



A robust optimization approach for the unrelated parallel machine scheduling problem

Jonathan De La Vega¹ · Alfredo Moreno^{2,3} · Reinaldo Morabito¹  · Pedro Munari¹

Received: 6 March 2020 / Accepted: 15 December 2021 / Published online: 28 January 2022
© The Author(s) under exclusive licence to Sociedad de Estadística e Investigación Operativa 2022

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 · Robust optimization

This work was supported by the Sao Paulo Research Foundation (FAPESP) [grant numbers 15/14582-7, 16/15966-6, 19/23596-2 and 16/01860-1], the National Council for Scientific and Technological Development (CNPq) [grant number 313220/2020-4]; and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES) [Finance code 001].

✉ Reinaldo Morabito
morabito@ufscar.br

¹ Production Engineering Department, Federal University of São Carlos, São Carlos, Brazil

² Facultad de Ingeniería Industrial, Universidad Pontificia Bolivariana, Montería, Colombia

³ Department of Logistics and Operations Management, HEC Montréal, Montreal, Canada

Mathematics Subject Classification 90-08 · 90-10 · 90B36 · 90C11 · 90C17

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 diverse sources of uncertainty such as machine breakdowns, environment changes, worker performance, among other machine and/or human factors (Chyu and Chang 2011; Liao and Su 2017). These uncertainties are natural in production planning and scheduling problems and, if ignored, they can lead to schedules that are infeasible or perform poorly under real conditions (Bougeret et al. 2019). 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 programming, stochastic programming, and robust optimization (Macedo et al. 2016; Rodríguez et al. 2009; Salmasnia et al. 2015; Tadayon and Smith 2015; González-Neira et al. 2017; Bougeret et al. 2019). Some of these studies considered the single machine scheduling problem (Tadayon and Smith 2015; Bougeret et al. 2019), while others addressed the UPMSP but without explicitly considering due-date constraints. 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. More 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 programming 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 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 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 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: (1) it does not require probability distribution information to describe parameter uncertainty. Information of the data probability distribution may be not available in some real contexts of multiple machine scheduling problems. In this case, managers can define realistic intervals for the uncertain parameter values based on their own experience and the available data; (2) the robust counterpart of an uncertain MIP model can be reformulated as a deterministic MIP depending on the uncertainty set; and (3) it provides a feasible solution (uncertainty-immunized solution) for any parameter realization belonging to a predetermined uncertainty set (Seo and Chung 2014). In the UPMSP, RO can provide robust schedules, which are desirable from a practical perspective because they protect against adverse conditions of the system (Bougere et al. 2019). 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 (Bertsimas and Sim 2004; Sungur et al. 2008; Alem and Morabito 2012). Otherwise, if we have only one uncertain parameter by constraint and apply the classical RO approach using the budgeted uncertainty set, the RO formulation would be equivalent to a deterministic formulation where the uncertain data assumes its worst-case value, making the formulation totally conservative. This is because the robust counterpart is obtained dualizing the protection function inserted into each constraint (see Sects. 3.2 and 3.3), and each protection function considers the variation in the uncertain parameters defined in its corresponding constraint only. Hence, to overcome the mentioned difficulty, we first present two deterministic models for the addressed UPMSP variant that are convenient for using RO. 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 Sect. 3. In Sect. 4, we report the results of the computational experiments with the proposed

RO approach. Finally, Sect. 5 presents final remarks and discusses perspectives for future research.

2 Problem description and deterministic formulations

In this section, we describe the UPMSP (Sect. 2.1) and present two deterministic mathematical formulations for the problem. The formulations use decision variables that describe job precedences (Sect. 2.2) and decision variables that describe the position of the jobs in the production sequence of the machines (Sect. 2.3).

2.1 The UPMSP

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 variant of the problem (also known as nominal problem), all job processing and setup times are assumed to be known in advance. As the machines are unrelated, the processing time to perform the operation of a given job depends on the machine. The same happens for the setup time to perform the operation of a given job, which further depends on the production sequence of the machine. 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 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. For example, V can be equal to the sum of all the processing and setup times.

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.

2.2 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 represent 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 defined, the deterministic precedence-based formulation is as follows:

$$\text{Minimize } C_{\max} \tag{1}$$

subject to:

$$\sum_{\substack{i \in \mathcal{N}_0: \\ i \neq j}} \sum_{k \in \mathcal{M}} x_{ijk} = 1, \quad \forall j \in \mathcal{N}, \tag{2}$$

$$\sum_{\substack{i \in \mathcal{N}_0: \\ i \neq h}} x_{ihk} - \sum_{\substack{j \in \mathcal{N}_0: \\ j \neq h}} x_{hjk} = 0, \quad \forall h \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{3}$$

$$\sum_{j \in \mathcal{N}_0} x_{0jk} = 1, \quad \forall k \in \mathcal{M}, \tag{4}$$

$$\sum_{i \in \mathcal{N}} \bar{p}_{ik} z_{ijk} + \sum_{\substack{i \in \mathcal{N}: \\ i \neq j}} \bar{s}_{lik} x_{lik} z_{ljk} \leq C_{jk}, \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{5}$$

$$C_{ik} \leq b_i, \quad \forall i \in \mathcal{B}, \forall k \in \mathcal{M}, \tag{6}$$

$$C_{\max} \geq C_{ik}, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{7}$$

$$R_{jk} \geq R_{ik} + 1 - |\mathcal{M}|(1 - x_{ijk}), \quad \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \tag{8}$$

$$Vz_{ijk} \geq R_{jk} - R_{ik} - V \left[2 - \sum_{l \in \mathcal{N}_0} x_{lik} - \sum_{l \in \mathcal{N}_0} x_{ljk} \right], \quad \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \tag{9}$$

$$2x_{ijk} \leq z_{iik} + z_{ijk}, \quad \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \tag{10}$$

$$\sum_{k \in \mathcal{M}} z_{iik} = 1, \quad \forall i \in \mathcal{N}, \tag{11}$$

$$C_{0k} = 0, \quad \forall k \in \mathcal{M}, \tag{12}$$

$$C_{ik} \geq 0, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{13}$$

$$0 \leq R_{ik} \leq |\mathcal{M}|, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{14}$$

$$x_{ijk}, z_{ijk} \in \{0, 1\}, \quad \forall i, j \in \mathcal{N}_0, \forall k \in \mathcal{M}. \tag{15}$$

The objective function (1) consists of minimizing the total time required to process all jobs (makespan). Constraints (2)–(4) are the flow constraints: (2) ensure that each job j must be processed in exactly one machine; (3) impose that each job h processed in a machine has an immediate predecessor and an immediate successor; and (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 (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 (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) 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 (6) ensure that the due dates of jobs of primary customers are met. The makespan is computed by constraints (7), together with the objective function. Note that C_{\max} corresponds to the completion time of the last processed job. Constraints (8) link the variables R_i and x_{ijk} based on two jobs i and j such that i immediately precedes j . Constraints (9) link 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 (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 (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 (12). Finally, constraints (13), (14) and (15) determine the type and domain of the decision variables.

The formulation (1)–(15) is a non-linear model because of the product of variables in constraints (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 (5):

$$\sum_{i \in \mathcal{N}} \bar{p}_{ik} z_{ijk} + \sum_{i \in \mathcal{N}} \sum_{l \in \mathcal{N}_0} \bar{s}_{lik} y_{lijk} \leq C_{jk}, \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \quad (16)$$

$$y_{lijk} \leq x_{lik}, \quad \forall i, j \in \mathcal{N}, \forall l \in \mathcal{N}_0, \forall k \in \mathcal{M}, \quad (17)$$

$$y_{lijk} \leq z_{ljk}, \quad \forall i, j \in \mathcal{N}, \forall l \in \mathcal{N}_0, \forall k \in \mathcal{M}, \quad (18)$$

$$y_{lijk} \geq x_{lik} + z_{ljk} - 1, \quad \forall i, j \in \mathcal{N}, \forall l \in \mathcal{N}_0, \forall k \in \mathcal{M}. \quad (19)$$

Note that if $x_{lik} = 1$ and $z_{ljk} = 1$, then constraints (19) imply that variable y_{lijk} is equal to 1. Otherwise, variable y_{lijk} is equal to 0. Thus, by replacing constraints (5) with the set of constraints (16)–(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 (19) and variables x_{lik}, z_{ljk} being binary, the set of constraints (16)–(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.

2.3 Deterministic position-based formulation

We can also model the UPMSPP 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 Sect. 2.2, 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:

$$\text{Minimize } C_{\max} \tag{20}$$

subject to:

$$\sum_{h \in \mathcal{P}} \sum_{k \in \mathcal{M}} x_{hjk} = 1, \quad \forall j \in \mathcal{N}, \tag{21}$$

$$\sum_{h \in \mathcal{P}} \sum_{k \in \mathcal{M}} x_{h0k} = |\mathcal{M}|, \tag{22}$$

$$\sum_{j \in \mathcal{N}} x_{hjk} \leq 1, \quad \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \tag{23}$$

$$\Theta_{hk} \geq \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{j \in \mathcal{N}} \bar{p}_{jk} x_{mjk} + \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \bar{s}_{ijk} x_{(m-1)ik} x_{mjk}, \quad \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \tag{24}$$

$$C_{jk} \geq \Theta_{hk} - V(1 - x_{hjk}), \quad \forall h \in \mathcal{P}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{25}$$

$$C_{jk} \leq b_j, \quad \forall j \in \mathcal{B}, \forall k \in \mathcal{M}, \tag{26}$$

$$\sum_{i \in \mathcal{N}_0} x_{(h+1)ik} \leq \sum_{j \in \mathcal{N}_0} x_{hjk}, \quad \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \tag{27}$$

$$x_{0jk} = 0, \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \quad (28)$$

$$x_{00k} = 1, \quad \forall k \in \mathcal{M}, \quad (29)$$

$$\Theta_{0k} = 0, \quad \forall k \in \mathcal{M}, \quad (30)$$

$$C_{0k} = 0, \quad \forall k \in \mathcal{M}, \quad (31)$$

$$C_{\max} \geq C_{ik}, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \quad (32)$$

$$C_{ik}, \Theta_{ik} \geq 0, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \quad (33)$$

$$x_{hjk} \in \{0, 1\}, \quad \forall h \in \mathcal{P}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}. \quad (34)$$

The objective function (20) consists of minimizing the production makespan C_{\max} . Constraints (21) ensure that each job j is assigned to a unique position of exactly one machine. Constraints (22) guarantee the assignment of the dummy job 0 to a single position of the permutations used in the machines. Constraints (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 (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 (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 (26) ensure that the due dates of the jobs from the primary customers are met. Constraints (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 (28) avoid assigning a job at position 0 of the permutations as this position is uniquely used by the dummy job according to constraints (29). Constraints (30) and (31) set initial values to variables Θ_{0k} and C_{0k} , respectively. Finally, constraints (32) impose lower bounds to the makespan and constraints (33) and (34) define the type and domain of the decision variables.

Constraints (24) can be linearized similarly to constraints (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 (24):

$$\Theta_{hk} \geq \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{j \in \mathcal{N}} \bar{p}_{jk} x_{mjk} + \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \bar{s}_{ijk} y_{mijk}, \tag{35}$$

$$\forall h \in \mathcal{P}, \forall k \in \mathcal{M},$$

$$y_{hijk} \leq x_{(h-1)ik}, \quad \forall h \in \mathcal{P}, \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{36}$$

$$y_{lijk} \leq x_{lijk}, \quad \forall h \in \mathcal{P}, \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \tag{37}$$

$$y_{hijk} \geq x_{(h-1)ik} + x_{hjk} - 1, \quad \forall l \in \mathcal{P}, \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}. \tag{38}$$

Constraints (38) ensure that variable y_{hij}^k 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 (35)–(38) is satisfied for the binary variables y_{hij}^k if this set is also satisfied for the continuous variables y_{hij}^k in the interval $[0, 1]$. Hence, we obtain a MIP formulation for the problem by replacing constraints (24) with the set of constraints (35)–(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 Sect. 2. RO is a mathematical programming based methodology to model and solve optimization problems under uncertain parameters (Ben-Tal et al. 2009; Bertsimas et al. 2011, 2015). It has been successfully applied in different contexts, such as production-inventory problems (Bertsimas and Thiele 2006; De Ruiter et al. 2017), portfolio management (Bertsimas et al. 2018), emergency medical systems (Bertsimas and Ng 2019), terminal traffic flow (Ng et al. 2020), vehicle routing (Munari et al. 2019; De La Vega et al. 2019; De La Vega et al. 2020) and scheduling (Ng et al. 2017; Bougeret et al. 2019). Different from the stochastic programming methodology, RO does not require a complete knowledge of the probability distributions of the random parameters. 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). 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 2003, 2004).

3.1 Uncertain parameters and uncertainty set for the robust UPMSp

The uncertain parameters, namely 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. 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 the primitive random variables ξ_{ik} and η_{ijk} to cardinality-constrained sets defined 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, the *budget of uncertainty* Γ_k defines the number of uncertain parameters allowed to vary from their respective nominal values in machine k . The uncertainty sets are defined as follows, for each machine k :

$$\mathcal{U}_k(\Gamma_k) = \left\{ (\xi, \eta) \in \mathbb{R}^{(1+|\mathcal{N}_0)|\mathcal{M}} : \begin{aligned} & -1 \leq \xi_{ik} \leq 1, \forall i \in \mathcal{N}; \\ & -1 \leq \eta_{ijk} \leq 1, \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}; \sum_{i \in \mathcal{N}} |\xi_{ik}| + \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} |\eta_{ijk}| \leq \Gamma_k \end{aligned} \right\}. \quad (39)$$

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 (Soyster 1973) result from assigning values to Γ_k , for each $k \in \mathcal{M}$, equal to 0 and $|\mathcal{M}| + |\mathcal{N}_0||\mathcal{M}|$, respectively. Indeed, in the nominal problem, none of the uncertain parameters deviates from their nominal value, while in the Soyster problem, all of them attain their worst case.

3.2 Robust precedence-based formulation

We now develop a RO model for the addressed UPMSp under uncertainty, based on the nominal MIP model defined in Sect. 2.2 and using the uncertainty set defined in equation (39). The robust counterpart of the constraints (16) can be written as follows:

$$\sum_{i \in \mathcal{N}} \left[\bar{p}_{ik} z_{ijk} + \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \bar{s}_{lik} y_{lijk} \right] + \mathcal{B}_k^j(\mathbf{z}, \mathbf{y}, \Gamma_k) \leq C_{jk}, \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \quad (40)$$

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

$$\begin{aligned} \max_{\xi \geq 0, \eta \geq 0} & \left\{ \sum_{i \in \mathcal{N}} \hat{p}_{ik} z_{ijk}^* \xi_{ik} + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \hat{s}_{lik} y_{lijk}^* \eta_{lik} : \right. \\ & \xi_{ik} \leq 1, \quad \forall i \in \mathcal{N}, \\ & \eta_{lik} \leq 1, \quad \forall l \in \mathcal{N}_0, \forall i \in \mathcal{N}, \\ & \left. \sum_{i \in \mathcal{N}} \xi_{ik} + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \eta_{lik} \leq \Gamma_k \right\}. \end{aligned} \quad (41)$$

By duality, we can rewrite (41) as

$$\begin{aligned} \min_{\alpha \geq 0, \beta \geq 0, \lambda \geq 0} & \left\{ \sum_{i \in \mathcal{N}} \alpha_{ijk} + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \beta_{lijk} + \Gamma_k \lambda_{jk} : \right. \\ & \alpha_{ijk} + \lambda_{jk} \geq \hat{p}_{ik} z_{ijk}^*, \quad \forall i \in \mathcal{N}, \\ & \left. \beta_{lijk} + \lambda_{jk} \geq \hat{s}_{lik} y_{lijk}^*, \quad l \in \mathcal{N}_0, l \neq j, i \in \mathcal{N} \right\}, \end{aligned} \quad (42)$$

where the decision variables α_{ijk} , β_{lijk} and λ_{jk} are the dual variables associated with the primal optimization problem (41). Replacing constraints (16) with (40) in the MIP model of Sect. 2.2, we obtain the following RO model for the addressed UPMSP under uncertainty:

$$\text{Minimize} \quad C_{\max} \quad (43)$$

$$\begin{aligned} \text{subject to :} \\ \text{constraints: (2)-(4), (6)-(15), (17)-(19),} \end{aligned} \quad (44)$$

$$\sum_{i \in \mathcal{N}} \left[\bar{p}_{ik} z_{ijk} + \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \bar{s}_{lik} y_{lijk} \right] + \sum_{i \in \mathcal{N}} \alpha_{ijk} + \sum_{i \in \mathcal{N}} \sum_{\substack{l \in \mathcal{N}_0 \\ l \neq j}} \beta_{lijk} + \Gamma_k \lambda_{jk} \quad (45)$$

$$\leq C_{jk}, \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{M},$$

$$\alpha_{ijk} + \lambda_{jk} \geq \hat{p}_{ik} z_{ijk}, \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{M}, \quad (46)$$

$$\beta_{lijk} + \lambda_{jk} \geq \hat{s}_{lik} y_{lijk}, \quad \forall l \in \mathcal{N}_0, \quad \forall i, j \in \mathcal{N}, \quad (47)$$

$$l \neq j, \quad \forall k \in \mathcal{M},$$

$$\lambda_{jk} \geq 0, \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \quad (48)$$

$$\alpha_{ijk} \geq 0, \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{M}, \quad (49)$$

$$\beta_{lijk} \geq 0, \quad \forall l \in \mathcal{N}_0, \forall i, j \in \mathcal{N}, l \neq j, \forall k \in \mathcal{M}. \quad (50)$$

The objective function (43) consists of minimizing the makespan. Constraints (44) have already been explained and correspond to the constraints of the nominal MIP model. The set of constraints (45)–(50) is obtained from the robust counterpart (40), where $\mathcal{B}_k^j(z, y, \Gamma_k)$ is defined in (42). Note that there is no need for explicitly adding the minimization of (42) in model (43)–(50), as we are only concerned with guaranteeing the feasibility of constraints (45). Indeed, if there is a solution that satisfies (45), then the minimum of (42) also satisfies these constraints.

3.3 Robust position-based formulation

We can develop another RO model for the addressed UPMSp variant, associated to the nominal MIP model of Sect. 2.3 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 (35) by their respective random variables, similarly to the development presented in the previous subsection. Using the uncertainty set (39), the robust counterpart of the constraints (35) can be stated as follows:

$$\Theta_{hk} \geq \sum_{\substack{m \in P: \\ m \leq h}} \sum_{j \in \mathcal{N}} \hat{P}_{jk} x_{mjk} + \sum_{\substack{m \in P: \\ m \leq h}} \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \hat{S}_{ijk} y_{mijk} + \mathcal{B}_k^h(\mathbf{x}, \mathbf{y}, \Gamma_k), \quad \forall k \in \mathcal{M}, \forall h \in \mathcal{P}, \tag{51}$$

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

$$\begin{aligned} \max_{\xi \geq 0, \eta \geq 0} & \left\{ \sum_{\substack{m \in P: \\ m \leq h}} \sum_{j \in \mathcal{N}} \hat{P}_{jk} x_{mjk}^* \xi_{jk} + \sum_{\substack{m \in P: \\ m \leq h}} \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \hat{S}_{ijk} y_{mijk}^* \eta_{ijk} : \right. \\ & \xi_{jk} \leq 1, \quad \forall j \in \mathcal{N}; \\ & \eta_{ijk} \leq 1, \quad \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}; \\ & \left. \sum_{j \in \mathcal{N}} \xi_{jk} + \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \eta_{ijk} \leq \Gamma_k \right\}. \end{aligned} \tag{52}$$

Applying the duality technique to primal protection function (52), we can rewrite it as follows:

$$\begin{aligned} \min_{\gamma \geq 0, \vartheta \geq 0, \mu \geq 0} & \left\{ \sum_{j \in \mathcal{N}} \gamma_{jkh} + \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \vartheta_{ijhk} + \Gamma_k \mu_{hk} : \right. \\ & \gamma_{jkh} + \mu_{hk} \geq \sum_{\substack{m \in P: \\ m \leq h}} \hat{P}_{jk} x_{mjk}^*, \quad \forall j \in \mathcal{N}, \\ & \left. \vartheta_{ijhk} + \mu_{hk} \geq \sum_{\substack{m \in P: \\ m \leq h}} \hat{S}_{ijk} y_{mijk}^*, \quad \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N} \right\}, \end{aligned} \tag{53}$$

where the decision variables γ_{jkh} , ϑ_{ijhk} and μ_{hk} are the dual variables associated with the constraints of the inner optimization problem of primal protection function (52). Therefore, the RO model for the addressed UPMSP under uncertainty, associated to the nominal position-based MIP model of Sect. 2.3 can be stated as follows:

$$\text{Minimize } C_{\max}. \quad (54)$$

subject to:

$$\text{constraints: (21)-(23), (25)-(34), (36)-(38)}. \quad (55)$$

$$\Theta_{hk} \geq \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{j \in \mathcal{N}} \bar{p}_{jk} x_{mjk} + \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \bar{s}_{ijk} y_{mijk} + \sum_{j \in \mathcal{N}} \gamma_{jkh} + \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}} \vartheta_{ijhk} + \Gamma_k \mu_{hk},$$

$$\forall k \in \mathcal{M}, \forall h \in \mathcal{P}, \quad (56)$$

$$\gamma_{jkh} + \mu_{hk} \geq \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \hat{p}_{jk} x_{mjk}, \quad \forall j \in \mathcal{N}, \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \quad (57)$$

$$\vartheta_{ijhk} + \mu_{hk} \geq \sum_{\substack{m \in \mathcal{P}: \\ m \leq h}} \hat{s}_{ijk} y_{mijk}, \quad \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}, \forall h \in \mathcal{P}, \forall k \in \mathcal{M}, \quad (58)$$

$$\gamma_{jkh} \geq 0, \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall h \in \mathcal{P}, \quad (59)$$

$$\vartheta_{ijhk} \geq 0, \quad \forall i \in \mathcal{N}_0, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall h \in \mathcal{P}. \quad (60)$$

$$\mu_{hk} \geq 0, \quad \forall k \in \mathcal{M}, \forall h \in \mathcal{P}, \quad (61)$$

3.4 Other objective functions

The RO models presented in Sects. 3.2 and 3.3 (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 (6) and (26) of the nominal models and by appropriately changing the model objective functions. For instance, to minimize the total tardiness in the precedence-based formulation, we can include the variable T_{ik} to represent the tardiness of job i in machine k . Then, we replace constraints (6) with

$$T_{ik} \geq C_{ik} - b_i, \quad \forall i \in \mathcal{B}, \forall k \in \mathcal{M},$$

and the objective function (1), with the minimization of $\sum_{i \in \mathcal{B}} \sum_{k \in \mathcal{M}} T_{ik}$. Note that only jobs with due date (jobs in set \mathcal{B}) can present tardiness.

4 Computational experiments

The goal of this section is threefold: first, to compare the computational performance of the precedence-based robust model and the position-based robust model (Sect. 4.2); second, to expose the benefits of considering parameter uncertainties into the problem (Sect. 4.3); third, to illustrate the impact of the uncertainties on the structure of the scheduling solutions (Sect. 4.4). All instances and parameter values used in the computational experiments are described in Sect. 4.1. 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 3600 s or the relative optimality gap becoming smaller than 10^{-4} .

4.1 Instances and parameter values

We carried out the computational experiments using adapted instances from the literature. 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) and Lin et al. (2011) for the deterministic UPMSP with sequence- and machine-dependent setup times and due date constraints. We selected 16 original instances from Chen (2009) and Lin et al. (2011). Using these instances, we perform preliminary computational experiments and evaluate the efficiency of the CPLEX solver on the robust MIP models within the time limit. Given the difficulty of solving large-scale instances within the time limit using CPLEX, we present results with instances limited to at most 10 jobs and 4 machines. More specifically, we selected the first $|\mathcal{M}|$ machines and $|\mathcal{N}|$ jobs of each instance, where $|\mathcal{M}| \in \{2, 4\}$ and $|\mathcal{N}| \in \{8, 10\}$, totaling 64 instances in set A.

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) and 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:

$$C_m = \left(\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{M}} (p_{jk} + \frac{\sum_{i \in \mathcal{N}} s_{ijk}}{|\mathcal{M}|}) \right) / |\mathcal{M}|. \quad (62)$$

Table 1 Characteristics of the sets of instances

Class	$ M $	$ 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			

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 machine-job $(|\mathcal{M}| - |\mathcal{N}|)$, with $|\mathcal{M}| \in \{2, 4\}$ and $|\mathcal{N}| \in \{8, 10\}$, totaling 20 instances in each of these sets.

Table 1 summarizes the sets of instances, which are publicly available at <http://www.dep.ufscar.br/munari/upmsp>. 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 δ as the percentage of maximum allowed deviation of the processing and setup times in relation to their nominal values, corresponding to $\delta = 30\%$, 40% and 50% . Thus, $\hat{p}_{ik} = \delta \bar{p}_{ik}$ and $\hat{s}_{ijk} = \delta \bar{s}_{ijk}$. The *budgets of uncertainty* were considered integers from zero to five, i.e., $\Gamma_1 = \dots = \Gamma_k = \Gamma = 0, 1, 2, 3, 4, 5$.

4.2 Computational performance of the robust models

In this section, we compare the computational performance of the precedence- and position-based robust models. For this, we use performance profiles (Dolan and Moré 2002), which are briefly explained as follows. Given a set \mathcal{P} of instances and a set \mathcal{F} of solution approaches, let gap_{fp} be the gap of the solution

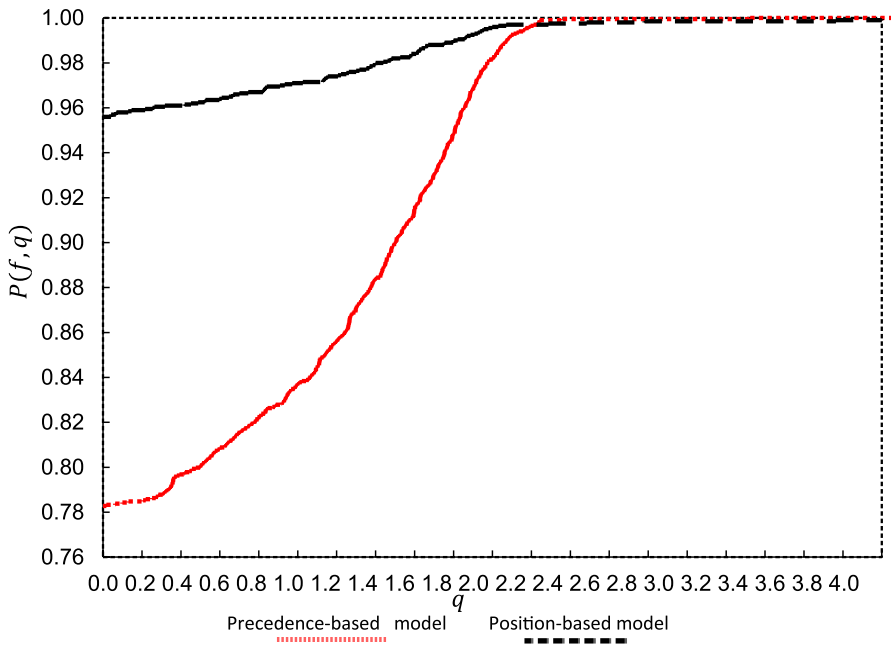


Fig. 1 Performance profile based on optimality gap

obtained for instance p using approach f . The value $P(f, q)$ (y -axis) when $q > 0$ (x -axis) indicates the fraction of instances for which the approach f provides solutions with a gap within a factor of 2^q of the best obtained gap, i.e., the fraction of instances for which $gap_{fp} + \epsilon \leq 2^q \cdot \min_{f' \in \mathcal{F}} \{gap_{f'p} + \epsilon\}$, where ϵ is a sufficiently small value set as 0.01. The performance profile curves for each of the robust models are shown in Fig. 1, from which we can infer the following for a given p instance:

- If we want to obtain a feasible solution with the best optimality gap, it is better to solve the position-based robust model. This is because at $q = 0.0$, CPLEX obtains feasible solutions with the best gaps in about 96% of the instances, using the position-based robust model. That percentage was just over 78% for the precedence-based robust model.
- If we want to obtain a feasible solution with gap within a factor of 2^q for values of $q \in [0.1, 2.4]$, it is preferable also to solve the position-based robust model. For those values of the factor q , the position-based robust model curve is always above the precedence-based robust model curve, which means that the first model has a higher chance of offering better quality solutions.
- For values of the q factor greater than 2.4, both robust models offer solutions within the same factor 2^q as the two curves overlap.

Therefore, based on these observations, we conclude that solving the position-based robust model is the best alternative to obtain feasible solutions with better gaps.

Tables 2 and 3 summarize the average computational results. 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 percentage of instances solved to optimality (%Opt.), the average objective function value (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. Tables 6 and 7 in Appendix A present the same results in a more detailed way, grouped by number of jobs and machines.

The results in Tables 2 and 3 confirm that the position-based robust model leads to a better overall performance than the precedence-based robust model for the problem instances used in these experiments. On average, CPLEX solved 85.75% of the instances to optimality with the position-based model, while with the precedence-based model it solved 69.53%, within the limit of 3600 s. Furthermore, the average gap and computational time are 3.1 and 4.4 times higher for the precedence-based model than for the position-based model.

From Tables 2 and 3, we also observe that, for higher values of Γ and δ , more instances become infeasible. The average gaps and computational times also increase, indicating that the instances become more challenging. Hence, the number of instances with a proven optimal solution tends to decrease as Γ and δ increase. For example, the average number of instances solved to proven optimality with the precedence-based model is 372 for $\Gamma = 0$ (out of 372) and 187 for $\Gamma = 5$ (out of 372). Similarly, for the same model, the number of instances solved to optimality

Table 2 Average computational results for different values of Γ

Γ	Inst.	Inf.	Precedence-based formulation				Position-based formulation					
			#Opt.	%Opt.	Avg. OF	Avg. gap	Avg. time	#Opt.	%Opt.	Avg. OF	Avg. gap	Avg. time
0	372	0	372	100.00	757	0.002	184	372	100.00	757	0.002	73
1	372	27	322	86.56	836	0.563	695	324	87.10	834	1.338	278
2	372	38	243	65.32	911	3.813	1430	313	84.14	909	1.416	300
3	372	41	221	59.41	966	6.279	1626	307	82.53	960	1.795	332
4	372	46	207	55.65	1003	7.477	1675	299	80.38	999	2.244	357
5	372	46	187	50.27	1028	9.414	1832	299	80.38	1023	2.075	356
Avg.	372	33	259	69.53	917	4.591	1240	319	85.75	914	1.479	282

Table 3 Average computational results for different values of δ

δ	Inst.*	Inf.	Precedence-based formulation				Position-based formulation					
			#Opt.	%Opt.	Avg. OF	Avg. time	Avg. gap	Avg. OF	Avg. OF	Avg. time		
0.3	744	45	555	74.60	874	3.064	1105	652	87.63	869	1.627	300
0.4	744	52	520	69.89	915	4.642	1239	650	87.37	911	1.441	286
0.5	744	101	477	64.11	952	5.793	1307	612	82.26	950	1.254	245
Avg.	744	66	517	69.53	914	4.499	1217	638	85.75	910	1.441	277

drops from 555 to 477 (out of 744) when δ increases from 0.3 to 0.5. Thus, considering uncertainty in the problem clearly makes it more difficult to solve using the proposed models.

4.3 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, exposing the benefits of incorporating parameter uncertainties into the problem. 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 $x^{\delta, \Gamma}$ be the optimal solution for a given deviation δ and budget of uncertainty Γ , obtained from the position-based formulation. The PoR is evaluated as $\frac{z(x^{\delta, \Gamma}) - z}{z} \cdot 100\%$, in which $z(x^{\delta, \Gamma})$ 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 makespan of these 10,000 generated samples (i.e., we sum the makespan of all the samples and divide the total sum by the number of samples).

Table 4 shows the average objective function cost (OF) of the robust model, 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., $\delta = 0.5$) – experiments using $\delta = 0.3$ and $\delta = 0.4$ show similar overall results. Table 8 in Appendix A presents the same results as in Table 4, but they are detailed for different numbers of jobs and machines. Note that, in general, most of the instances present solutions with low risk when the number of uncertain parameters that can assume their worst case is greater than two (i.e., $\Gamma > 2$). 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%.

The objective function cost and the expected cost of the solutions 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

Table 4 Average 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 $\delta = 0.5$

Γ	A						B1						B2						B3					
	OF	Exp. cost	PoR	Risk	Exp. cost	PoR	OF	Exp. cost	PoR	Risk	Exp. cost	PoR	OF	Exp. cost	PoR	Risk	OF	Exp. cost	PoR	Risk	Exp. cost	PoR	Risk	
0	880	997	0.00	28.80	450	507	0.00	48.66	710	803	0.00	44.69	715	809	0.00	44.69	715	809	0.00	44.69	715	809	0.00	50.70
1	1053	1023	19.70	7.37	498	514	10.71	4.65	791	809	11.43	1.34	793	815	11.00	1.32	793	815	11.00	1.32	793	815	11.00	1.32
2	1184	1026	34.63	1.55	539	515	19.68	2.11	871	818	22.64	0.00	870	823	21.74	0.00	870	823	21.74	0.00	870	823	21.74	0.00
3	1261	1019	43.31	0.00	575	520	27.86	0.00	934	819	31.51	0.00	928	824	29.92	0.00	928	824	29.92	0.00	928	824	29.92	0.00
4	1304	1025	48.25	0.00	608	520	35.07	0.00	980	816	38.05	0.00	982	827	37.45	0.00	982	827	37.45	0.00	982	827	37.45	0.00
5	1336	1024	51.84	0.00	632	521	40.42	0.00	1015	816	42.86	0.00	1012	823	41.64	0.00	1012	823	41.64	0.00	1012	823	41.64	0.00

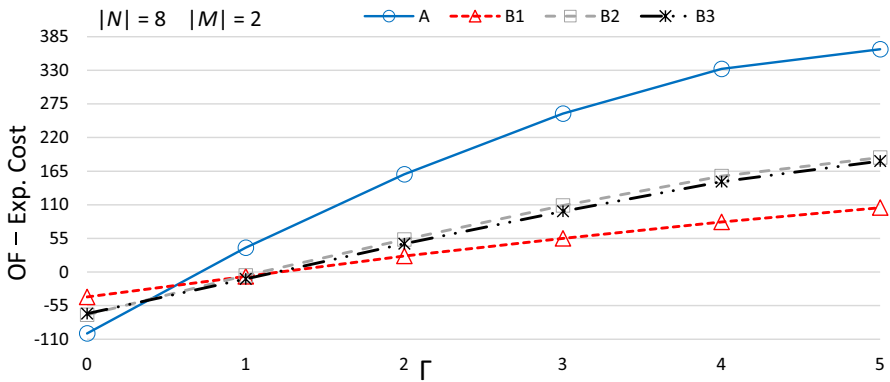


Fig. 2 Difference between the objective function cost and the expected cost of the solutions for instances with 8 jobs and 2 machines

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

Table 5 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

solutions that underestimate the empirical actual cost of the solutions, while high values of Γ provide solutions that overestimate this measure.

4.4 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 three of these jobs are from primary customers and thus their due dates cannot be violated. The third column of Table 5 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 scenarios with 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 in scenarios where $\Gamma' = 0, 1, 2, 3$ uncertain parameters attain their worst-case deviations.

Table 5 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 bold 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.

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 RO solution obtained for $\Gamma = 0$ (i.e., the deterministic/nominal model) 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 $\Gamma' > \Gamma$, all solutions are infeasible, as the due date of at least one job is violated. 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). 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).

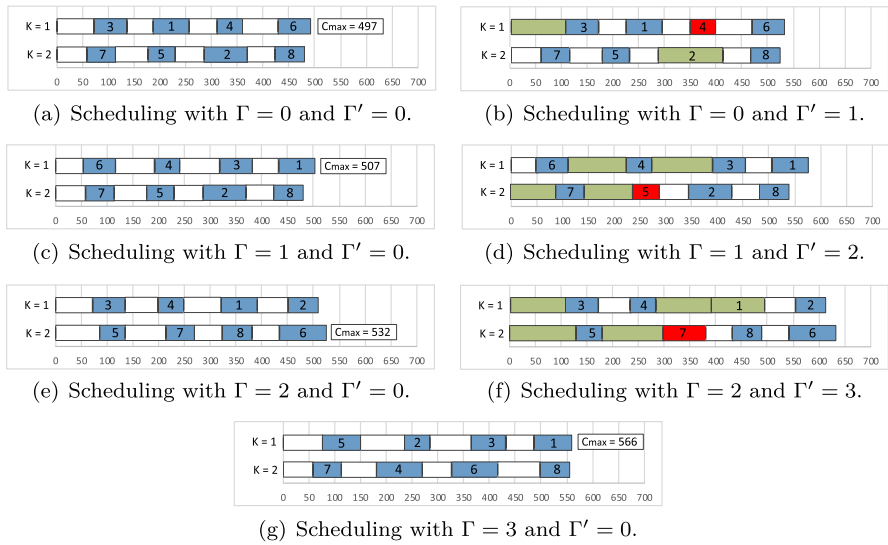


Fig. 3 Solutions of the illustrative example for different values of Γ and Γ'

Figure 3 shows the impact of considering $\Gamma' > \Gamma$ in some solutions of Table 5. 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 3a shows the (deterministic) solution obtained by the RO models for $\Gamma = 0$. Machine $k = 1$ processes jobs 3, 1, 4 and 6, while machine $k = 2$ processes jobs 7, 5, 2 and 8. Figure 3b shows that this deterministic solution becomes infeasible for $\Gamma' = 1$ if the first setup time of machine $k = 1$ increases, as it makes the completion time of job 4 higher than its due date. To avoid this infeasibility, we can use the RO models with $\Gamma = 1$ to obtain the solution given in Fig. 3c, which has a different production sequence of jobs in machine $k = 1$. 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 Fig. 3d). A solution obtained with $\Gamma = 2$ avoids such infeasibility, as presented in Fig. 3e, which has a new assignment of jobs to machines and different production sequences. Finally, this solution becomes infeasible as well if up to three parameters attain their worst-case ($\Gamma' = 3$), as presented in Fig. 3f, in which the due date of job 7 is violated. Figure 3g shows a feasible solution for this last scenario, obtained using the RO models with $\Gamma = 3$, in which jobs 4, 5 and 7 (the ones 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. Therefore, the increase of Γ values in the RO models changes the schedule configurations to provide more protected solutions, at the cost of higher makespans.

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 precedence-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, leading to managerial insights that can bring benefits in real contexts. For instance, the results suggest that the risk is substantially reduced for relatively small values of Γ , even considering large values of δ . These robust schedules can be useful for risk-averse managers, as the cost increases of these solutions are not too high. Managing these decisions in a tractable way and under uncertainty is often a challenge in practice. Furthermore, it was observed that ignoring the parameter uncertainties can result in infeasible schedules or in schedules with higher makespans. Regarding the computational performances observed for the two models, CPLEX was able to solve a larger number of instances to optimality using the robust position-based model than when using the robust precedence-based model.

Using the proposed models, we were able to find solutions for small instances only, considering a reasonable time limit. Hence, an interesting line of future research would be to develop effective methods to solve medium and large instances, such as meta-heuristics and tailored exact methods based on the problem 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.

Appendix A: Additional computational results

See Tables 6, 7, and 8.

Table 6 Average computational results for different values of Γ

$ \mathcal{M} $	$ \mathcal{M} $	Γ	Inst.*	Inf.	Precedence-based formulation				Position-based formulation			
					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	1049	0.002	387.12	81	1048	0.001	106.74
8	2	2	93	14	76	1131	0.930	664.09	79	1131	0.000	114.85
8	2	3	93	17	70	1220	1.509	809.32	76	1210	0.000	105.72
8	2	4	93	20	65	1291	2.720	939.15	73	1280	0.001	98.85
8	2	5	93	20	59	1321	4.162	1107.16	73	1310	0.000	88.68
10	2	0	93	0	93	1169	0.003	239.29	93	1169	0.005	81.94
10	2	1	93	12	66	1253	1.071	1382.71	81	1247	0.002	35.69
10	2	2	93	13	27	1343	6.337	2835.20	80	1336	0.003	37.64
10	2	3	93	13	21	1411	9.648	3110.61	80	1411	0.002	38.99
10	2	4	93	13	16	1469	13.306	3124.91	80	1469	0.003	35.94
10	2	5	93	13	9	1525	17.075	3295.03	80	1524	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

Table 6 (continued)

M	M	Γ	Inst.*	Inf.	Precedence-based formulation			Position-based formulation				
					Opt.	Avg. OF	Avg. gap	Avg. time	Opt.	Avg. OF	Avg. gap	Avg. time
10	4	2	93	9	49	638	7,983	1878.48	63	638	5,662	969.56
10	4	3	93	9	39	680	13,962	2255.01	60	670	7,177	1093.88
10	4	4	93	10	36	693	13,882	2358.36	56	686	8,974	1224.67
10	4	5	93	10	29	704	16,417	2613.63	56	695	8,298	1219.63
Avg.			93	8	65	917	4,591	1240.31	80	914	1,479	282.45

* The number of instances is calculated as the product of the number of deterministic instances with IM jobs and IM machines (16 from set A, 5 from set B1, 5 from set B2, 5 from set B3) and the number of possible δ values (0.3, 0.4, 0.5), i.e., $31 \cdot 3 = 93$

Table 7 Average computational results for different values of δ

M	M	δ	Inst.*	Inf.	Precedence-based formulation			Position-based formulation				
					Opt.	Avg. OF	Avg. gap	Avg. time	Opt.	Avg. OF	Avg. gap	Avg. time
8	2	0.3	186	20	161	1101	0.66	575.25	166	1100	0.00	92.07
8	2	0.4	186	23	156	1162	1.28	624.82	163	1151	0.00	101.97
8	2	0.5	186	40	127	1198	2.60	788.26	146	1198	0.00	102.93
10	2	0.3	186	20	85	1323	5.68	2129.96	166	1306	0.00	45.61
10	2	0.4	186	20	72	1364	7.93	2338.20	166	1361	0.00	45.72
10	2	0.5	186	24	75	1397	9.39	2346.02	162	1397	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	1441.61	134	591	6.50	988.04
10	4	0.4	186	9	106	627	9.35	1713.01	135	624	5.76	913.53
10	4	0.5	186	27	99	669	11.17	1820.78	128	659	5.01	761.13
Avg.			186	16.5	129.33	914	4.499	1217.15	159.50	910	1.441	277.23

* The number of instances is calculated as the product of the number of deterministic instances with |M| jobs and |M| machines (16 from set A, 5 from set B1, 5 from set B2, 5 from set B3) and the number of possible Γ values (0, 1, 2, 3, 4, 5), i.e., $31 \cdot 6 = 186$

Table 8 Average 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 $\delta = 0.5$

M	M	Γ	B1					B2					B3						
			OF	Exp. cost	PoR	Risk	Risk	OF	Exp. cost	PoR	Risk	Risk	OF	Exp. cost	PoR	Risk	Risk		
8	2	0	1194	1347	0.00	30.06	0.00	518	582	0.00	71.65	823	931	0.00	39.81	828	933	0.00	61.22
8	2	1	1412	1392	18.21	21.03	570	592	10.16	2.50	2.50	918	945	11.52	0.17	912	948	10.25	1.04
8	2	2	1558	1398	30.49	0.00	621	601	20.01	0.00	0.00	1005	961	22.14	0.00	990	950	19.60	0.00
8	2	3	1684	1384	40.98	0.00	668	613	28.97	0.00	0.00	1077	959	30.89	0.00	1057	955	27.76	0.00
8	2	4	1766	1380	47.84	0.00	710	618	37.18	0.00	0.00	1130	950	37.38	0.00	1121	955	35.41	0.00
8	2	5	1808	1381	51.35	0.00	741	618	43.11	0.00	0.00	1170	949	42.24	0.00	1166	954	40.85	0.00
10	2	0	1422	1610	0.00	24.60	647	727	0.00	76.16	1019	1148	0.00	5.29	1031	1164	0.00	65.40	
10	2	1	1652	1657	16.17	0.04	699	738	8.13	16.09	1099	1149	7.91	5.20	1111	1171	7.82	1.80	
10	2	2	1833	1678	28.90	0.00	750	745	15.93	8.43	1185	1161	16.33	0.00	1196	1186	16.05	0.00	
10	2	3	1963	1668	38.04	0.00	787	746	21.68	0.00	0.00	1261	1166	23.81	0.00	1264	1185	22.61	0.00
10	2	4	2053	1658	44.37	0.00	826	748	27.71	0.00	0.00	1330	1165	30.53	0.00	1329	1187	28.93	0.00
10	2	5	2138	1657	50.31	0.00	859	748	32.82	0.00	0.00	1396	1168	37.06	0.00	1392	1186	35.09	0.00
8	4	0	416	474	0.00	17.61	274	310	0.00	20.00	418	477	0.00	55.13	423	484	0.00	57.05	
8	4	1	538	478	29.25	8.43	319	314	16.42	0.00	0.00	494	483	18.18	0.00	500	485	18.32	0.00
8	4	2	623	492	49.91	6.19	355	311	29.44	0.00	0.00	570	488	36.27	0.00	572	488	35.31	0.00
8	4	3	640	489	53.98	0.00	384	310	40.02	0.00	0.00	609	486	45.60	0.00	607	487	43.59	0.00
8	4	4	640	486	53.99	0.00	410	309	49.51	0.00	0.00	639	485	52.97	0.00	637	486	50.78	0.00
8	4	5	640	486	53.99	0.00	410	310	49.51	0.00	0.00	639	485	52.97	0.00	637	487	50.78	0.00

Table 8 (continued)

$ \mathcal{M} $	$ \mathcal{M}' $	Γ	A	B1					B2					B3				
				OF	Exp. cost	PoR	Risk	Risk	OF	Exp. cost	PoR	Risk	Risk	OF	Exp. cost	PoR	Risk	Risk
10	4	0	487	554	0.00	42.95	362	408	0.00	26.84	581	657	0.00	78.52	577	654	0.00	19.12
10	4	1	611	563	25.53	0.00	405	412	11.77	0.00	654	660	12.60	0.00	649	658	12.40	2.46
10	4	2	722	536	48.46	0.00	429	404	18.51	0.00	724	663	24.61	0.00	722	668	25.04	0.00
10	4	3	756	533	55.30	0.00	464	409	28.08	0.00	789	664	35.77	0.00	786	671	36.07	0.00
10	4	4	758	574	55.70	0.00	486	405	34.25	0.00	822	664	41.45	0.00	842	680	45.83	0.00
10	4	5	758	574	55.70	0.00	519	409	43.28	0.00	852	663	46.64	0.00	853	666	47.79	0.00

Acknowledgements The authors thank the anonymous reviewers for their valuable comments.

References

- Afzalirad M, Rezaeian J (2016) Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Comput Ind Eng* 98:40–52
- Alem D, Morabito R (2012) Production planning in furniture settings via robust optimization. *Comput Oper Res* 39(2):139–150
- Arnaout J-P, Rabadi G, Musa R (2010) A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. *J Intell Manuf* 21(6):693–701
- Arnaout J-PP, Musa R, Rabadi G, Rami J-PA, Ghaith M (2014) A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines—part II: enhancements and experimentations. *J Intell Manuf* 25(1):43–53
- Avalos-Rosales O, Angel-Bello F, Alvarez A (2015) Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times. *Int J Adv Manuf Technol* 76(9):1705–1718
- Ben-Tal A, Nemirovski A (2000) Robust solutions of linear programming problems contaminated with uncertain data. *Math Program Ser B* 88:411–424
- Ben-Tal A, El Ghaoui L, Nemirovski A (2009) Robust optimization. Princeton Series in Applied Mathematics. Princeton University Press, Princeton
- Bertsimas D, Ng Y (2019) Robust and stochastic formulations for ambulance deployment and dispatch. *Eur J Oper Res* 279(2):557–571
- Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. *Math Program* 98(1–3):49–71
- Bertsimas D, Sim M (2004) The price of robustness. *Oper Res* 52(1):35–53
- Bertsimas D, Thiele A (2006) A robust optimization approach to inventory theory. *Oper Res* 54(1):150–168
- Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. *SIAM Rev* 53(3):464–501
- Bertsimas D, Goyal V, Lu B (2015) A tight characterization of the performance of static solutions in two-stage adjustable robust linear optimization. *Math Program* 150(2):281–319
- Bertsimas D, Gupta V, Kallus N (2018) Data-driven robust optimization. *Math Program* 167(2):235–292
- Bougeret M, Pessoa AA, Poss M (2019) Robust scheduling with budgeted uncertainty. *Discrete Appl Math* 261:93–107
- Chen J-F (2009) Scheduling on unrelated parallel machines with sequence- and machine-dependent setup times and due-date constraints. *Int J Adv Manuf Technol* 44(11–12):1204–1212
- Chen J-F (2015) Unrelated parallel-machine scheduling to minimize total weighted completion time. *J Intell Manuf* 26(6):1099–1112
- Cheng T, Sin C (1990) A state-of-the-art review of parallel-machine scheduling research. *Eur J Oper Res* 47(3):271–292
- Chyu C-C, Chang W-S (2011) Optimizing fuzzy makespan and tardiness for unrelated parallel machine scheduling with archived metaheuristics. *Int J Adv Manuf Technol* 57(5–8):763–776
- De La Vega J, Munari P, Morabito R (2019) Robust optimization for the vehicle routing problem with multiple deliverymen. *Cent Eur J Oper Res* 27(4):905–936
- De La Vega J, Munari P, Morabito R (2020) Exact approaches to the robust vehicle routing problem with time windows and multiple deliverymen. *Comput Oper Res* 124:105062
- De Ruiter FJCT, Ben-Tal A, Brekelmans RCM, den Hertog D (2017) Robust optimization of uncertain multistage inventory systems with inexact data in decision rules. *Comput Manag Sci* 14(1):45–66
- Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. *Math Program* 91(2):201–213
- Feng X, Zheng F, Xu Y (2016) Robust scheduling of a two-stage hybrid flow shop with uncertain interval processing times. *Int J Prod Res* 54(12):3706–3717

- Gharehgozli A, Tavakkoli-Moghaddam R, Zaerpour N (2009) A fuzzy-mixed-integer goal programming model for a parallel-machine scheduling problem with sequence-dependent setup times and release dates. *Robot Comput Integr Manuf* 25(4–5):853–859
- Gomes FRA, Mateus GR (2017) Improved combinatorial benders decomposition for a scheduling problem with unrelated parallel machines. *J Appl Math* 2017:1–10
- González-Neira EM, Montoya-Torres JR, Barrera D (2017) Flow-shop scheduling problem under uncertainties: review and trends. *Int J Ind Eng Comput* 8(4):399–426
- Gorissen BL, Yamkoğlu I, den Hertog D (2015) A practical guide to robust optimization. *Omega* 53:124–137
- Hu H, Ng K KH, Qin Y (2016) Robust parallel machine scheduling problem with uncertainties and sequence-dependent setup time. *Sci Program*, 13
- Liao TW, 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. *Appl Soft Comput* 56:65–81
- Lin S-W, Lu C-C, Ying K-C (2011) Minimization of total tardiness on unrelated parallel machines with sequence- and machine-dependent setup times under due date constraints. *Int J Adv Manuf Technol* 53(1–4):353–361
- Macedo PB, Alem D, Santos M, Junior ML, Moreno A (2016) Hybrid manufacturing and remanufacturing lot-sizing problem with stochastic demand, return, and setup costs. *Int J Adv Manuf Technol* 82(5):1241–1257
- Munari P, Moreno A, De La Vega J, Alem D, Gondzio J, Morabito R (2019) The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method. *Transp Sci* 53(4):1043–1066
- Ng K, Lee C, Chan FT, Qin Y (2017) Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min–max regret approach. *Transp Res Part E Logist Transp Rev* 106:115–136
- Ng K, Lee C, Chan FT, Chen C-H, Qin Y (2020) A two-stage robust optimisation for terminal traffic flow problem. *Appl Soft Comput* 89:106048
- Nogueira JP, Arroyo JEC, Villadiego HMM, Gonçalves LB (2014) Hybrid GRASP heuristics to solve an unrelated parallel machine scheduling problem with earliness and tardiness penalties. *Electron Notes Theor Comput Sci* 302:53–72
- Ravetti MG, Mateus GR, Rocha PL, Pardalos PM (2007) A scheduling problem with unrelated parallel machines and sequence dependent setups. *Int J Oper Res* 2(4):380
- Rodríguez SV, Albornoz VM, Plá LM (2009) A two-stage stochastic programming model for scheduling replacements in sow farms. *Top* 17(1):171–189
- Sadati A, Tavakkoli-moghaddam R, Naderi B, Mohammadi M (2017) Solving a new multi-objective unrelated parallel machines scheduling problem by hybrid teaching-learning based optimization. *Int J Eng* 30(2):224–233
- Salehi Mir MS, 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. *Appl Soft Comput* 41:488–504
- Salmasnia A, Khatami M, Kazemzadeh RB, Zegordi SH (2015) Bi-objective single machine scheduling problem with stochastic processing times. *Top* 23(1):275–297
- Seo K-K, Chung BD (2014) Robust optimization for identical parallel machine scheduling with uncertain processing time. *J Adv Mech Design Syst Manuf* 8(2):1–11
- Soyster AL (1973) Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper Res* 21(5):1154–1157
- Sungur I, Ordóñez F, Dessouky M (2008) A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Trans (Inst Ind Eng)* 40(5):509–523
- Tadayon B, Smith JC (2015) Algorithms and complexity analysis for robust single-machine scheduling problems. *J Sched* 18(6):575–592
- Torabi S, Sahebjamnia N, Mansouri S, Bajestani MA (2013) A particle swarm optimization for a fuzzy multi-objective unrelated parallel machines scheduling problem. *Appl Soft Comput* 13(12):4750–4762
- Xu X, Cui W, Lin J, Qian Y (2013) Robust makespan minimisation in identical parallel machine scheduling problem with interval data. *Int J Prod Res* 51(12):3532–3548

- Ying K-C, Lee Z-J, Lin S-W (2012) Makespan minimization for scheduling unrelated parallel machines with setup times. *J Intell Manuf* 23(5):1795–1803
- Zeidi JR, MohammadHosseini S (2015) Scheduling unrelated parallel machines with sequence-dependent setup times. *Int J Adv Manuf Technol* 81(9–12):1487–1496

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.