be the binary variable that assumes the value of 1 if job i is assigned to position $h-1$ and job j is assigned to position h of the permutation of machine k. The following set of constraints describes the linearization of constraints (2.24):

$$\Theta_{hk} \geq \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \sum_{j\in\mathcal{N}} \bar{p}_{jk}x_{mjk} + \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \sum_{i\in\mathcal{N}_0} \sum_{j\in\mathcal{N}} \bar{s}_{ijk}y_{mijk}, \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \tag{2.35}$$

$$y_{hijk} \leq x_{(h-1)ik}, \forall h \in \mathcal{P}, \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.36}$$

$$y_{lijk} \leq x_{ljk}, \forall h \in \mathcal{P}, \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{2.37}$$

$$y_{hijk} \geq x_{(h-1)ik} + x_{hjk} - 1, \forall l \in \mathcal{P}, \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}. \tag{2.38}$$

Constraints (2.38) ensure that variable y^k_{hij} is equal to 1 if both variables $x_{(h-1)ik}$ and x_{hjk} are also equal to 1. From these constraints, we observe that the set of constraints (2.35)-(2.38) is satisfied for the binary variables y^k_{hij} if this set is also satisfied for the continuous variables y^k_{hij} in the interval $[0, 1]$. Hence, we obtain a MIP formulation for the problem by replacing constraints (2.24) with the set of constraints (2.35)-(2.38).

3 Robust optimization approaches

This section presents RO models for the UPMSP with uncertain processing and setup times based on the nominal MIP models of Section 2. Before developing the models, a brief review of the RO approach is presented.

3.1 Robust optimization: Preliminaries

RO is a mathematical programming based methodology to model and solve optimization problems under uncertain parameters (Ben-Tal et al., 2009; Bertsimas et al., 2015). Different from the stochastic programming methodology, RO does not require a complete knowledge of the probability distributions of the random parameters (Sniedovich, 2012). If they are known, they can be used to build the so-called uncertainty set (Ben-Tal et al., 2009). This set contains all possible realizations of the uncertain parameters and it can be represented by a box, a polyhedron, an ellipsoid or intersections of these sets (Gorissen et al., 2015). RO is a methodology based on worst-case analysis and involves determining here-and-now decisions that optimize some criteria and are insensitive to the possible variations of the uncertain parameters (Bertsimas and Sim, 2004; Ben-Tal et al., 2009; Sniedovich, 2012). The decisions prescribed by the RO methodology ensure 100% of solution feasibility if the uncertain parameter values belong to the uncertainty set, i.e., the solution is deterministically feasible. Even if the uncertainty set does not contain all possible realizations of the uncertain parameters, the RO methodology generally ensures solution feasibility with high probability (Bertsimas and Sim, 2004).

To facilitate the presentation of the OR models proposed in the next subsections, consider the following linear programming model with n decision variables and m constraints, for any given positive integers m and n:

Minimize

$$\sum_{j=1}^{n} c_j x_j \tag{3.1}$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \forall i = 1, \dots, m, \tag{3.2}$$

$$x_j \in \{0, 1\}, \forall j = 1, \dots, n, \tag{3.3}$$

where x_j is a binary decision variable and $c_j, b_i, a_{ij} \in \mathbb{R}$ are input parameters, for all $i = 1, \dots, m$ and $j = 1, \dots, n$. We initially assume that all these parameters are known in advance. Hence, model (3.1)-(3.3) represents a deterministic (nominal) situation and can be used to determine here-and-now minimum cost solutions that are feasible on average when applied in practice.

When some or all of the input parameter values of a problem are unknown at the time of the planning process, it may be relevant to consider the parameter uncertainties for determining the solutions of this problem. Assume that some coefficients in constraints (3.2) are subject to uncertainty. Let J_i be the set of indices of uncertain coefficients of constraint i, for all $i = 1, \dots, m$. We denote by $\tilde{a}_{ij}$, $j \in J_i$ the uncertain coefficient in row i, which is modeled as an independent random variable, limited and symmetric, assuming values in the interval $[\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$, where $\bar{a}_{ij}$ and $\hat{a}_{ij}$ represent the expected (nominal) value and the maximum deviation (from the nominal value), respectively. For each $\tilde{a}_{ij}$, we associate another random variable ξ_j (called the primitive random variable) that assumes values in the interval $[-1, 1]$, such that $\tilde{a}_{ij} = \bar{a}_{ij} + \hat{a}_{ij}\xi_j$. Thus, we build the uncertainty set based on the primitive uncertain parameter ξ_j, leading to the following cardinality-constrained set Bertsimas and Sim (2004):

$$\mathcal{U}_i(\Gamma_i) = \left\{ \boldsymbol{\xi} \in \mathbb{R}^{|J_i|} \mid -1 \leq \xi_j \leq 1, \forall j \in J_i, \sum_{j \in J_i} \xi_j \leq \Gamma_i \right\}, \tag{3.4}$$

for each constraint $i = 1, \ldots, m$. The parameter Γ_i is known as the *budget of uncertainty* and defines the number of primitive uncertain parameters ξ_j allowed to vary in constraint i. If $\Gamma_i = 0$, then the solution of the nominal model corresponds to a highly unprotected solution against uncertainties. It has high chances of being infeasible in practice, in spite of having the best possible objective function value. On the other hand, if $\Gamma_i = |J_i|$, then all uncertain parameters ξ_j are allowed to vary. This situation is very conservative, corresponds to the Soyster method (Soyster, 1973) and its solution ensures full protection against uncertainties, although it has the worst possible objective function value with respect to other choices for Γ_i. By choosing values between 0 and $|J_i|$, it is possible to select values for the *budget of uncertainty* that appropriately considers the trade-off between solution performance and risk. By solution risk, we mean the probability of the robust model solution being infeasible.

Using the uncertainty set (3.4), we can define the robust counterpart of the nominal model (3.1)-(3.3) as:

Minimize

$$\sum_{j=1}^{n} c_j x_j \tag{3.5}$$

Subject to:

$$\sum_{j=1}^{n} \bar{a}_{ij} x_j + \sum_{j \in J_i} \hat{a}_{ij} \xi_j x_j \leq b_i, \forall i = 1, \ldots, m, \quad \forall \boldsymbol{\xi} \in \mathcal{U}_i(\Gamma_i), \tag{3.6}$$

$$x_j \in \{0, 1\}, \forall j = 1, \ldots, n. \tag{3.7}$$

Given that RO is a methodology that searches for the best here-and-now solution $\boldsymbol{x}$ considering all possible realizations of the primitive uncertain parameter $\boldsymbol{\xi}$, the model (3.5)-(3.7) can be rewritten as:

Minimize

$$\sum_{j=1}^{n} c_j x_j \tag{3.8}$$

Subject to:

$$\sum_{j=1}^{n} \bar{a}_{ij} x_j + \max_{\boldsymbol{\xi} \in \mathcal{U}_i(\Gamma_i)} \left\{ \sum_{j \in J_i} \hat{a}_{ij} \xi_j x_j \right\} \leq b_i, \forall i = 1, \ldots, m, \tag{3.9}$$

$$x_j \in \{0, 1\}, \forall j = 1, \ldots, n. \tag{3.10}$$

Let the inner maximization on the left-hand side of constraints (3.9) be defined as the function $\mathcal{B}_i(\boldsymbol{x}, \Gamma_i)$, known as the protection function in the RO literature. Note that the higher this function value, the higher the solution protection against the uncertainties in the coefficients of constraint i. Model (3.8)-(3.10) cannot be directly solved by standard integer linear programming methods as it involves an inner maximization problem. A strategy to remove this inner optimization from the model is to apply duality theory. For a given solution $\boldsymbol{x}^\star$, the protection function $\mathcal{B}_i(\boldsymbol{x}^\star, \Gamma_i)$ of constraint i is equivalent to the following linear optimization model:

$$\max_{\boldsymbol{\xi} \geq 0} \left\{ \sum_{j \in J_i} \hat{a}_{ij} \xi_j x_j^\star \mid \xi_j \leq 1, \forall j \in J_i, \sum_{j \in J_i} \xi_j \leq \Gamma_i \right\}. \tag{3.11}$$

Considering α_{ij}, $j \in J_i$, and β_i as the dual variables of the constraints of the optimization model (3.11), the solution to this model can be obtained solving its corresponding dual problem, given by:

$$\min_{\alpha_{ij}, \beta_i \geq 0} \left\{ \sum_{j \in J_i} \alpha_{ji} + \Gamma_i \beta_i \mid \alpha_{ij} + \beta_i \geq \hat{a}_{ij} x_j^\star, \ \forall j \in J_i \right\}. \tag{3.12}$$

This development is possible because the cardinality constrained set is a convex, limited and closed set. The optimization model (3.11) is known as the primal protection function, while (3.12) is known as the dual protection function.

Substituting (3.12) in model (3.8)-(3.10), we obtain an integer linear programming problem that is equivalent to the uncertain model (3.5)-(3.7), given by:

Minimize

$$\sum_{j=1}^{n} c_j x_j \tag{3.13}$$

Subject to:

$$\sum_{j=1}^{n} \bar{a}_{ij} x_j + \sum_{j \in J_i} \alpha_{ji} + \Gamma_i \beta_i \leq b_i, \forall i = 1, \ldots, m, \tag{3.14}$$

$$\alpha_{ij} + \beta_i \geq \hat{a}_{ij} x_j, \forall i = 1, \ldots, m, \ \ \forall j = 1, \ldots, n, \tag{3.15}$$

$$\alpha_{ij}, \beta_i \geq 0, \forall i = 1, \ldots, m, \ \ \forall j = 1, \ldots, n. \tag{3.16}$$

$$x_j \in \{0, 1\}, \forall j = 1, \ldots, n, \tag{3.17}$$

The decision variables α_{ij} and β_i define the levels of individual and global protections against the uncertainties of coefficients in constraint i, respectively. Notice that there is no need for explicitly adding the minimization of (3.12) in model (3.13)-(3.16), as we are concerned with guaranteeing the feasibility of constraints (3.14). Indeed, if there is a solution that satisfies (3.14), then the minimum of (3.12) also satisfies these constraints.

3.2 Uncertain parameters and uncertainty set

The uncertain parameters, processing time $\tilde{p}_{ik}$ and setup time $\tilde{s}_{ijk}$ are modeled as independent, limited and symmetric random variables that assume values in the intervals $[\bar{p}_{ik} - \hat{p}_{ik}, \bar{p}_{ik} + \hat{p}_{ik}]$ and $[\bar{s}_{ijk} - \hat{s}_{ijk}, \bar{s}_{ijk} + \hat{s}_{ijk}]$, respectively, where $\bar{p}_{ik}$ and $\bar{s}_{ijk}$ represent the expected (nominal) values, and $\hat{p}_{ik}$ and $\hat{s}_{ijk}$ represent the maximum deviations (from their nominal values) allowed for the possible realizations of the random variables. Following the developments presented in Section 3.1, we define the primitive random variables ξ_{ik} and η_{ijk}, which assume values in the interval $[-1, 1]$, and rewrite the original random variables as $\tilde{p}_{ik} = \bar{p}_{ik} + \hat{p}_{ik}\xi_{ik}$ and $\tilde{s}_{ijk} = \bar{s}_{ijk} + \hat{s}_{ijk}\eta_{ijk}$. For the sake of simplicity, we assume that the processing and setup times of all jobs in all machines are uncertain.

We restrict the possible realizations of primitive uncertainty parameters ξ_{ik} and η_{ijk} to the cardinality-constrained set (3.4). However, we define an uncertainty set for each machine k. Thus, the risk is distributed among the production schedules of each machine and, consequently, the highly unlikely scenario in which all the uncertainties of the processing and setup times are concentrated in a single machine is excluded. Thus, Γ_k defines the number of uncertain parameters allowed to vary from their respective nominal values in machine k. The uncertainty sets is defined as follows, for each machine k:

$$\mathcal{U}_k(\Gamma_k) = \Bigg\{ (\boldsymbol{\xi}, \boldsymbol{\eta}) \in \mathbb{R}^{|\mathcal{N}|+|\mathcal{N}_0|\times|\mathcal{N}|} : \\ -1 \leq \xi_{ik} \leq 1,\, \forall i \in \mathcal{N},\, -1 \leq \eta_{ijk} \leq 1,\, \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N},\, \sum_{i \in \mathcal{N}} \xi_{ik} + \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \eta_{ijk} \leq \Gamma_k \Bigg\}. \tag{3.18}$$

Notice that as the uncertainty set depends on the *budget of uncertainty* Γ_k, the larger the number of uncertain parameters ξ_{ik} and η_{ijk} allowed to vary, the higher the conservatism of the robust approach. Note also that the nominal and Soyster problems result from assigning values to the *budget of uncertainty*, Γ_k, for each $k \in \mathcal{M}$, equal to 0 and $|\mathcal{N}|+|\mathcal{N}_0|\times|\mathcal{N}|$, respectively.

3.3 Robust precedence-based formulation

We now develop a RO model for the addressed UPMSP under uncertainty, based on the nominal MIP model defined in Section 2.1 and using the uncertainty set define in equation (3.18). Using the developments presented in Section 3.1, the robust counterpart of the constraints (2.16) can be written as follows:

$$\sum_{i \in \mathcal{N}} \Bigg[\bar{p}_{ik} z_{ijk} + \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \bar{s}_{lik} y_{lijk} \Bigg] + \mathcal{B}_k^j(\boldsymbol{z}, \boldsymbol{y}, \Gamma_k) \leq C_{jk},\, \forall j \in \mathcal{N},\, \forall k \in \mathcal{M}. \tag{3.19}$$

where the $\mathcal{B}_k^j(\boldsymbol{z}, \boldsymbol{y}, \Gamma_k)$ define the primal protection function. For a given solution $(\boldsymbol{z}^\star, \boldsymbol{y}^\star)$, the protection function $\mathcal{B}_k^j(\boldsymbol{z}^\star, \boldsymbol{y}^\star, \Gamma_k)$ of constraint (j, k) is equivalent to the following linear optimization model:

$$\max_{\boldsymbol{\xi} \geq 0, \boldsymbol{\eta} \geq 0} \Bigg\{ \sum_{i \in \mathcal{N}} \hat{p}_{ik} z^\star_{ijk} \xi_{ik} + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \hat{s}_{lik} y^\star_{lijk} \eta_{lik} : \xi_{ik} \leq 1, \forall i \in \mathcal{N}, \eta_{lik} \leq 1, \forall l \in \mathcal{N}_0,\, \forall i \in \mathcal{N}, \sum_{i \in \mathcal{N}} \xi_{ik} \\ + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \eta_{lik} \leq \Gamma_k \Bigg\}. \tag{3.20}$$

By duality, we can rewrite (3.20) as

$$\min_{\boldsymbol{\alpha} \geq 0, \boldsymbol{\beta} \geq 0, \boldsymbol{\lambda} \geq 0} \left\{ \sum_{i \in \mathcal{N}} \alpha_{ijk} + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \beta_{lijk} + \Gamma_k \lambda_{jk} : \alpha_{ijk} + \lambda_{jk} \geq \hat{p}_{ik} z^{\star}_{ijk},\ \forall i \in \mathcal{N}, \beta_{lijk} + \lambda_{jk} \geq \hat{s}_{lik} y^{\star}_{lijk},\ l \in \mathcal{N}_0,\ l \neq j,\ i \in \mathcal{N} \right\}, \tag{3.21}$$

where the decision variables α_{ijk}, β_{lijk} and λ_{jk} are the dual variables associated with the primal optimization problem (3.20). Replacing constraints (2.16) with (3.19) in the MIP model of Section 2.1, we obtain the following RO model for the addressed UPMSP under uncertainty:

Minimize

$$C_{max}. \tag{3.22}$$

Subject to:

$$\text{constraints: (2.2)-(2.4), (2.6)-(2.15), (2.17)-(2.19)}. \tag{3.23}$$

$$\sum_{i \in \mathcal{N}} \left[\bar{p}_{ik} z_{ijk} + \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \bar{s}_{lik} y_{lijk} \right] + \sum_{i \in \mathcal{N}} \alpha_{ijk} + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \beta_{lijk} + \Gamma_k \lambda_{jk} \leq C_{jk},\ \forall j \in \mathcal{N},\ \forall k \in \mathcal{M}, \tag{3.24}$$

$$\alpha_{ijk} + \lambda_{jk} \geq \hat{p}_{ik} z_{ijk},\ \forall i, j \in \mathcal{N},\ \forall k \in \mathcal{M}, \tag{3.25}$$

$$\beta_{lijk} + \lambda_{jk} \geq \hat{s}_{lik} y_{lijk},\ \forall l \in \mathcal{N}_0,\ \forall i, j \in \mathcal{N}, l \neq j,\ \forall k \in \mathcal{M}, \tag{3.26}$$

$$\lambda_{jk} \geq 0,\ \forall j \in \mathcal{N},\ \forall k \in \mathcal{M}, \tag{3.27}$$

$$\alpha_{ijk} \geq 0,\ \forall i, j \in \mathcal{N},\ \forall k \in \mathcal{M}, \tag{3.28}$$

$$\beta_{lijk} \geq 0,\ \forall l \in \mathcal{N}_0,\ \forall i, j \in \mathcal{N},\ l \neq j,\ \forall k \in \mathcal{M}. \tag{3.29}$$

The objective function (3.22) consists of minimizing the makespan. Constraints (3.23) have already been explained and correspond to the constraints of the nominal MIP model. The set of constraints (3.24)-(3.29) is obtained by using the cardinality-constrained set (3.18) and applying the concepts of Section 3.1 to constraints (2.16). As mentioned in Section 3.1, there is no need for explicitly adding the minimization of (3.21) in the model (3.22)-(3.29).

3.4 Robust position-based formulation

We can develop another RO model for the addressed UPMSP variant, associated to the nominal MIP model of Section 2.2 based on decision variables that describe the position of the jobs in the production sequence of each machine. To formulate this robust model, we replace the nominal parameters in constraints (2.35) by their respective random variables, similarly to the development presented in the previous subsection. Using the uncertainty set (3.18) and the concepts of Section 3.1, the robust counterpart of the constraints (2.35) can be stated as:

$$\Theta_{hk} \geq \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{j \in \mathcal{N}} \bar{p}_{jk} x_{mjk} + \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \bar{s}_{ijk} y_{mijk} + \mathcal{B}^h_k(\boldsymbol{x}, \boldsymbol{y}, \Gamma_k), \forall k \in \mathcal{M},\ \forall h \in \mathcal{P}, \tag{3.30}$$

where the primal protection function for a given solution $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$, $\mathcal{B}_k^h(\boldsymbol{x}^\star, \boldsymbol{y}^\star, \Gamma_k)$, is given by the following linear optimization model:

$$\max_{\boldsymbol{\xi} \geq 0, \boldsymbol{\eta} \geq 0} \left\{ \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \sum_{j\in\mathcal{N}} \hat{p}_{jk} x^\star_{mjk} \xi_{jk} \right.$$
$$\left. + \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \sum_{i\in\mathcal{N}_0} \sum_{j\in\mathcal{N}} \hat{s}_{ijk} y^\star_{mijk} \eta_{ijk} : \xi_{jk} \leq 1,\ \forall j\in\mathcal{N}; \eta_{ijk} \leq 1, \forall i\in\mathcal{N}_0,\ \forall j\in\mathcal{N}; \sum_{j\in\mathcal{N}} \xi_{jk} + \sum_{i\in\mathcal{N}_0} \sum_{j\in\mathcal{N}} \eta_{ijk} \leq \Gamma_k \right\}. \tag{3.31}$$

Applying the duality technique to primal protection function (3.31), we can rewrite it as:

$$\min_{\boldsymbol{\gamma} \geq 0, \boldsymbol{\vartheta} \geq 0, \boldsymbol{\mu} \geq 0} \left\{ \sum_{j\in\mathcal{N}} \gamma_{jhk} + \sum_{i\in\mathcal{N}_0} \sum_{j\in\mathcal{N}} \vartheta_{ijhk} \right.$$
$$\left. + \Gamma_k \mu_{hk} : \gamma_{jhk} + \mu_{hk} \geq \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \hat{p}_{jk} x^\star_{mjk},\ \forall j\in\mathcal{N}, \vartheta^h_{ijk} + \mu^h_k \geq \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \hat{s}_{ijk} y^\star_{mijk},\ \forall i\in\mathcal{N}_0,\ \forall j\in\mathcal{N} \right\}. \tag{3.32}$$

where the decision variables γ_{jhk}, ϑ_{ijhk} and μ_{hk} are the dual variables associated with the constraints of the inner optimization problem of primal protection function (3.31). Therefore, the RO model for the addressed UPMSP under uncertainty, associated to the nominal position-based MIP model of Section 2.2 can be stated as:

Minimize
$$C_{max}. \tag{3.33}$$
subject to:
$$\text{constraints: (2.21)-(2.23), (2.25)-(2.34), (2.36)-(2.38)}. \tag{3.34}$$

$$\Theta_{hk} \geq \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \sum_{j\in\mathcal{N}} \bar{p}_{jk} x_{mjk} + \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \sum_{i\in\mathcal{N}_0} \sum_{j\in\mathcal{N}} \bar{s}_{ijk} y_{mijk} + \sum_{j\in\mathcal{N}} \gamma^h_{jk} + \sum_{i\in\mathcal{N}_0} \sum_{j\in\mathcal{N}} \vartheta^h_{ijk} + \Gamma_k \mu^h_k, \forall k\in\mathcal{M},\ \forall h\in\mathcal{P}, \tag{3.35}$$

$$\gamma_{jhk} + \mu_{hk} \geq \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \hat{p}_{jk} x_{mjk},\ \forall j\in\mathcal{N},\ \forall h\in\mathcal{P},\ \forall k\in\mathcal{M}, \tag{3.36}$$

$$\vartheta_{ijhk} + \mu_{hk} \geq \sum_{\substack{m\in\mathcal{P}:\\ m\leq h}} \hat{s}_{ijk} y_{mijk},\ \forall i\in\mathcal{N}_0,\ \forall j\in\mathcal{N},\ \forall h\in\mathcal{P},\ \forall k\in\mathcal{M}, \tag{3.37}$$

$$\gamma_{jhk} \geq 0,\ \forall j\in\mathcal{N},\ \forall k\in\mathcal{M},\ \forall h\in\mathcal{P}, \tag{3.38}$$
$$\vartheta_{ijhk} \geq 0,\ \forall i\in\mathcal{N}_0,\ \forall j\in\mathcal{N},\ \forall k\in\mathcal{M},\ \forall h\in\mathcal{P}. \tag{3.39}$$
$$\mu_{hk} \geq 0,\ \forall k\in\mathcal{M},\ \forall h\in\mathcal{P}, \tag{3.40}$$

This RO model and the one presented in the previous subsection (precedence-based and position-based formulations) can be easily modified to deal with other criteria, such as minimizing total tardiness or the maximum tardiness. This is done by simply adding auxiliary (slack) variables to constraints (2.6) and (2.26) of the nominal models and by appropriately changing the model objective functions.

4 Computational experiments

In this section, we report the computational experiments performed with the proposed RO models to show the impact of the parameter uncertainties in the scheduling decisions and to compare the performance of the proposed RO approach. The models were implemented in C++ programming language and solved using the general-purpose optimization software IBM CPLEX version 12.7, with its default configuration. A Linux PC with a CPU Intel Core i7 3.4 GHz and 16.0 GB of memory was used to run the experiments. The stopping criterion was due to either the elapsed time exceeding the time limit of 3,600 seconds or the optimality gap becoming smaller than 10^{-4}.

We used four sets of instances, namely A, B1, B2 and B3, which are based on other problem instances from the literature. The first set (A) is based on the instances used in Chen (2009); Lin et al. (2011) for the deterministic UPMSP with sequence- and machine-dependent setup times and due date constraints. The instances in this set were generated by combining different factors, such as the number of jobs ($|\mathcal{N}|$), number of machines ($|\mathcal{M}|$), number of families, due date priority factor (τ), due date range factor (R) and proportion of jobs from primary customers. We selected 64 instances from Chen (2009); Lin et al. (2011) that combine different levels of the factors (two levels for each of the six factors). However, we did not consider all the jobs and machines of the original instances; instead, we selected a maximum of 4 machines and 10 jobs to efficiently solve the problem within a computational time limit of 3,600 seconds.

Sets B1, B2 and B3 are based on instances available on the webpage of the Scheduling Research Virtual Center (`http://schedulingresearch.com/`) for the deterministic UPMSP with sequence- and machine-dependent setup times (Arnaout et al., 2010, 2014), but without considering due date constraints. Therefore, for these instances, we generated the due dates of the jobs as in Chen (2009); Lin et al. (2011) using an uniform distribution in the range $[C_m \cdot (1-\tau-R/2), C_m \cdot (1-\tau+R/2)]$, in which $\tau \in [0.4, 0.8]$, $R \in [0.4, 1]$ and C_m is calculated as follows:

$$C_m = (\sum_{j\in\mathcal{N}} \sum_{k\in\mathcal{M}} (p_{jk} + (\sum_{i\in\mathcal{N}} s_{ijk})/|\mathcal{N}|)/|\mathcal{M}|)/|\mathcal{N}|. \quad (4.1)$$

Set B1 considers instances in which the processing times dominate the setup times. Set B2 considers instances in which the setup times dominate the processing times. Finally, set B3 considers instances with balanced processing and setup times. We selected five instances from Arnaout et al. (2010, 2014) for each pair job-machine (2-8, 2-8, 4-8, 4-10), totalizing 20 instances in each of these sets.

Table 1 summarizes the sets of instances. Column *Modification* describes the changes carried out in the original instances. The processing and setup times of these instances are used as nominal values of the corresponding parameters in the definition of the uncertainty set. To incorporate uncertainty, we defined processing and setup time deviations corresponding to 30%, 40% and 50% of the nominal values. The

4.1 Impact of uncertainties

To illustrate the impact of the uncertainties on the problem solutions, we describe the results of an instance (arbitrarily chosen) from set B3, which has balanced processing and setup times. The instance has 2 machines and 8 jobs, in which 3 of these jobs are from primary customers and thus

Tab. 1: Characteristics of the sets of instances.

Class	$\|\mathcal{M}\|$	$\|\mathcal{N}\|$	Instances	Characteristics	Source	Modification
A	2, 4	8, 10	64	Six factors	Chen (2009)	Maximum of 4 machines and 10 jobs
B1	2, 4	8, 10	20	Processing times dominate setup times	Arnaout et al. (2010)	Due dates generated as in Chen (2009)
B2	2, 4	8, 10	20	Setup times dominate processing times	Arnaout et al. (2010)	Due dates generated as in Chen (2009)
B3	2, 4	8, 10	20	Balanced processing and setup times	Arnaout et al. (2010)	Due dates generated as in Chen (2009)
Total			124			

their due dates cannot be violated. The third column of Table 2 shows these due dates (jobs 4, 5 and 7).

In this experiment, we used processing and setup time deviations corresponding to 50%, and $\Gamma = 0, 1, 2, 3$. We first solved the proposed RO models to obtain robust solutions for each different value of Γ. Then, we analyzed the impact of considering a different number of uncertain parameters assuming their worst-case values, namely $\Gamma' = 0, 1, 2, 3$, for each obtained solution with a value of Γ. For instance, for the robust solution obtained with $\Gamma = 2$, we verify the performance of this solution when $\Gamma' = 0, 1, 2, 3$ uncertain parameters attain their worst-case deviations.

Table 2 shows the earliest completion times of jobs 4, 5 and 7 from primary customers, and the makespan C_{max}, for each of the four values of $\Gamma' = 0, 1, 2, 3$ in the each of the four robust solutions obtained by considering $\Gamma = 0, 1, 2, 3$ (last four columns of the table). For example, the value 273 highlighted in the last column of the table corresponds to the earliest completion time of job 4 when $\Gamma' = 0$ parameters attain their worst case value in an optimal solution provided by the RO models with $\Gamma = 3$. If the completion time of a job violates its due date, its value is underlined in the table.

Tab. 2: Impact of uncertainties on the solutions of the illustrative example for different values of Γ.

	Jobs	Due date	Solution of the model with $\Gamma = 0$	$\Gamma = 1$	$\Gamma = 2$	$\Gamma = 3$
Impact of $\Gamma' = 0$	4	398	363	243	250	**273**
	5	275	231	231	139	153
	7	361	114	114	275	114
	C_{max}		497	507	532	566
Impact of $\Gamma' = 1$	4	398	<u>399</u>	281	286	318
	5	275	263	263	182	192
	7	361	143	143	318	143
	C_{max}		533	546	578	609
Impact of $\Gamma' = 2$	4	398	<u>434</u>	312	286	347
	5	275	<u>292</u>	<u>292</u>	208	229
	7	361	171	171	358	171
	C_{max}		568	584	621	652
Impact of $\Gamma' = 3$	4	398	<u>466</u>	339	318	381
	5	275	<u>292</u>	<u>320</u>	208	229
	7	361	171	171	<u>386</u>	171
	C_{max}		600	619	661	693

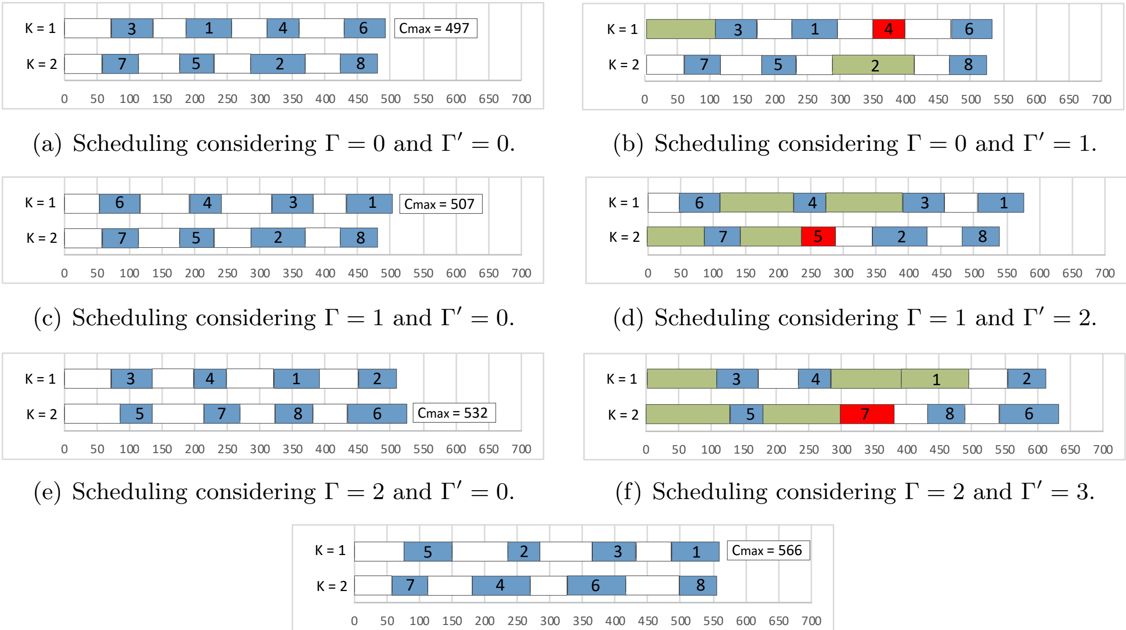
Note that as expected, for $\Gamma' = 0$ (first four lines of the table), all solutions obtained by the RO models for different values of Γ are indeed feasible. However, when $\Gamma' = 1$ (next four lines of the table), the deterministic RO solution obtained for $\Gamma = 0$ becomes infeasible, because the completion time of job 4 (399 in the fourth column of the table) becomes higher than the due date of job 4 (398 in the third column of the table). In fact, we can observe that for $\Gamma' > \Gamma$, the model solutions are infeasible, i.e., a solution obtained by the RO models considering Γ uncertain parameters in their worst case deviations becomes infeasible if the actual number of parameters in their worst case deviations, Γ', is larger than Γ. As a consequence, the deterministic model solution ($\Gamma = 0$) becomes infeasible when at least one uncertain parameter assumes the worst case deviation (i.e., for $\Gamma' > 0$).

Observe in the table that the solution values (C_{max}) increase as the values of Γ and Γ' increase. For example, the makespan of the solution for $\Gamma = 0$ and $\Gamma' = 0$ (497) is 13.8% smaller than the makespan of the solution for $\Gamma = 3$ and $\Gamma' = 0$ (566). Notice that in both cases we have $\Gamma' = 0$, but the solution for $\Gamma = 3$ has a higher makespan because it is obtained considering up to three uncertain parameters assuming their worst case values. The solution value for $\Gamma = 3$ is 13.8% worse to be uncertainty-immunized (protected) against these three parameters, even if the actual values of these parameters may not be at their worst case deviations. Similarly, the solution value for $\Gamma = 3$ and $\Gamma' = 0$ (566) is 22.4% better than the one for $\Gamma = 3$ and $\Gamma' = 3$ (693).

Figure 1 shows the impact of considering $\Gamma' > \Gamma$ in some solutions of Table 2. The white boxes indicate setup times; the blue boxes indicate processing times and the jobs; the green boxes indicate the uncertain parameters actually assuming their worst case values; and the red boxes correspond to the infeasibility of the job due dates. Figure 1(a) shows that the (deterministic) solution obtained by the RO models for $\Gamma = 0$ is feasible for $\Gamma' = 0$. Machine $k = 1$ processes jobs 3, 1, 4 and 6, while machine $k = 2$ processes jobs 7, 5, 2 and 8. However, Figure 1(b) shows that for $\Gamma' = 1$, this deterministic solution becomes infeasible, because the increase in the first setup time of machine $k = 1$ makes the completion time of job 4 higher than its due date. To avoid this infeasibility, a new solution that changes the production sequence of jobs in machine $k = 1$ is provided by the RO models for $\Gamma = 1$ (see Figure 1(c)). However, if the number of uncertain parameters assuming their worst case deviations increases to $\Gamma' = 2$, this solution also becomes infeasible, violating the due date of job 5 (see Figure 1(d)). In this case, to avoid infeasibility, a new solution obtained by the RO models for $\Gamma = 2$ swaps jobs between the machines and changes the production sequence of the jobs in both the machines (see Figure 1(e)). This solution becomes infeasible for $\Gamma' = 3$ as it violates the due date of job 7 (see Figure 1(f)). Therefore, a new swap and reallocation of jobs is done in the new model solution for $\Gamma = 3$, which avoids this infeasibility (see Figure 1(g)). Note that in this solution, jobs 4, 5 and 7 with due date constraints are processed first in the machines, protecting the production scheduling against uncertainties, but at the expense of a higher production makespan. Thus, the increase of Γ values in the RO models changes the schedule configurations to provide more protected solutions, at the cost of higher makespans.

4.2 Trade-off between cost and risk

In this section, we analyze the trade-off between cost and risk of the RO solutions for different problem instances. For this purpose, we determine the *price of robustness* (PoR), the empirical probability of constraint violation (Risk) and the expected cost of the solution (Exp. Cost) (i.e., the expected makespan of the solution). Let $x^{\alpha,\Gamma}$ be the optimal solution for a given deviation α and



(a) Scheduling considering $\Gamma = 0$ and $\Gamma' = 0$.

(b) Scheduling considering $\Gamma = 0$ and $\Gamma' = 1$.

(c) Scheduling considering $\Gamma = 1$ and $\Gamma' = 0$.

(d) Scheduling considering $\Gamma = 1$ and $\Gamma' = 2$.

(e) Scheduling considering $\Gamma = 2$ and $\Gamma' = 0$.

(f) Scheduling considering $\Gamma = 2$ and $\Gamma' = 3$.

(g) Scheduling considering $\Gamma = 3$ and $\Gamma' = 0$.

Fig. 1: Solutions of the illustrative example for different values of Γ and Γ'.

budget of uncertainty Γ. The PoR is evaluated as $\frac{z(x^{\alpha,\Gamma})-z}{z} \cdot 100\%$, in which $z(x^{\alpha,\Gamma})$ is the optimal objective cost (i.e., the optimal makespan) of the robust model and z is the optimal objective function cost of the deterministic model. The Risk is determined via a Monte Carlo simulation that randomly generates a sufficiently large number of realizations for the random variables and tests the feasibility of a fixed (given) solution.

The Monte Carlo simulation was performed by generating random realizations for the processing and setup times in the half-intervals $[\bar{p}_{ik}, \bar{p}_{ik} + \hat{p}_{ik}]$ and $[\bar{s}_{lik}, \bar{s}_{lik} + \hat{s}_{lik}]$, respectively. To perform the simulation, 10,000 different uniformly distributed samples were generated and, for each sample of the simulation, we verified if the solution was infeasible for at least one realization of each sample (out of the 10,000 generated). After performing all simulation iterations, we counted the number of samples for which the solution was infeasible, which was divided by the total number of samples, to obtain the estimated risk. The expected cost (Exp. Cost) of the solutions was calculated as the average cost of these 10,000 generated samples (i.e., the average makespan of these samples).

Table 3 shows the objective function cost (OF), the expected cost of the solution (Exp. Cost), the price of robustness (PoR) and the risk for the different sets of instances with a deviation of 50% (i.e., $\alpha = 0.5$). Experiments using $\alpha = 0.3$ and $\alpha = 0.4$ show similar overall results. Note that, in general, most of the instances present solutions with low risk when more than two uncertain parameters are considered ($\Gamma > 2$) assuming the worst case values. For the solutions in which the risk is equal to zero, the objective function cost increases by less than 50% with respect to the objective function cost of the deterministic problem. For example, the average risk of instances of set B1 with 10 jobs and 2 machines considering that $\Gamma = 0, 1, 2$ uncertain parameters achieve their worst case is 76.16%, 16.09% and 8.43%, respectively. When $\Gamma = 3$, the risk decreases to zero while the cost of the solution increases by 17.68%.

Tab. 3: Objective function cost (OF), expected cost of the solution (Exp. Cost), price of robustness (PoR), and empirical probability of constraints violation (Risk) for the different sets of instances with $\alpha = 0.5$.

			A				B1				B2				B3			
$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{M}\rvert$	Γ	OF	Exp. Cost	PoR	Risk	OF	Exp. Cost	PoR	Risk	OF	Exp. Cost	PoR	Risk	OF	Exp. Cost	PoR	Risk
8	2	0	1,194	1,295	0.00	30.06	518	559	0.00	71.65	823	893	0.00	39.81	828	896	0.00	61.22
8	2	1	1,376	1,336	15.21	21.03	560	567	8.09	2.50	901	906	9.47	0.17	896	907	8.21	1.04
8	2	2	1,499	1,339	25.53	0.00	600	574	15.89	0.00	965	912	17.30	0.00	957	911	15.60	0.00
8	2	3	1,600	1,340	33.93	0.00	637	582	23.07	0.00	1,024	915	24.46	0.00	1,005	906	21.47	0.00
8	2	4	1,664	1,332	39.34	0.00	666	584	28.56	0.00	1,067	911	29.72	0.00	1,056	908	27.59	0.00
8	2	5	1,696	1,332	42.01	0.00	689	584	33.08	0.00	1,097	910	33.33	0.00	1,089	908	31.59	0.00
10	2	0	1,422	1,545	0.00	24.60	647	697	0.00	**76.16**	1,019	1,102	0.00	5.29	1,031	1,117	0.00	65.40
10	2	1	1,612	1,589	13.35	0.04	688	705	6.36	**16.09**	1,083	1,102	6.30	5.20	1,094	1,121	6.12	1.80
10	2	2	1,752	1,600	23.21	0.00	729	713	12.67	**8.43**	1,150	1,110	12.85	0.00	1,160	1,132	12.51	0.00
10	2	3	1,858	1,598	30.62	0.00	761	716	**17.68**	**0.00**	1,211	1,115	18.91	0.00	1,218	1,134	18.14	0.00
10	2	4	1,931	1,592	35.78	0.00	790	716	22.19	0.00	1,267	1,116	24.36	0.00	1,269	1,135	23.17	0.00
10	2	5	2,004	1,593	40.88	0.00	816	716	26.19	0.00	1,320	1,118	29.56	0.00	1,320	1,134	28.06	0.00
8	4	0	416	454	0.00	17.61	274	297	0.00	20.00	418	455	0.00	55.13	423	462	0.00	57.05
8	4	1	514	458	23.59	8.43	309	299	12.85	0.00	479	462	14.58	0.00	485	464	14.73	0.00
8	4	2	581	464	39.69	6.19	338	297	23.27	0.00	536	463	28.20	0.00	540	465	27.86	0.00
8	4	3	592	462	42.26	0.00	361	297	31.58	0.00	566	461	35.47	0.00	569	465	34.75	0.00
8	4	4	591	460	42.03	0.00	380	296	38.80	0.00	592	461	41.56	0.00	594	464	40.56	0.00
8	4	5	591	460	42.03	0.00	380	297	38.80	0.00	592	461	41.56	0.00	594	464	40.56	0.00
10	4	0	489	530	0.00	42.95	362	391	0.00	26.84	581	630	0.00	78.52	577	627	0.00	19.12
10	4	1	588	537	20.24	0.00	396	394	9.32	0.00	640	633	10.07	0.00	634	629	9.72	2.46
10	4	2	669	534	36.94	0.00	420	363	15.92	0.00	694	634	19.51	0.00	688	633	19.11	0.00
10	4	3	690	530	41.29	0.00	444	393	22.76	0.00	747	635	28.54	0.00	739	634	28.07	0.00
10	4	4	696	542	42.47	0.00	465	393	28.50	0.00	772	634	32.94	0.00	775	640	34.20	0.00
10	4	5	696	542	42.47	0.00	485	392	33.87	0.00	795	634	36.82	0.00	787	631	36.30	0.00

The objective function cost and the expected cost of the solution increase for higher values of Γ. However, the increase of the expected cost is slower than the increase of the objective function cost. Thus, for most instances with $\Gamma = 0$, the expected cost of the solutions is higher than the objective function cost. On the other hand, for $\Gamma > 2$, the expected cost of the solutions is smaller than the objective function cost in most of the instances. These results suggest that ignoring the impact of the uncertainties (with $\Gamma = 0$) could provide infeasible solutions in practice violating the due date of the customers (as indicated by the risk) or solutions with a cost higher than desired (as indicated by the expected cost higher than the objective function cost). Figure 2 shows the difference between the objective function cost and the expected cost (OF − Exp. Cost) for instances with 8 jobs and 2 machines. Notice that small values of Γ provide solutions that underestimate the empirical actual cost of the solutions, while high values of Γ provide solutions that overestimate this measure.

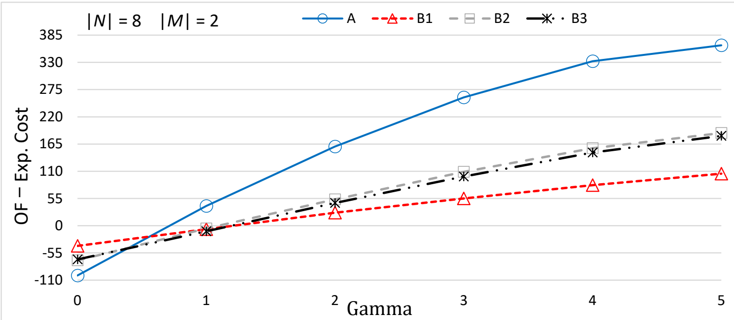


Fig. 2: Difference between the objective function cost and the expected cost of the solutions for instances with 8 jobs and 2 machines.

4.3 Computational performance of the robust models

We now compare the computational performance of the precedence-based robust model and the position-based robust model for different values of Γ and α. Tables 4 and 5 show the average computational results for different values of Γ and α, respectively. The tables present the total number of instances (Inst.) in each group (row of the table), the number of infeasible instances (Inf.), the number of instances solved to optimality (Opt.), the average objective function (Avg. OF), the average gap in percentage (Avg. gap) and the average elapsed time in seconds (Avg. time) for both RO models proposed in this paper. Note that more instances become infeasible for higher values of Γ and α. On the other hand, the number of instances solved to optimality with the RO models and the average gap increase for higher values of Γ and α. For instances with 10 jobs and 4 machines, for example, the number of instances solved to optimality with the precedence-based model is 93 (out of 93) with $\Gamma = 0$ and 29 (out of 93) with $\Gamma = 5$. Similarly, the number of instances with 10 jobs and 4 machines solved to optimality with the precedence-based model is 123 (out of 186) with $\alpha = 0.3$ and 99 (out of 186) with $\alpha = 0.5$. Thus, considering uncertainty in the problem clearly makes it more difficult to solve using the proposed models.

The results indicate that the position-based robust model has a better performance than the precedence-based robust model for the problem instances used in these experiments. The position-based model solved 89.76% of the instances to optimality, while the precedence-based model solved

Tab. 4: Computational results for different values of Γ.

					Precedence-based formulation				Position-based formulation			
$\|\mathcal{N}\|$	$\|\mathcal{M}\|$	Γ	Inst.	Inf.	Opt.	Avg. OF	Avg. gap	Avg. time	Opt.	Avg. OF	Avg. gap	Avg. time
8	2	0	93	0	93	966	0.002	173.00	93	966	0.001	80.52
8	2	1	93	12	81	1,049	0.002	387.12	81	1,048	0.001	106.74
8	2	2	93	14	76	1,131	0.930	664.09	79	1,131	0.000	114.85
8	2	3	93	17	70	1,220	1.509	809.32	76	1,210	0.000	105.72
8	2	4	93	20	65	1,291	2.720	939.15	73	1,280	0.001	98.85
8	2	5	93	20	59	1,321	4.162	1,107.16	73	1,310	0.000	88.68
10	2	0	93	0	93	1,169	0.003	239.29	93	1,169	0.005	81.94
10	2	1	93	12	66	1,253	1.071	1,382.71	81	1,247	0.002	35.69
10	2	2	93	13	27	1,343	6.337	2,835.20	80	1,336	0.003	37.64
10	2	3	93	13	21	1,411	9.648	3,110.61	80	1,411	0.002	38.99
10	2	4	93	13	16	1,469	13.306	3,124.91	80	1,469	0.003	35.94
10	2	5	93	13	9	1,525	17.075	3,295.03	80	1,524	0.003	38.15
8	4	0	93	0	93	394	0.000	103.35	93	394	0.000	82.94
8	4	1	93	0	93	471	0.000	292.39	93	471	0.000	68.09
8	4	2	93	2	91	532	0.001	340.95	91	532	0.000	78.58
8	4	3	93	2	91	551	0.000	328.68	91	551	0.000	87.63
8	4	4	93	3	90	561	0.000	278.51	90	561	0.000	67.29
8	4	5	93	3	90	562	0.000	312.00	90	561	0.000	76.13
10	4	0	93	0	93	497	0.002	219.45	93	497	0.003	46.26
10	4	1	93	3	82	572	1.180	719.11	69	572	5.350	900.47
10	4	2	93	9	49	638	7.983	1,878.48	63	638	5.662	969.56
10	4	3	93	9	39	680	13.962	2,255.01	60	670	7.177	1,093.88
10	4	4	93	10	36	693	13.882	2,358.36	56	686	8.974	1,224.67
10	4	5	93	10	29	704	16.417	2,613.63	56	695	8.298	1,219.63
Avg.			93	8	65	917	4.591	1,240.31	80	914	1.479	282.45

Tab. 5: Computational results for different values of α.

					Precedence-based formulation				Position-based formulation			
$\|\mathcal{N}\|$	$\|\mathcal{M}\|$	α	Inst.	Inf.	Opt.	Avg. OF	Avg. gap	Avg. time	Opt.	Avg. OF	Avg. gap	Avg. time
8	2	0.3	186	20	161	1,101	0.66	575.25	166	1,100	0.00	92.07
8	2	0.4	186	23	156	1,162	1.28	624.82	163	1,151	0.00	101.97
8	2	0.5	186	40	127	1,198	2.60	788.26	146	1,198	0.00	102.93
10	2	0.3	186	20	85	1,323	5.68	2,129.96	166	1,306	0.00	45.61
10	2	0.4	186	20	72	1,364	7.93	2,338.20	166	1,361	0.00	45.72
10	2	0.5	186	24	75	1,397	9.39	2,346.02	162	1,397	0.00	45.72
8	4	0.3	186	0	186	480	0.00	274.72	186	480	0.00	75.84
8	4	0.4	186	0	186	509	0.00	281.73	186	509	0.00	83.15
8	4	0.5	186	10	176	546	0.00	271.49	176	546	0.00	71.06
10	4	0.3	186	5	123	592	5.92	1,441.61	134	591	6.50	988.04
10	4	0.4	186	9	106	627	9.35	1,713.01	135	624	5.76	913.53
10	4	0.5	186	27	99	669	11.17	1,820.78	128	659	5.01	761.13
Avg.				16.5	129.33	914	4.499	1,217.15	159.50	910	1.441	277.23

69.5%, within the limit of 3,600 seconds. Furthermore, the average gap is 3 times higher for the precedence-based model than for the position-based model.

5 Conclusions

We presented a robust optimization approach for the UPMSP with due date constraints and sequence- and machine-dependent setup times under uncertain processing and setup times. We proposed two robust optimization models based on mixed integer programming models of the deterministic problem. The robust prece-dence-based model uses decision variables that describe the precedences of jobs, while the robust position-based model uses decision variables that describe the position of the jobs in the production sequence of each machine. Both models are suitable for incorporating uncertainties in the job processing times and in the machine setup times, using the robust optimization paradigm.

Computational results using instances from the literature and a Monte-Carlo simulation showed that the robust models represent the problem appropriately and lead to effective and robust schedules, when solved by general purpose optimization software. These results also indicated that the models can be useful for analyzing the impact of the uncertainties and the trade-off between robustness and conservativeness of the solutions. It was observed that ignoring the parameter uncertainties can result in infeasible schedules or high cost schedules. Regarding the computational performances of the two models, the robust position-based model was able to solve a larger number of problem instances to optimality than the robust precedence-based model.

An interesting line of future research would be to develop more effective methods for solving these robust MIP models, such as meta-heuristic and exact methods based on decomposition. Particularly, the heuristics and meta-heuristics proposed in the literature to solve the deterministic version of the problem could be extended to solve the robust version. It is worth mentioning that the two proposed robust optimization models can be easily modified to deal with other criteria, such as minimizing total tardiness or the maximum tardiness. Another promising line of research would be to investigate how to adapt the models to consider other uncertain parameters in the UPMSP, such as in the due dates. Future research could also explore other approaches to cope with uncertainties, such as two-stage stochastic programming with recourse, fuzzy programming techniques, adjustable robust optimization, robust optimization with recourse, risk-neutral and risk-averse stochastic programming techniques.

6 Acknowledgments

This work was supported by the Sao Paulo Research Foundation (FAPESP) [grant numbers 15/14582-7, 16/15966-6, 16/23366-9 and 16/01860-1].

References

Afzalirad, M. and Rezaeian, J. (2016). Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Computers & Industrial Engineering*, 98:40–52.

Arnaout, J.-p., Rabadi, G., and Musa, R. (2010). A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, 21(6):693–701.

Arnaout, J.-p. P., Musa, R., Rabadi, G., Rami, J.-p. A., and Ghaith, M. (2014). A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines - Part II: Enhancements and experimentations. *Journal of Intelligent Manufacturing*, 25(1):43–53.

Avalos-Rosales, O., Angel-Bello, F., and Alvarez, A. (2015). Efficient metaheuristic algorithm and reformulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, 76(9):1705–1718.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust Optimization.* Princeton Series in Applied Mathematics. Princeton University Press.

Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming Series B*, 88:411–424.

Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and applications of robust optimization. *Society for Industrial and Applied Mathematics*, 53(3):446–501.

Bertsimas, D., Goyal, V., and Lu, B. (2015). A tight characterization of the performance of static solutions in two-stage adjustable robust linear optimization. *Mathematical Programming*, 150(2):281–319.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.

Chen, J.-F. (2009). Scheduling on unrelated parallel machines with sequence- and machine-dependent setup times and due-date constraints. *The International Journal of Advanced Manufacturing Technology*, 44(11-12):1204–1212.

Chen, J.-F. (2015). Unrelated parallel-machine scheduling to minimize total weighted completion time. *Journal of Intelligent Manufacturing*, 26(6):1099–1112.

Cheng, T. and Sin, C. (1990). A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, 47(3):271–292.

Chyu, C.-C. and Chang, W.-S. (2011). Optimizing fuzzy makespan and tardiness for unrelated parallel machine scheduling with archived metaheuristics. *The International Journal of Advanced Manufacturing Technology*, 57(5-8):763–776.

Feng, X., Zheng, F., and Xu, Y. (2016). Robust scheduling of a two-stage hybrid flow shop with uncertain interval processing times. *International Journal of Production Research*, 54(12):3706–3717.

Gharehgozli, A., Tavakkoli-Moghaddam, R., and Zaerpour, N. (2009). A fuzzy-mixed-integer goal programming model for a parallel-machine scheduling problem with sequence-dependent setup times and release dates. *Robotics and Computer-Integrated Manufacturing*, 25(4-5):853–859.

Gomes, F. R. A. and Mateus, G. R. (2017). Improved Combinatorial Benders Decomposition for a Scheduling Problem with Unrelated Parallel Machines. *Journal of Applied Mathematics*, 2017:1–10.

González-Neira, E. M., Montoya-Torres, J. R., and Barrera, D. (2017). Flow-shop scheduling problem under uncertainties: Review and trends. *International Journal of Industrial Engineering Computations*, 8(4):399–426.

Gorissen, B. L., Yamkoğlu, I., and den Hertog, D. (2015). A practical guide to robust optimization. *Omega*, 53(0):124 – 137.

Hu, H., Ng, K. K. H., and Qin, Y. (2016). Robust parallel machine scheduling problem with uncertainties and sequence-dependent setup time. *Scientific Programming*, page 13.

Liao, T. W. and Su, P. (2017). Parallel machine scheduling in fuzzy environment with hybrid ant colony optimization including a comparison of fuzzy number ranking methods in consideration of spread of fuzziness. *Applied Soft Computing*, 56:65–81.

Lin, S.-W., Lu, C.-C., and Ying, K.-C. (2011). Minimization of total tardiness on unrelated parallel machines with sequence- and machine-dependent setup times under due date constraints. *The International Journal of Advanced Manufacturing Technology*, 53(1-4):353–361.

Macedo, P. B., Alem, D., Santos, M., Junior, M. L., and Moreno, A. (2016). Hybrid manufacturing and remanufacturing lot-sizing problem with stochastic demand, return, and setup costs. *The International Journal of Advanced Manufacturing Technology*, 82(5):1241–1257.

Nogueira, J. P., Arroyo, J. E. C., Villadiego, H. M. M., and Gonçalves, L. B. (2014). Hybrid GRASP heuristics to solve an unrelated parallel machine scheduling problem with earliness and tardiness penalties. *Electronic Notes in Theoretical Computer Science*, 302:53–72.

Ravetti, M. G., Mateus, G. R., Rocha, P. L., and Pardalos, P. M. (2007). A scheduling problem with unrelated parallel machines and sequence dependent setups. *International Journal of Operational Research*, 2(4):380.

Sadati, A., Tavakkoli-moghaddam, R., Naderi, B., and Mohammadi, M. (2017). Solving a new multi-objective unrelated parallel machines scheduling problem by hybrid teaching-learning based optimization. *International Journal of Engineering*, 30(2):224–233.

Salehi Mir, M. S. and Rezaeian, J. (2016). A robust hybrid approach based on particle swarm optimization and genetic algorithm to minimize the total machine load on unrelated parallel machines. *Applied Soft Computing*, 41:488–504.

Seo, K.-K. and Chung, B. D. (2014). Robust optimization for identical parallel machine scheduling with uncertain processing time. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 8(2):1–11.

Sniedovich, M. (2012). Robust optimization - the elephant in the robust-satisficing room. Technical report, University of Melbourne.

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operation Research*, 21(5):1154–1157.

Torabi, S., Sahebjamnia, N., Mansouri, S., and Bajestani, M. A. (2013). A particle swarm optimization for a fuzzy multi-objective unrelated parallel machines scheduling problem. *Applied Soft Computing*, 13(12):4750–4762.

Xu, X., Cui, W., Lin, J., and Qian, Y. (2013). Robust makespan minimisation in identical parallel machine scheduling problem with interval data. *International Journal of Production Research*, 51(12):3532–3548.

Ying, K.-C., Lee, Z.-J., and Lin, S.-W. (2012). Makespan minimization for scheduling unrelated parallel machines with setup times. *Journal of Intelligent Manufacturing*, 23(5):1795–1803.

Zeidi, J. R. and MohammadHosseini, S. (2015). Scheduling unrelated parallel machines with sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, 81(9-12):1487–1496.