A branch-and-bound approach to minimise the value-at-risk of the makespan in a stochastic two-machine flow shop

Lei Liu and Marcello Urgo
Department of Mechanical Engineering, Politecnico di Milano, Milano, Italy

May 2023

Abstract

Planning and scheduling approaches in real manufacturing environments entail the need to cope with random attributes and variables to match the characteristics of real scheduling problems where uncertain events are frequent. Moreover, the capability of devising robust schedules, which are less sensitive to the disruptive effects of unexpected events, is a major request in real applications. In this paper, a branch-and-bound approach is proposed to solve the two-machine permutation flow shop scheduling problem with stochastic processing times. The objective is the minimisation of the value-at-risk of the makespan, to support decision-makers in the trade-off between the expected performance and the mitigation of the impact of extreme scenarios. A Markovian Activity Network (MAN) model is adopted to estimate the distribution of the makespan and assess the value-at-risk for both partial and complete schedules. Phase-type distributions are used to enable general distributions for processing times while maintaining the capability to exploit a Markovian approach. The effectiveness and performance of the proposed approach are demonstrated through a set of computational experiments.

Key words: Flow shop; Stochastic scheduling; Phase-type; Markovian activity networks; Value-at-risk

1 Introduction

Production planning and control need to cope with the intrinsic uncertainty of real manufacturing environments (Aytug et al. 2005), characterised by incomplete information, and unexpected events that may stem from a wide range of sources, e.g., the duration of production activities could vary, new activities, like rush orders or reworks, could need to be executed with a higher priority, order cancellations, random machine breakdowns, and shortage of materials (Allahverdi and Aydilek 2010; Alfieri et al. 2011). Special and relevant cases are remanufacturing processes that, differently from manufacturing ones, are characterised by considerable uncertainty due to the variable and unpredictable wear of used parts (Liu and Urgo 2022).

Stochastic and robust scheduling approaches have been developed to support planning and scheduling decisions, aiming at modelling uncertainties, mitigating the impact of uncertain events, and protecting the performance of a production schedule (Urgo and Váncza 2019). In stochastic scheduling approaches, relevant sources of uncertainty are modelled by defining random variables and the associated probability distributions. Different measures, such as expected makespan, maximum regret, value-at-risk and conditional value-at-risk, are exploited to indicate the robustness of the schedule. Among these, value-at-risk and conditional value-at-risk are designed to optimise the overall performance while avoiding the impact of extreme events that may lead to very poor performance of the objective function (Radke et al. 2013). Using these risk measures to support the devising of a robust scheduling solution is one of the most promising topics (Tolio et al. 2011; Alfieri et al. 2012; Urgo and Váncza 2019; Filippi et al. 2020).

However, exploiting risk measures as the optimisation criterion (e.g., minimising the value-at-risk of the makespan) entails estimating the distribution of the considered objective function, which could be difficult even for relatively small scheduling problems (Dodin 1985, 1996). Markovian Activity Network (MAN), which exploits a Markovian model to represent the execution of the activities in the network as a Continuous Time Markov Chain (CTMC), is a powerful tool to address this estimation problem (Kulkarni and Adlakha 1986) and has been extended to model generally distributed processing times by using phase-type distributions (Angius et al. 2021).

This paper studies the requirements for scheduling remanufacturing activities modelled as a two-machine permutation flow shop and the parts processed in batches. Furthermore, the dimension of each batch is not known in advance since some parts could be too damaged to be repaired and must be substituted with new ones. Processing times also entail a certain degree of uncertainty, i.e., parts with a higher degree of damage require longer processing times compared to less severe damages. The uncertainty associated with these factors is embedded in the processing times for a batch of parts, described through a probability distribution. Without loss of generality, phase-type distributions, which can approximate general distributions, are incorporated into the Markovian Activity Network model to estimate the distribution of the objective function. To mitigate the propagation of uncertain events throughout the production process, an exact branch-and-bound algorithm is being proposed to minimise the value-at-risk of the makespan.

The paper is organised as follows: Section 2 reviews relevant literature, Section 3 describes the addressed scheduling problem and the risk measure used, Section 4 presents the proposed branch-and-bound approach while the results of the experiments are reported in Section 5. Section 6 provides the managerial insights to guide decision-makers when developing robust schedules for their manufacturing systems. Finally, Section 7 gives the final considerations and conclusions.

2 Literature Review

Within the vast corpus of contributions related to flow shop scheduling (Johnson 1954), the stochastic version of this problem, where the processing times of the jobs are modelled through probability distributions, has attracted significant attention due to its relevance in relation to the characteristics of real manufacturing environments. Due to the complexity of stochastic scheduling problems, most of the existing literature models the uncertainty through discrete scenarios to support optimisation approaches based on stochastic programming (Fathollahi-Fard et al. 2021; Gholizadeh et al. 2021). In the cases where uncertainty is modelled through stochastic probability distributions, exponential distributions are often used (e.g., to model processing times), resulting in simplified solution approaches(Emmons and Vairaktarakis 2012). For this class of problems, if the objective is to minimise the expected makespan, TalwarâĂŹs rule (Talwar 1967) has been proven to be optimal (Cunningham and Dutta 1973).

Aiming at the generation of robust schedules, specific objective functions have been proposed. The most popular is minimising the expected maximum completion time Gourgand et al. (2000). Although able to consider the impact of extreme scenarios, this criteria ignores the actual probability distributions linked to uncertain parameters, resulting in possible overcautious decisions (Tolio and Urgo 2013). To incorporate the available information on stochastic variables, the variance of the objective function could be considered, together with the expected value, to optimise the expectation-variance tradeoff (De et al. 1992; Sarin et al. 2010).

Another class of robust scheduling approaches grounds on risk measures to pursue a trade-off between the value of the objective function and the mitigation of the impact of extreme cases (Liu et al. 2019, 2021; Benmansour et al. 2012). A popular risk mitigation criterion is the minimax regret, i.e., minimising the worst-case increment in the objective function that may occur because scheduling decisions are taken before the actual realisation of uncertainty (Savage 1951; Averbakh and Berman 1997; Xu et al. 2013; Levorato et al. 2022). Kouvelis et al. (2000) and Kasperski et al. (2012) address the minimisation of the maximum regret for a two-machine flow shop, grounding on the assessment of the worst performance over all the potential realisations of the processing times of the jobs. The uncertainty affecting the processing times is modelled through two frameworks, one considering discrete scenarios and the other based on continuous processing time intervals.

Nevertheless, the accurate modelling of uncertainty through a discrete set of scenarios requires an extremely high number of them, making the problem difficult to tackle. At the same time, exponential distribution also have limitations, since they they are not suitable to represent the variability of real manufacturing processes. Referring to the possible optimisation criteria to pursue robustness, besides the minimisation of the expected makespan, that fails in capturing the actual impact of uncertainty (Tolio and Urgo 2013), also the minimisation of the variance does not match the requirements, because it indiscriminately penalises both positive and negative deviations from the mean value (Sarin et al. 2014; Meloni and Pranzo 2020). Hence, this criterion is inappropriate when the goal is to hedge against the makespan exceeding a certain value only, while we do not want to penalise downward deviations (Wiesemann 2010). At the same time, grounding scheduling decisions on worst-case scenarios, which may be unlikely to occur, often results in excessively cautious and too conservative decisions(Bertsimas and Sim 2004; Tetenov 2012).

To overcome the limitations of these optimisation criteria, the actual distribution of the objective function must be calculated as a support to the definition of more meaningful criteria. Different fitting approaches, e.g., mixtures of normal distributions (Sarin et al. 2010) and phase-type distributions (OâĂŹCinneide 1990), have been developed to support the estimation of the distribution of some objective functions, e.g., the makepsan, without too constraining hypotheses on the distributions of uncertain variables, e.g, processing times. Grounding on this, one-sided risk measures, e.g., the value-at-risk (VaR), can be used to consider the trade-off between the expected performance and the protection against extreme scenarios. Risk measures have been broadly used in the financial area, e.g., in portfolio management, to hedge against uncertainties and deal with extreme scenarios, but not so widespread in the scheduling area. Within the contributions addressing this area, the value-at-risk and conditional-value-at-risk have been considered a valuable optimisation criterion and mostly applied to single machine scheduling problems (Filippi et al. 2020). In this perspective, Sarin et al. (2014) proposed a scenario-based mixed-integer program formulation for minimising the conditional value-at-risk for the single machine and parallel machine scheduling problem to minimise the total weighted tardiness. Atakan et al. (2017) addressed a single machine scheduling problem to minimise the value-at-risk of the total tardiness and the total weighted tardiness. Chang et al. (2017) proposed a robust optimisation model for the single machine scheduling problem, with random job processing times, exploiting information on their mean and covariance to find an optimal schedule minimising the conditional value-at-risk of the total flow time. A branch-and-bound approach was proposed for the stochastic single machine scheduling problem with uncertain processing time and release time in Urgo and Váncza (2019) to minimise the valueâĂŘatâĂŘrisk of the maximum lateness, and for a flow shop stochastic scheduling approach minimising the CVaR of the residual work content in Urgo (2019). Kasperski and Zieliński (2019) discussed a wide class of single-machine scheduling problems with uncertain job processing times and due dates and applied risk measure criteria (VaR and CVaR) to obtain an optimal solution. Meloni and Pranzo (2020) evaluated the conditional value-at-risk of the makespan for a resource-constrained project scheduling problem where, for each activity, an interval for processing times is defined in the integer domain and the evaluation of quantile- and superquantiles-based risk measures for the interval-valued processing times in scheduling problems is addressed in Meloni and Pranzo (2023). Several researchers also proposed heuristic approaches for stochastic scheduling problems considering risk measures (Rezaei et al. 2020; Villarinho et al. 2021). A summary of the available approaches for these classes of scheduling problems is shown in Table 1, where P1 refers to single machine scheduling problems, Pm parallel machine ones, Fm stands for flow shop scheduling, and Jm denotes job shop scheduling problems. Grounding on this summary, it emerges that no contribution exists addressing exact approaches to solve the two-machine flow shop scheduling problem with generally distributed processing times to minimise the value-at-risk of makespan.

The exact estimation of the distribution of the makespan in scheduling problems where the processing time of a job is described with a probability distribution is recognised as a significant difficulty (Dodin 1985, 1996). Markovian Activity Networks (MAN) have been proposed to exactly estimate this distribution, given that the processing times of the jobs follow an exponential distribution (Kulkarni and Adlakha 1986). To overcome the limitation to exponential distributions only, extensions have been proposed to cope with generally distributed processing times (Urgo 2014). Creemers (2015, 2018) addressed the

Table 1 Relevant studies on risk measure based stochastic scheduling.

References	Objective function	Problem	Uncertainty model	Computational technique	Solution methodologies
Talwar (1967)	$E(C_{max})$	F2	Exp distribution	Exact	Dispatchng Rule
Creemers (2015)	$E(C_{max})$	RCPSP	Exp and Coxian	Exact	Dynamic programming
Seif et al. (2020)	$E(Cost)$	Fm	Discrete scenarios	Exact	Stochastic MIP
Sarin et al. (2010)	$E(C_{max})$ &Var	Pm/Fm/Jm	Mixture normal	Approximation	Clark equation
Kouvelis et al. (2000)	Max regret	F2	Interval data	Exact	Branch-and-bound
Levorato et al. (2022)	Maximum	F2	Interval budgeted data	Exact	MILP
Kasperski et al. (2012)	Max regret	F2	Discrete scenarios	Approximation	Theoretical bounds
Atakan et al. (2017)	VaR	P1	Discrete scenarios	Exact	Stochastic programming
Urgo and Váncza (2019)	VaR	P1	Triangular	Exact	Branch-and-bound
Urgo (2019)	CVaR	Fm/no-wait	Stochastic distribution	Exact	Branch-and-bound
Sarin et al. (2014)	CVaR	P1	Discrete scenarios	Exact	Stochastic programming
Chang et al. (2017)	CVaR	P1	Ambiguity set	Exact	MIP
Meloni and Pranzo (2020)	CVaR	RCPSP	Interval data	Approximation	Theoretical bounds
Rezaei et al. (2020)	CVaR	RCPSP	Discrete scenarios	Heuristics	NSGA-II/MOVDO
Kasperski and Zieliński (2019)	CVaR and VaR	P1	Discrete scenarios	Approximation	MIP
Villarinho et al. (2021)	CVaR and VaR	Fm	Log-normal and weibull	Heuristic	BR-FF
Meloni and Pranzo (2023)	CVaR and VaR	RCPSP	Interval data	Approximation	Theoretical bounds

resource-constrained project scheduling problem (RCPSP) using specific classes of phase-type distributions to cope with non-exponential processing times and, grounding on this, optimal scheduling policies were derived based on continuous-time Markov chain models. Angius et al. (2021) proposed a general approach for modelling the execution of a network of activities with generally distributed processing times through a Markov chain and general phase-type distributions. All these works leverage the capability to estimate the distribution of an objective function (e.g., the makespan) to enable risk measures to address robustness.

This paper addresses a two-machine flow shop scheduling problem with generally distributed processing times to minimise the value-at-risk of the makespan. To our best knowledge, grounding on Table 1, this is the first work addressing this class of scheduling problem and objective function. Moreover, a novel optimisation framework is proposed based on a branch-and-bound algorithm and the associated bounding criteria. Heuristic approaches, fundamental to coping with larger instances, are not considered in this paper and will be the objective for future research.

3 Problem formulation

The problem under study is a two-machine permutation flow shop where a set of n jobs $\{a, b, ..., n\}$ are processed on two machines in series. The scheduling of the jobs is defined through a sequencing decision vector $\boldsymbol{x}$, with $x_{[k]}$ containing the indication of the job to be in the k-th position of the sequence. The decision variables in this vector, together with the characteristics of permutation flow shops, and hypothesising that each operation is started as soon as possible, completely define the schedule.

A vector of independent random variables $\boldsymbol{y} = \{p_{1a}, ..., p_{2n}\}$ models the random processing times of the jobs. These variables are governed by a phase-type probability measure $\mathbb{P}$ on $\mathbb{Y} = (0, +\infty)$, and are independent of the sequencing decisions in $\boldsymbol{x}$.

Due to this, also the makespan is a random variable depending on $\boldsymbol{x}$ and $\boldsymbol{y}$. Thus, the probability distribution of the makespan, $f_{C_{max}}(\boldsymbol{x}, \boldsymbol{y})$, depends on $\boldsymbol{x}$ and $\boldsymbol{y}$.

For a given schedule $\boldsymbol{x}$, the resulting cumulative density function (cdf) for the makespan is defined as:

$$F_{C_{max}}(\boldsymbol{x}, \boldsymbol{y}, \zeta) = P(f_{C_{max}}(\boldsymbol{x}, \boldsymbol{y}) \leq \zeta | \boldsymbol{x}) \tag{1}$$

The objective of the scheduling approach is the minimisation of the value-at-risk (VaR) of the makespan.

Definition 3.1. Given that C_{max} is a random variable and $\alpha \in (0, 1)$ is the desired risk level, the VaR_α of C_{max} is defined as,

$$VaR_\alpha(C_{max}) = min\{\zeta : F_{C_{max}}(\boldsymbol{x}, \boldsymbol{y}, \zeta) \geq 1 - \alpha\} \tag{2}$$

Thus, the VaR_α of C_{max}, associated with a schedule decision $\boldsymbol{x}$, denoted as $\zeta_\alpha(\boldsymbol{x}, \boldsymbol{y})$, is defined according to the following:

$$\zeta_\alpha(\boldsymbol{x}, \boldsymbol{y}) = min\{\zeta | F_{C_{max}}(\boldsymbol{x}, \boldsymbol{y}, \zeta) \geq 1 - \alpha\} \tag{3}$$

A summary of parameters and decision variables modelling the described scheduling problem are summarised in Table 2.

Given that the sequencing decision vector $\boldsymbol{x}$ does not depend on the values of the stochastic variables in $\boldsymbol{y}$, and C_{max} is a regular scheduling objective function (Pinedo 2016), $C_{max}(\boldsymbol{x}, \boldsymbol{y})$ is continuous and non-decreasing in $\boldsymbol{y}$. Thus, its value is non-decreasing when a new job is sequenced, or additional constraints are added to the problem. For this reason, the value of the objective function (VaR) of a partial schedule is a lower bound for the objective function of schedules containing that partial schedule (Ma and Wong 2010).

Based on these assumptions, in the next section, a branch and bound algorithm is proposed to search for a schedule that minimises the value-at-risk of the makespan.

Table 2 Parameters and variables.

Sets	
j	jobs, $j = a, ..., n$
i	machines, $i = 1, 2$
Parameters	
α	risk level
p_{ij}	processing time of job j on machine i
$\boldsymbol{y}$	array of processing time variables, $\boldsymbol{y} = \{p_{1a}, ..., p_{2n}\}$
Variables	
$\boldsymbol{x}$	sequencing decision array
$C_{max}(\boldsymbol{x}, \boldsymbol{y})$	makespan associated to sequence $\boldsymbol{x}$ and processing times $\boldsymbol{y}$
$f_{C_{max}}(\boldsymbol{x}, \boldsymbol{y})$	probability distribution function (pdf) of the makespan
$F_{C_{max}}(\boldsymbol{x}, \boldsymbol{y})$	cumulative density function (cdf) of the makespan
$\zeta_\alpha(\boldsymbol{x}, \boldsymbol{y})$	VaR_α of C_{max} associated with sequencing decisions $\boldsymbol{x}$

4 Solution Approach

The minimisation of the value-at-risk of the makespan is operated through a branch-and-bound algorithm grounding on the following building blocks:

1. *Initial bound and dominance rule.* Available heuristic approaches and theoretical results are exploited to identify an initial upper bound and possible dominance among the candidate solutions.

2. *Branching scheme and search strategy.* A branching scheme is defined to generate the nodes in the branching tree and the depth-first strategy is exploited to support the search for the optimal solution.

3. *Evaluation of the nodes.* A Markovian Activity Network is generated for each node of the tree, to estimate the distribution of the makespan. For the nodes representing partial solutions (schedules), a lower bound is obtained.

4.1 Initial bound and dominance rule

As described in Section 2, the rule proposed in Talwar (1967) provides an optimal schedule in a stochastic two-machine flow shop scheduling problem with exponentially distributed processing times and the minimisation of the expected makespan as the objective function. Furthermore, the proposed rule can be used as a heuristic approach for generally distributed processing times. In these cases, the rule provides reasonable results, although not optimal, especially when the processing time distributions of the whole set of activities are largely different in terms of their domain and do not overlap in a great deal, i.e., distributions' variances are small and/or means are widely not separated (Baker and Trietsch 2011). Hence, this rule is exploited to identify an initial value of the objective function and use it as the initial incumbent solution for the branch-and-bound algorithm (Emmons and Vairaktarakis 2012). Thus, according to Talwar (1967), for each job j, the expected value of the processing times on the two machines (1 and 2) is defined as $E(j_1)$ and $E(j_2)$ respectively. The initial solution can be determined by arranging the jobs according to the following rule where the arrow denotes a decreasing order:

$$S^* = \searrow \left(\frac{1}{E(j_1)} - \frac{1}{E(j_2)}\right) \tag{4}$$

Furthermore, existing theoretical results are exploited to identify possible dominance among the candidate solutions and reduce the solution space. According to (Chang and Yao 1993, Theoream 4.16-i), the following theorem applies.

Theorem 4.1. *Job j' should precede job j in order to minimise the makespan in the sense of stochastic ordering, if $j'_1 \leq_{lr} j_1$ and $j'_2 \geq_{lr} j_2$*

Here $j'_2 \geq_{lr} j_2$ denotes that the random variable j'_2 is larger than j_2 in the likelihood ratio sense, i.e., $P(j'_2 = t)/P(j_2 = t)$ is non-decreasing in t ($t =$ 0, 1, 2...) (Marshall et al. 1979). Notice that stochastic ordering also implies an ordering in terms of the value-at-risk (Bäuerle and Müller 2006). Thus, before starting the branch-and-bound algorithm, the theorem 4.1 above is used to identify, for each job j, a set of precedence constraints that must not be

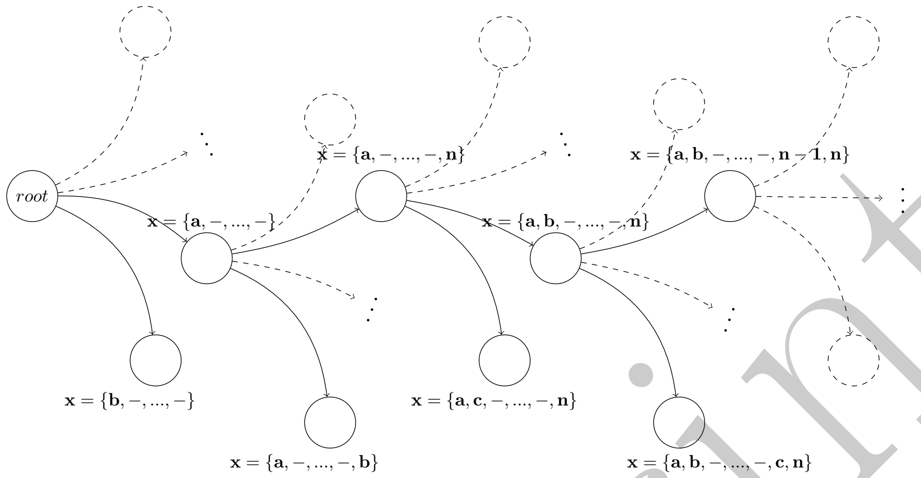

Figure 1 Branching Scheme.

violated. During the exploration of the branching tree, any solution (node) violating a precedence constraint in the list will be pruned.

4.2 Branching scheme

As described in Section 3, a solution to the addressed scheduling problem is defined by the decision variables in $\boldsymbol{x}$. Specifically, $\boldsymbol{x}_k$ denotes the index of the job in the k-th position of the sequence. To support the proposed branch-and-bound algorithm, a branching three is defined according to the following scheme. The tree starts from the root node, where no job has been sequenced. Then, jobs are alternatively sequenced at the beginning and end of the schedule. Hence, level 1 in the branching tree addresses the sequencing of the first job in the sequence, level 2 the last job in the sequence, level 3 the second job, level 4 the second last, and so on, until all the jobs are sequenced, see Figure 1. The depth-first strategy is used to explore the branch tree.

In comparison with a more traditional branching scheme, where the jobs are sequenced starting from the first to the last position in the schedule, the proposed scheme has the advantage of being less sensitive to the possible dominance between groups of activities on the two machines, aiming at guaranteeing more stable performance on different problem instances (Potts 1980; Gmys et al. 2020).

4.3 Evaluation of leaf nodes

Leaf nodes in the branching tree are associated with a complete schedule of the n jobs and entail the calculation of the value-at-risk α (VaRα) of the makespan of the schedule.

The makespan depends on the length of the critical path in the network of activities but, when stochastic processing times are considered, more than a single path has the probability of being critical (Dodin 1985). Hence, the estimation of the distribution of the makespan is intrinsically difficult to calculate (Dodin 1996). The evaluations of these nodes ground on a Markovian Activity Network approach to calculate the distribution of the makespan, under the hypothesis that processing times are modelled through phase-type distributions.

The addressed two-machine flow shop scheduling problem is represented with an Activity on Arc (AoA) network based on an acyclic-directed graph $G = (V, A)$ with a set of nodes V and a set of arcs A. Each arc in A represents an activity, while the nodes in V represent states, modelling the progress in the execution of the activities. At a given time t, an activity can only be in one of the following states (Kulkarni and Adlakha 1986):

- active: it is being executed and it is labeled with the name of the activity, e.g., (a_1)
- dormant: it has been completed, but there is an uncompleted activity incident on the same destination node, and it is labelled with the name of the activity with a star, e.g., (a_1^*)
- idle: it is neither active nor dormant, and it is not listed in the label of the state

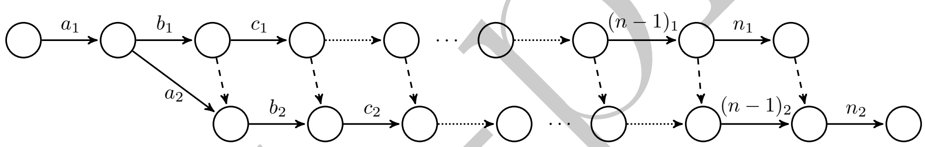

Figure 2 AoA activity network for a complete schedule.

Given the considered two-machine flow shop a complete sequencing of the jobs $\{a, b, ..., n\}$ can be modelled with the AoA network in Figure 2. Grounding on this network, the set of states modelling the execution of the network can be obtained. Starting from the state representing the processing of the first job is processing on machine 1, i.e., a_1, once activity a_1 is completed, it will transition to the state where both activities (first job on machine 2 and second job on machine 1) are being processed, i.e., (a_2, b_1), and then it may transition to one among two independent states: the first job completed on machine 2 and a second job in process on machine 1 (a_2^*, b_1); or machine 1 has completed the second job and has started the processing of the third job, while the first job is still being processed on machine 2 (a_2, c_1). This scheme is pursued until the absorbing state is reached, representing the complete processing of all the jobs on the two machines. Under the hypothesis of exponentially distributed processing times (Kulkarni and Adlakha 1986), the described scheme is a Continuous Time Markov Chain (CTMC). Moreover, for the problem under consideration, the structure of the state space and the associated transitions only depend on the number of jobs. Thus, for a given number of jobs, a general structure of the CTMC is derived and used in all the nodes without the need to generate it multiple times, see Figure 3.

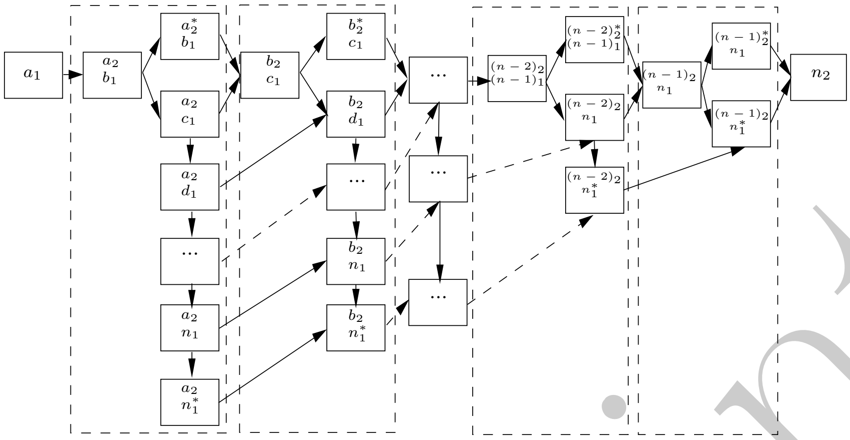

Figure 3 States generation scheme.

Based on the states in Figure 3, the infinitesimal generator of the CTMC representing the execution of the network of activities with phase-type distributions of the processing times can be obtained using a Kronecker algebra approach (Angius et al. 2021). Thus, the makespan of the network of activities is the time to absorption of the described CTMC, whose distribution can be calculated according to:

$$F(t) = 1 - \beta e^{Tt}\mathbf{1} \tag{5}$$

where β is the initial probability vector, T denotes the transition matrix not considering the absorbing state and $\mathbf{1}$ is an all-ones vector (Ross et al. 1996; Urgo 2014).

The quantile of this distribution corresponding to the VaR_α is obtained through a bisection method to find the root of:

$$1 - \alpha = 1 - \beta e^{\zeta * T}\mathbf{1} \tag{6}$$

where β, T and $\mathbf{1}$ are the same as in Eq. (5), α is the considered risk level, and ζ is the VaR_α value to be estimated.

A lower and upper bound for the VaR are also provided to the bisection method, to speed up the search for the root. The value of the VaR of the parent node is used as the lower bound value. In fact, since the VaR of makespan is a regular objective function and, since a child node is obtained from the parent by inserting a new job in the schedule, then the VaR of the child node can only be larger or equal to the VaR of its parent node (Pinedo 2016). The upper bound value, on the contrary, is assigned the value of the current best solution. Thus, if the bisection method fails to find the root, it must be larger than the current best solution, and the node can be pruned.

4.4 Evaluation of nodes representing partial schedules

For the nodes representing a partial schedule, i.e., with only a subset of the jobs sequenced, bounds have to be obtained for the considered objective function.

According to the branching scheme described in Section 4.2, s jobs have been already sequenced, whereof u are sequenced in the first positions of the schedule and v in the last positions (with $u+v=s$), while the sequencing of the remaining $n-s$ jobs is not decided yet. For the s assigned jobs, an approach similar to the one described in Section 4.3 is used to derive the initial and final segment of the associated network of activities (Figure 4). On the contrary, for the jobs to be sequenced, their processing times on the two machines are modelled by two activities (k_1 and k_2) whose processing times are the sum of the processing times of the operations belonging to unscheduled jobs. Due to this, possible precedence relations between these operations are omitted. The resulting AoA activity network is represented in Figure 4, and the same approach described in Section 4.3 is used to generate the state space, the CTMC considering phase-type distributed processing times and the estimation of the VaR.

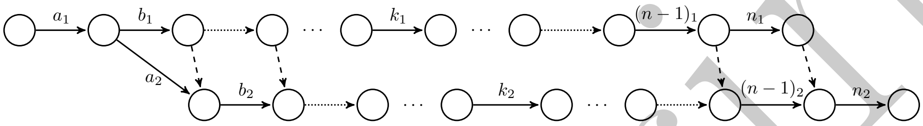

Figure 4 AoA activity network for a partial schedule.

In this case, since the VaR of the makespan is a regular objective function (see Section 4.3), its value can only remain the same or increase when a new job is scheduled and, hence, additional precedence relations are considered. Thus, the calculated VaR is a lower bound of the VaRs of all the nodes in the branch departing from the considered node (Ma and Wong 2010). Nodes associated with a partial schedule are pruned when the associated lower bound is larger than or equal to the best-known solution.

5 Computational results

The proposed branch-and-bound algorithm has been coded in C++, taking advantage of the BoB++ (Djerrah et al. 2006) and Eigen (Guennebaud et al. 2010) libraries. A set of experiments have been designed and executed to assess the performance and the effectiveness of the branch-and-bound approach. All the experiments have been carried out on a Windows 7 workstation with a 2.6 GHz Intel Xeon processor and 64 GB of RAM. A CPU time limit of 3600 seconds has been set for all the experiments.

5.1 Generation of the test instances

A set of test instances has been generated considering $n = 10$ and 20 jobs. The processing times of the jobs on the machines are modelled through phase-type distributions. These distributions have been randomly generated using the BuTools library(Horváth and Telek 2016) starting from the desired mean and the number of phases (Butools 2018). Specifically, the value of the mean is randomly sampled from three different uniform distributions with support [0,20], [30,50] and [60,80]. The number of phases of the distributions is randomly chosen between 1 and 4. The generation approach available in BuTools provides a randomly generated phase-type distribution, with no control on higher-order moments (e.g., variance and skewness).

Multiple experiments have been carried out considering different risk levels α (1%, 5%, 10% and 20%). A set of 20 test instances have been generated, for each combination of the number of jobs n and risk level α, for a total of 160 instances.

5.2 Performance of the algorithm

The first aim of the experiments is to assess the performance of the proposed approach in terms of the time needed to find the optimal solution, and how this scales as the dimension of the problem increases.

Considering all the 160 instances, the results are reported in Table 3, showing the performance of the branch-and-bound algorithm in terms of solution time and the number of evaluated nodes. The average, minimum, maximum values and standard deviation are reported for each combination of the number of jobs and risk level.

With respect to the experiments carried out, the average solution time is 1029.5 s, ranging from a minimum of 1.4 s to the imposed time limit of 3600 s. Considering the results in relation to the number of jobs in the test instances, we see that, in the cases with 10 jobs, the algorithm can find the optimal solution in an average time of 257.2 s, ranging from a minimum of 1.4 s to a maximum of 1205.2 s. To find the optimal schedule, an average of 5785 nodes had to be evaluated, with respect to the whole branching tree containing $6.2 * 10^6$ nodes, which corresponds to 0.09% of them. When considering 20-job instances, the solution time predictably increases, on average 1801.8 s are needed to solve an instance to optimality, ranging from a minimum of 28.4 s to a maximum of 3600 s, and an average of 11844 nodes of the whole branching tree containing $4.2 * 10^{18}$ need to be analysed.

Table 3 Results.

number of jobs	risk level	solution time(s)				evaluated nodes			
		mean	min	max	SD	mean	min	max	SD
10	1	579.9	12.7	1205.2	436.3	12060	173	19967	6012
	5	224.7	4.4	633.1	218.4	5295	81	16569	5393
	10	105.7	1.4	400.0	119.3	2534	54	7954	2410
	20	118.4	3.1	346.6	95.5	3523	90	7979	2503
	ALL	257.2	1.4	1205.2	316.6	5785	54	19967	5737
20	1	3296.5	1449.7	3600.0	660.0	22483	154	67708	16365
	5	1357.1	31.1	3600	1339.6	9287	115	50371	12200
	10	1285.3	41.3	3469.6	1089.2	9003	135	45114	10792
	20	1254.3	28.4	3321.8	1142.2	6605	95	29842	7266
	ALL	1801.8	28.4	3600	1375.3	11844	95	67708	10090
ALL		1029.5	1.4	3600	1260.9	8814	54	67708	8896

For the subset of 20-job instances (36.7% of the whole set) that could not be solved to optimal, the GAP between the current best solution and the lower

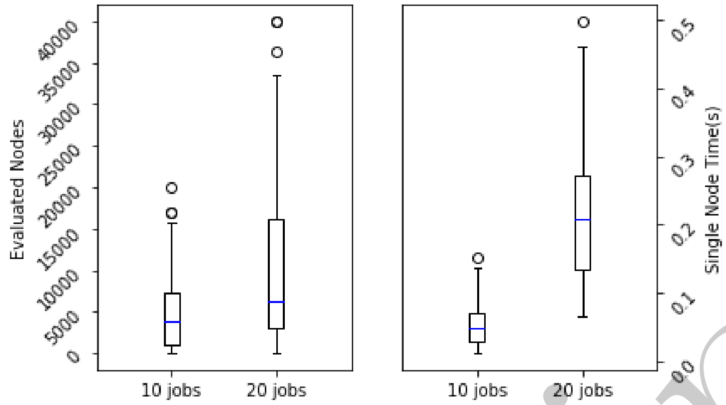

Figure 5 Evaluated nodes and single node evaluation time plot.

bound was investigated, resulting in a value ranging from 0.0% to 2.1%, with an average of 1.7%.

With the aim of further investigating the performance of the approach in terms of solution time, we exploited additional investigations. Different plots are presented to examine how various factors influence the performance of the algorithm.

The number of jobs was clearly identified as one of the potentially impacting factors, as shown in Figure 5, due to two main reasons. Firstly, the branching tree will be larger as the number of jobs increases, entailing a potentially higher number of nodes to be evaluated during the search. Furthermore, while sequencing a higher number of jobs, also the average time needed to evaluate a node is expected to increase, due to the need of coping with larger infinitesimal generator matrices in the evaluation of both partial and complete schedules. Finally, the plots in Figure 5 also show the presence of outliers. Thus, the specific characteristics of the scheduling instances also have a possible impact on the performance of the solution approach.

Another preliminary analysis was carried out to investigate the possible impact of the risk level. The plots of the solution time and the number of evaluated nodes, detailed in terms of the different risk levels, are presented in Figure 6. It emerges that the risk level potentially impacts the solution time, with values of α equal to 5% and 1% seemingly causing higher solution times and more nodes to be evaluated. This impact seems evident for 10-job instances only. However, it must be noticed that, while solving 20-job instances with a 1% risk level, most (about 80%) reached the time limit and were not included in this analysis and the related plots.

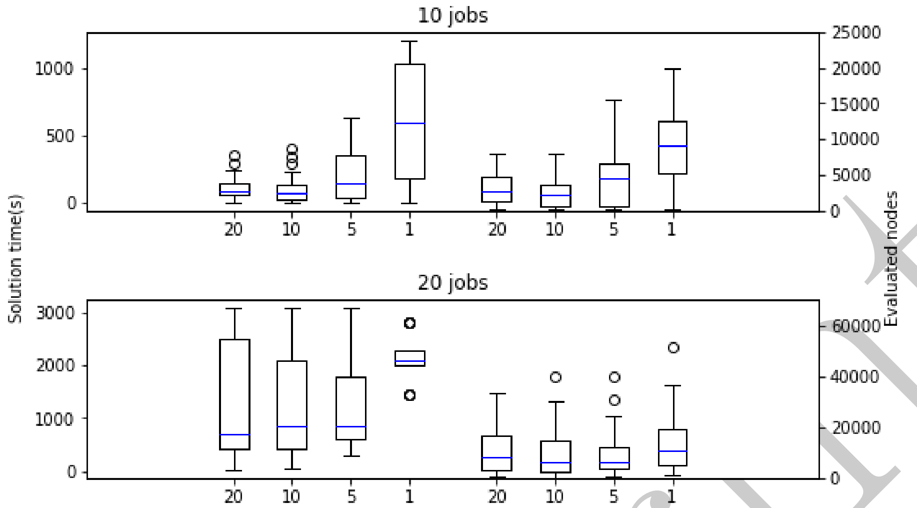

Figure 6 Box plot of solution time and the number of evaluated nodes according to the risk level considered (excluding experiments that reached the time limit).

To support the observation emerged from the plots in Figures 5 and 6, an ANOVA analysis has been operated. However, the hypothesis on normal residuals was not respected, even operating proper transformations on the data. Thus, Mood's median non-parametric tests have been used to analyse the results. The results of the tests are reported in Table 4. The first two rows relates to the investigation of the impact of the number of jobs and the risk level on the solution time. The number of jobs factor yields a statistically significant result, as anticipated, while the risk level factor does not. Thus, the number of jobs clearly impacts the solution time (solution time *vs* number of jobs), but the dependence on the risk level (solution time *vs* risk level), although the reasoning on Figure 6 could not be demonstrated through the analysis of the data. An additional set of tests has been done to investigate the impact of the number of jobs on the number of nodes in the branching tree to be evaluated to reach the optimal solution (evaluated nodes *vs* number of jobs). The results of the test did not provide evidence of this dependency. Thus, an additional test was operated to check the possible dependency on time to evaluate a single node that, on the contrary, provided statistical significance. Thus, the number of jobs clearly impacts the solution time, and this is due to the higher computational effort required to estimate the distribution of the makespan with a higher number of jobs, requiring a larger infinitesimal generator matrix.

A similar analysis was carried out to investigate the dependency on the specific risk level. Still, the test related to the number of nodes (evaluated nodes *vs* risk level) and the time to solve a node (evaluation time per node *vs* risk level) did not provide statistical evidence. Nevertheless, as the diagrams in Figure 6 provided different evidence, it has been hypothesised that the lack of statistical significance could be due to the small number of 20-job instances since most of the related experiments reached the time limit and were excluded from the analysis, especially with a risk level equal to 1%. Thus, tests were carried out on 10-job instances only (last two rows in Table 4), resulting in the evidence that, at least within this subset of experiments, the dependency on the risk level was clearly significant for both the solution time and the number of evaluated nodes.

Thus, the experiments demonstrated that the time to solve an instance depends on the number of jobs to be scheduled. This was clearly an expected result, but the analysis confirmed that the main motivation for the longer solution time is due to the time needed to estimate the distribution of the makespan for a larger Markov chain. On the contrary, the effectiveness of the proposed initial solution and bounds does not change with the number of jobs, as the number of evaluated nodes in the search does not depend on this. In contrast, the selected risk level impacts the solution time. This was evident for 10 job instances only. Nevertheless, if the experiments had been carried out without a time limit, this dependence could probably have also emerged for 20-job instances. This seems to be attributed to the higher number of nodes that need to be evaluated, especially for a risk level equal to 1%. This is probably because, when the distributions of the different schedules become very close, as is expected in the right-tail, the algorithm requires more nodes to be evaluated to find the optimal solution.

Table 4 Nonparametric pairwise comparison tests results.

Mood's Median Test	**DF**	χ^2	**p-value**
solution time *vs* number of jobs	1	59.64	<0.0001
solution time *vs* risk level	3	4.77	0.19
evaluated nodes *vs* number of jobs	1	0.34	0.56
evaluated nodes *vs* risk level	3	11.09	0.011
evaluation time per node *vs* number of jobs	1	65.01	<0.0001
evaluation time per node *vs* risk level	3	1.98	0.58
solution time *vs* risk level (10-job instances)	3	19.60	<0.0001
evaluated nodes *vs* risk level (10-job instances)	3	22.00	<0.0001

An additional investigation has been executed to assess the efficiency of the initial solution and the dominance rule. Concerning the initial solution and bound, while each instance was solved, the number of times the best solution was updated, with respect to the initial one, has been collected. Figure 7(a) shows that the number of times this solution is updated is always less than 10. Thus the initial solution based on the heuristic Talwar rule speeds up the branch-and-bound algorithm in the search for the optimal solution. Nevertheless, the contribution of the branch-and-bound algorithm is also relevant, in fact, it can improve the initial solution of 8.1% on average, with a minimum and maximum improvement of the value of the objective function equal to 0.0% and 12.3% respectively.

Referring to the effectiveness of the proposed dominance rule (Section 4.1), listing precedence constraints causing dominance among the solutions, Figure 7(b) shows the fraction of nodes pruned grounding the dominance rule with respect to the overall number of evaluated nodes. In the experiments carried out, 15.0% of the nodes, on average, are pruned according to the dominance rule during the search, with a minimum and maximum fraction of 1.2% and 59.7% respectively, proving the effectiveness of the dominance rule.

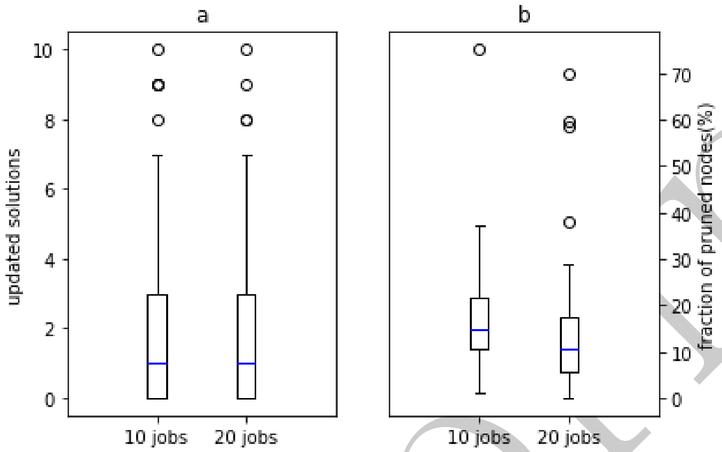

Figure 7 Effectiveness of the proposed initial solution (a) and dominance rule (b).

As mentioned in the preliminary analysis of the results, while solving 20-job instances, the branch-and-bound algorithm could not find the optimal solution within the given time limit for about 80% of the experiments. An additional investigation has been carried out to assess the quality of the incumbent solution obtained, although not optimal. As the branch-and-bound algorithm is stopped, the list of evaluated nodes in the branching tree is analysed to estimate a global lower bound for the objective function (LB_G). The difference between the initial solution value S_0 and LB_G provides an estimation of the gap to be filled to reach optimality. An indicator $\Delta\%$ is defined in Eq. 7 to evaluate the effectiveness of the branch-and-bound algorithm to improve the initial solution S_0 and to estimate the gap to the optimal solution.

$$\Delta\% = \frac{S_0 - BB_{inc}}{S_0 - LB_G} \tag{7}$$

Table 5 Estimated gap for 30- and 50-job instances.

job number	Δ%		
	min	max	mean
30	22.1	87.6	46.3
50	19.6	78.2	45.4

The results in Table 5 shows that the proposed branch-and-bound algorithm can improve the initial solution by 45.5% towards the global lower bound, demonstrating the effectiveness of the proposed approach even on larger instances where the capability to complete the optimisation cannot be guaranteed.

5.3 Comparison with alternative approaches

To evaluate the benefits of a scheduling approach based on Markovian Activity Networks and the minimization of the value-at-risk, alternative robust scheduling approaches have been implemented and compared both from the accuracy and effectiveness points of view. Specifically, scheduling approaches based on expectation-variance analysis (Sarin et al. 2010) and maximum regret minimization (Kouvelis et al. 2000) have been considered for the comparison.

Approaches based on the expectation-variance analysis use a finite mixture model to approximate a generally distributed processing time in terms of a convex combination of normal distributions, providing, in many cases, very good results (Sarin et al. 2010).

The mean and variance of the makespan are then computed accordingly. In a two-machine flow shop scheduling problem, many paths exist, starting from the first sequenced activity on the first machine to the last one on the second machine. All these paths share at least a couple of activities. Thus the distributions of their completion times are correlated. Thus, the computation of the mean and variance of the makespan is operated through an approximation based on the Clark equation (Clark 1961).

Due to this approximation, a first comparison, focused on accuracy, is operated between the proposed Markovian Activity Network (MAN) approach and the one based on the expectation-variance (EV) one described above, with the aim at estimating the accuracy in the calculation of the mean and variance of the makespan. A representative 10-job instance (see Appendix A) is chosen among the ones described in Section 5.1. Grounding on this instance, ten randomly chosen schedules are defined. Thus the mean and variance value of the makespan is calculated using the MAN and EV approaches. The exact value of the mean and variance is estimated using a Monte Carlo simulation approach with 10^7 samples. The relative deviations of the two approaches from the exact value are presented in Table 6, showing that the EV approach can entail a significant error both in the estimation of the mean and variance of the makespan. In contrast, the MAN approach demonstrates a higher accuracy.

Table 6 Accuracy in the estimation of the mean and variance of the makespan.

		relative error	
		expectation-variance analysis	Markovian activity network
mean	avg	5.6%	0.03%
	min	4.9%	0.02%
	max	7.5%	0.04%
variance	avg	5.3%	0.07%
	min	4.7%	0.05%
	max	6.9%	0.08%

A different type of analysis has been carried out to compare the effectiveness of different objective functions to obtain a robust schedule, i.e., value-at-risk, expectation-variance, and maximum regret.

Starting from the 10-job representative instance described above, different schedules are defined, looking for those showing different behaviours in terms of VaR and mean-variance. Among these, two different schedules $\boldsymbol{x}_1 = \{1, 10, 7, 4, 2, 3, 6, 9, 8, 5\}$ and $\boldsymbol{x}_2 = \{10, 1, 7, 4, 2, 3, 6, 5, 8, 9\}$ are observed with the difference in terms of mean and variance is 0.03% and 0.6% respectively. In comparison, the difference in terms of $VaR_{10\%}$ is 2.1% (Figure 8). Hence, the two schedules would be very similar for the approach based on expectation-variance while, if the minimisation of the VaR is operated, one of them is dominating the other one. This difference is mostly because the density function of the distributions of the makespan for the two schedules have different tails on the right side (Figure 8). Thus, the use of risk measures like the value-at-risk provides the capability of considering the right tail of the distribution of the makespan to mitigate the impact of worst cases, in line with Sarin et al. (2014).

The comparison with an approach minimising the maximum regret also refers to this representative instance and grounds on three measures, i.e., the maximum regret, the value-at-risk, and the maximum possible value of the makespan. Since the support of the distribution is not bounded, the 0.01% and 99.99% quantiles are considered to obtain the latter. For this comparison, we take into consideration two alternative schedules: $\boldsymbol{x}_{VaR} = \{1, 10, 7, 4, 2, 3, 6, 9, 8, 5\}$ is the optimal schedule minimising the VaR of the makespan; $\boldsymbol{x}_{maxRegr} = \{1, 10, 2, 6, 8, 5, 3, 7, 9, 4\}$ is the optimal schedule minimising the maximum regret of the makespan.

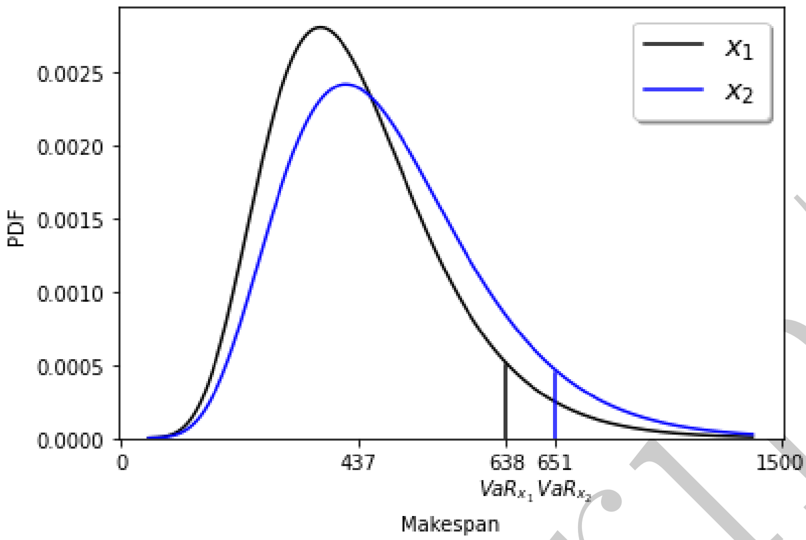

Figure 8 Probability density function of the makespan for $\boldsymbol{x}_1$ (black) and $\boldsymbol{x}_2$ (blue) and respective values for the $VaR_{10\%}$.

The results of this comparison are reported in Table 7. They show that both the minimisation of the maximum regret and the value-at-risk provide reasonably robust schedules. Nevertheless, the approach based on the minimisation of the VaR, besides minimising this value, also guarantees significantly better protection against the worst case, i.e., the maximum possible makespan (12% better than the maximum regret approach).

Table 7 Comparison between the schedules minimizing the VaR and maximum regret.

	VaR	max regret	max value
$\boldsymbol{x}_{VaR}$	638.4	124.4	1541.94
$\boldsymbol{x}_{maxRegr}$	708.9	112.8	1722.33
$\Delta\%$	+11.0%	-10.2%	+12%

Hence, to better investigate the comparison between the maximum regret and value-at-risk minimisation approach, a complete analysis has been operated over all the problem instances.

The minimisation of the maximum regret grounds on a simplified model of the processing times of the jobs, only considering the extreme values (minimum and maximum). Thus, it is possible to solve 10-job instances in less than 5 s. However, for more than 90% of the instances with 20 jobs, it was impossible to get the optimal schedule in less than 30 minutes. This aligns with the results in Kouvelis et al. (2000), providing results for instances with a maximum of 15 jobs. Thus, considering the whole picture, the proposed VaR approach can solve the generated instances in a reasonable time.

For each instance, the optimal schedule minimising the maximum regret of the makespan ($s_{maxRegr}$) is obtained and compared with the schedule obtained through the branch-and-bound approach (s_{VaR}), minimising the VaR. Based on the experiments, the maximum regret value of s_{VaR} is about 9.0% larger than that of $s_{maxRegr}$ on average. In contrast, the VaR of s_{VaR} is on average 7.4% smaller than that of $s_{maxRegr}$, demonstrating that the schedule solution obtained from the proposed VaR approach is performing worse for the maximum regret criterion.

Besides these expected absolute results, the probability for schedule ${}_{VaR}$ to incur in a maximum value of the makespan higher than $s_{maxRegr}$ is also calculated. Thus, given max_{VaR} the maximum possible value of the makespan for s_{VaR} and $max_{maxRegr}$ the maximum possible value for $s_{maxRegr}$, the probability of having $max_{VaR} \geq max_{maxRegr}$ has been estimated, grounding on the distribution of the makespan for the two schedules. The results of this assessment are that

$$P(max_{VaR} \geq max_{maxRegr}) \approx 0 \tag{8}$$

which means that the probability for the schedule obtained with the proposed approach to cause a makespan larger than the one obtained with the minimax regret approach is negligible. Thus, the VaR approach can provide a reasonable and less cautious solution to balance the risk associated with extreme scenarios. The detailed results of the experiments are not reported since, for almost all instances, the probability defined in Eq. 8 is lower than 0.01%.

6 Managerial insights

Planning and scheduling in real manufacturing environments entail the need to cope with multiple sources of uncertainty whereof processing times are the most significant (see Section 1). Decision-makers are requested to balance the pursuit of production performances and, at the same time, mitigate the risks associated with uncertainty. The proposed scheduling approach supports this trade-off by devising robust schedules to minimise the value-at-risk of the makespan, i.e., a well-established measure of risk.

This requires the estimation of the distribution of the objective function that, especially together with the need to cope with generally-distributed processing times, is a computation-critical problem. The proposed approach provides a higher accuracy in estimating the selected risk measures, compared with other methods that may be simpler to implement and faster to run, but could lead to significant errors and non-optimal schedule decisions (Section 5.3). Thus it provides managers with reliable information to ground their decisions. Pursuing accuracy clearly entails a higher computational load, constituting a significant limitation while coping with large scheduling problems.

Nevertheless, from our point of view, this does not significantly hinder the benefits of using the proposed approach. In fact, in real manufacturing plants, small-/medium-size problems are extremely common. Furthermore, scheduling on long time horizons is often operated at an aggregate level (Kusiak 1989; Buxey 1989; Liu and Urgo 2022), i.e., considering the aggregation of groups of operations into a macro activity. Thus, providing the possibility to work with longer scheduling horizons while being limited in the number of jobs.

7 Conclusions

In this paper, a branch-and-bound algorithm has been proposed and demonstrated for the two-machine stochastic flow shop scheduling problem to minimise the value-at-risk (VaR) of the makespan. The aim is to find a robust schedule capable of protecting against the occurrence of unfavourable events using the VaR as a risk measure guiding the search for robustness. A Markovian Activity Network (MAN) approach has been adopted to estimate the distribution of the makespan, modelling the processing of the jobs through a Markov chain, and exploiting phase type distributions to cope with realistic distributions in a wide range of application areas (e.g., industrial processes).

Further developments will focus on improving computation performance addressing the following aspects:

1. taking advantage of a modular definition of the infinitesimal generator matrix (Angius et al. 2021) to reuse the estimation operated on partial schedules within the exploration of the same branching tree.

2. exploit alternative approaches to operate the exponential matrix and derive the distribution of the makespan (e.g., Krylov (Sidje 1998), CAM (Al-Mohy and Higham 2011)).

3. improve the root finding approach to decrease the number of iterations when estimating the Value-at-Risk of the distribution of the makespan.

4. develop the integration of heuristic algorithms to cope with larger instances. Examples are the Iterated Greedy(IG) heuristic (Ruiz and Stützle 2007), which has been demonstrated to be effective in the deterministic flow shop scheduling problem.

Furthermore, the performance of the proposed approach also demonstrated dependency on the specific scheduling instance to be solved. This dependence is well known in the deterministic version of the two-machine flow shop scheduling problem and it grounds on the possible presence of dominance among the distributions of the processing times on the two machines (Emmons and Vairaktarakis 2012). These dominance criteria do not have a stochastic version; thus, this area should be further investigated to support improving the performance of solution approaches.

Data availability statement

The data that support the findings of this study are openly available in Figshare https://figshare.com/articles/dataset/DistributionDataSet/22179829

Notes on contributors

Lei Liu is currently an early-stage researcher holding a European Marie Curie Fellowship and a third-year PhD candidate at the Mechanical Engineering Department of Politecnico di Milano. He received a B.S. degree (2013) from Northeast Forestry University and a M.S. (2017) from Tsinghua University, China. His research of interest lies on stochastic decision-making with risk measures and Markov network analysis, applications on machine scheduling and project management problems.

Marcello Urgo is an Assistant Professor in the Department of Mechanical Engineering at Politecnico di Milano. He holds a PhD in Manufacturing Systems and Technologies focused on robust production planning and scheduling. His main research interests are in the field of (1) robust production planning and scheduling, (2) digital twin approaches for the design and management of manufacturing systems, (3) modelling and monitoring of manual manufacturing processes, and (4) artificial intelligence in manufacturing. He is the author of more than 70 peer-reviewed papers published in scientific journals and presented at international conferences. He is a member of the Italian Association for Manufacturing Technology (AITEM), an associate member of the CIRP - International Academy for Production Engineering, member of the EURO working group on project management and scheduling.

References

Al-Mohy, A. H. and Higham, N. J. (2011). Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM Journal on Scientific Computing*, 33(2):488–511.

Alfieri, A., Tolio, T., and Urgo, M. (2011). A project scheduling approach to production planning with feeding precedence relations. *International Journal of Production Research*, 49(4):995–1020.

Alfieri, A., Tolio, T., and Urgo, M. (2012). A two-stage stochastic programming project scheduling approach to production planning. *The International Journal of Advanced Manufacturing Technology*, 62(1):279–290.

Allahverdi, A. and Aydilek, H. (2010). Heuristics for the two-machine flowshop scheduling problem to minimise makespan with bounded processing times. *International Journal of Production Research*, 48(21):6367–6385.

Angius, A., Horváth, A., and Urgo, M. (2021). A kronecker algebra formulation for markov activity networks with phase-type distributions. *Mathematics*, 9(12):1404.

Atakan, S., Bülbül, K., and Noyan, N. (2017). Minimizing value-at-risk in single-machine scheduling. *Annals of Operations Research*, 248(1-2):25–73.

Averbakh, I. and Berman, O. (1997). Minimax regret p-center location on a network with demand uncertainty. *Location Science*, 5(4):247–254.

Aytug, H., Lawley, M. A., McKay, K., Mohan, S., and Uzsoy, R. (2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161(1):86–110.

Baker, K. R. and Trietsch, D. (2011). Three heuristic procedures for the stochastic, two-machine flow shop problem. *Journal of Scheduling*, 14(5):445–454.

Bäuerle, N. and Müller, A. (2006). Stochastic orders and risk measures: consistency and bounds. *Insurance: Mathematics and Economics*, 38(1):132–148.

Benmansour, R., Allaoui, H., and Artiba, A. (2012). Stochastic single machine scheduling with random common due date. *International Journal of Production Research*, 50(13):3560–3571.

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

Butools (2018). Butools 2.0. `http://webspn.hit.bme.hu/~telek/tools/butools/doc/RandomPH.html#butools.ph.RandomPH`.

Buxey, G. (1989). Production scheduling: Practice and theory. *European Journal of Operational Research*, 39(1):17–31.

Chang, C.-S. and Yao, D. D. (1993). Rearrangement, majorization and stochastic scheduling. *Mathematics of Operations Research*, 18(3):658–684.

Chang, Z., Song, S., Zhang, Y., Ding, J.-Y., Zhang, R., and Chiong, R. (2017). Distributionally robust single machine scheduling with risk aversion. *European Journal of Operational Research*, 256(1):261–274.

Clark, C. E. (1961). The greatest of a finite set of random variables. *Operations Research*, 9(2):145–162.

Creemers, S. (2015). Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. *Journal of Scheduling*, 18(3):263–273.

Creemers, S. (2018). Maximizing the expected net present value of a project with phase-type distributed activity durations: An efficient globally optimal solution procedure. *European Journal of Operational Research*, 267(1):16–22.

Cunningham, A. A. and Dutta, S. K. (1973). Scheduling jobs with exponentially distributed processing times, on two machines of a flow shop. *Naval Research Logistics Quarterly*, 20(1):69–81.

De, P., Ghosh, J. B., and Wells, C. E. (1992). Expectation-variance analyss of job sequences under processing time uncertainty. *International Journal of Production Economics*, 28(3):289–297.

Djerrah, A., Le Cun, B., Cung, V.-D., and Roucairol, C. (2006). Bob++: Framework for solving optimization problems with branch-and-bound methods. In *2006 15th IEEE International Conference on High Performance Distributed Computing*, pages 369–370. IEEE.

Dodin, B. (1985). Bounding the project completion time distribution in pert networks. *Operations Research*, 33(4):862–881.

Dodin, B. (1996). Determining the optimal sequences and the distributional properties of their completion times in stochastic flow shops. *Computers & Operations Research*, 23(9):829–843.

Emmons, H. and Vairaktarakis, G. (2012). *Flow shop scheduling: theoretical results, algorithms, and applications*, volume 182. Springer Science & Business Media.

Fathollahi-Fard, A. M., Woodward, L., and Akhrif, O. (2021). Sustainable distributed permutation flow-shop scheduling model based on a triple bottom line concept. *Journal of Industrial Information Integration*, 24:100233.

Filippi, C., Guastaroba, G., and Speranza, M. G. (2020). Conditional value-at-risk beyond finance: a survey. *International Transactions in Operational Research*, 27(3):1277–1319.

Gholizadeh, H., Fazlollahtabar, H., Fathollahi-Fard, A. M., and Dulebenets, M. A. (2021). Preventive maintenance for the flexible flowshop scheduling under uncertainty: A waste-to-energy system. *Environmental Science and Pollution Research*, pages 1–20.

Gmys, J., Mezmaz, M., Melab, N., and Tuyttens, D. (2020). A computationally efficient branch-and-bound algorithm for the permutation flow-shop scheduling problem. *European Journal of Operational Research*, 284(3):814–833.

Gourgand, M., Grangeon, N., and Norre, S. (2000). A review of the static stochastic flow-shop scheduling problem. *Journal of Decision Systems*, 9(2):1–31.

Guennebaud, G., Jacob, B., et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

Horváth, G. and Telek, M. (2016). Butools 2: a rich toolbox for markovian performance evaluation. In *VALUETOOLS*.

Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1):61–68.

Kasperski, A., Kurpisz, A., and Zieliński, P. (2012). Approximating a two-machine flow shop scheduling under discrete scenario uncertainty. *European Journal of Operational Research*, 217(1):36–43.

Kasperski, A. and Zieliński, P. (2019). Risk-averse single machine scheduling: complexity and approximation. *Journal of Scheduling*, 22(5):567–580.

Kouvelis, P., Daniels, R. L., and Vairaktarakis, G. (2000). Robust scheduling of a two-machine flow shop with uncertain processing times. *IIE Transactions*, 32(5):421–432.

Kulkarni, V. G. and Adlakha, V. (1986). Markov and markov-regenerative pert networks. *Operations Research*, 34(5):769–781.

Kusiak, A. (1989). Aggregate scheduling of a flexible machining and assembly system. *IEEE Transactions on Robotics and Automation*, 5(4):451–459.

Levorato, M., Figueiredo, R., and Frota, Y. (2022). Exact solutions for the two-machine robust flow shop with budgeted uncertainty. *European Journal of Operational Research*, 300(1):46–57.

Liu, L. and Urgo, M. (2022). A robust scheduling framework for re-manufacturing activities of turbine blades. *Applied Sciences*, 12(6):3034.

Liu, M., Liu, X., Chu, F., Zheng, F., and Chu, C. (2019). Service-oriented robust parallel machine scheduling. *International Journal of Production Research*, 57(12):3814–3830.

Liu, X., Chu, F., Zheng, F., Chu, C., and Liu, M. (2021). Parallel machine scheduling with stochastic release times and processing times. *International Journal of Production Research*, 59(20):6327–6346.

Ma, C. and Wong, W.-K. (2010). Stochastic dominance and risk measure: A decision-theoretic foundation for var and c-var. *European Journal of Operational Research*, 207(2):927–935.

Marshall, A. W., Olkin, I., and Arnold, B. C. (1979). *Inequalities: Theory of Majorization and Its Applications*, volume 143. Springer.

Meloni, C. and Pranzo, M. (2020). Expected shortfall for the makespan in activity networks under imperfect information. *Flexible Services and Manufacturing Journal*, 32(3):668–692.

Meloni, C. and Pranzo, M. (2023). Evaluation of the quantiles and superquantiles of the makespan in interval valued activity networks. *Computers & Operations Research*, 151:106098.

Neuts, M. F. (1994). *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Corporation.

OâĂŹCinneide, C. A. (1990). Characterization of phase-type distributions. *Stochastic Models*, 6(1):1–57.

Pinedo, M. (2016). Scheduling: Theory, algorithms, and systems fifth. *Cham: Springer International Publishing.*

Potts, C. (1980). An adaptive branching rule for the permutation flow-shop problem. *European Journal of Operational Research*, 5(1):19–25.

Radke, A. M., Tolio, T., Tseng, M. M., and Urgo, M. (2013). A risk management-based evaluation of inventory allocations for make-to-order production. *CIRP Annals*, 62(1):459–462.

Rezaei, F., Najafi, A. A., and Ramezanian, R. (2020). Mean-conditional value at risk model for the stochastic project scheduling problem. *Computers & Industrial Engineering*, 142:106356.

Ross, S. M., Kelly, J. J., Sullivan, R. J., Perry, W. J., Mercer, D., Davis, R. M., Washburn, T. D., Sager, E. V., Boyce, J. B., and Bristow, V. L. (1996). *Stochastic Processes*, volume 2. Wiley New York.

Ruiz, R. and Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049.

Sarin, S. C., Nagarajan, B., and Liao, L. (2010). *Stochastic scheduling: expectation-variance analysis of a schedule*. Cambridge University Press.

Sarin, S. C., Sherali, H. D., and Liao, L. (2014). Minimizing conditional-value-at-risk for stochastic scheduling problems. *Journal of Scheduling*, 17(1):5–15.

Savage, L. J. (1951). The theory of statistical decision. *Journal of the American Statistical Association*, 46(253):55–67.

Seif, J., Dehghanimohammadabadi, M., and Yu, A. J. (2020). Integrated preventive maintenance and flow shop scheduling under uncertainty. *Flexible Services and Manufacturing Journal*, 32:852–887.

Sidje, R. B. (1998). Expokit: A software package for computing matrix exponentials. *ACM Transactions on Mathematical Software (TOMS)*, 24(1):130–156.

Talwar, P. (1967). A note on sequencing problems with uncertain job times. *Journal of the Operations Research Society of Japan*, 9(3-4):93–97.

Tetenov, A. (2012). Statistical treatment choice based on asymmetric minimax regret criteria. *Journal of Econometrics*, 166(1):157–165.

Tolio, T. and Urgo, M. (2013). Design of flexible transfer lines: A case-based reconfiguration cost assessment. *Journal of Manufacturing Systems*, 32(2):325–334.

Tolio, T., Urgo, M., and Váncza, J. (2011). Robust production control against propagation of disruptions. *CIRP Annals*, 60(1):489–492.

Urgo, M. (2014). *Stochastic Scheduling with General Distributed Activity Durations Using Markov Activity Networks and Phase-Type Distributions*. Nova.

Urgo, M. (2019). A branch-and-bound approach to schedule a no-wait flow shop to minimize the cvar of the residual work content. *Computers & Industrial Engineering*, 129:67–75.

Urgo, M. and Váncza, J. (2019). A branch-and-bound approach for the single machine maximum lateness stochastic scheduling problem to minimize the value-at-risk. *Flexible Services and Manufacturing Journal*, 31(2):472–496.

Villarinho, P. A., Panadero, J., Pessoa, L. S., Juan, A. A., and Oliveira, F. L. C. (2021). A simheuristic algorithm for the stochastic permutation flow-shop problem with delivery dates and cumulative payoffs. *International Transactions in Operational Research*, 28(2):716–737.

Wiesemann, W. (2010). *Optimisation of temporal networks under uncertainty*. PhD thesis, Imperial College London.

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.

Appendix A

For the representative 10-job instance in Section 5.3, the phase-type processing time distributions of the jobs on each machine are reported in Table 8. Each phase-type distribution is denoted by an initial vector $\boldsymbol{\beta}$ and a matrix T (Neuts 1994).

Table 8 Distributions of the processing times for the instance in Section 5.3.

job j		distribution p_{j1}	distribution p_{j2}
1	β	[1 0]	[1]
	T	$\begin{bmatrix} -3.655 & 3.655 \\ 6.394 & -7.77 \end{bmatrix}$	[-0.018]
2	β	[1]	[1 0]
	T	[-0.13]	$\begin{bmatrix} -0.053 & 0.044 \\ 0 & -0.039 \end{bmatrix}$
3	β	[1]	[1 0]
	T	[-0.084]	$\begin{bmatrix} -0.03 & 0.012 \\ 0.032 & -0.032 \end{bmatrix}$
4	β	[1]	[1 0]
	T	[-0.14]	$\begin{bmatrix} -0.062 & 0.024 \\ 0.02 & -0.02 \end{bmatrix}$
5	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.042 & 0.018 \\ 0 & -0.011 \end{bmatrix}$	[-0.21]
6	β	[1,0]	[1]
	T	$\begin{bmatrix} -0.035 & 0.009 \\ 0.021 & -0.021 \end{bmatrix}$	[-0.016]
7	β	[1]	[1 0]
	T	[-0.166]	$\begin{bmatrix} -0.185 & 0.185 \\ 0.485 & -0.572 \end{bmatrix}$
8	β	[1 0 0 0]	[1]
	T	$\begin{bmatrix} -0.128 & 0.044 & 0.012 & 0.03 \\ 0.061 & -0.061 & 0 & 0 \\ 0.051 & 0 & -0.051 & 0 \\ 0 & 0 & 0.088 & -0.088 \end{bmatrix}$	[-0.081]
9	β	[1]	[1 0 0 0]
	T	[-0.021]	$\begin{bmatrix} -0.854 & 0.386 & 0.076 & 0.164 \\ 0 & -0.37 & 0 & 0.37 \\ 0 & 0.277 & -0.277 & 0 \\ 0.35 & 0 & 0 & -0.35 \end{bmatrix}$
10	β	[1]	[1 0 0 0]
	T	[-0.24]	$\begin{bmatrix} -0.42 & 0.241 & 0 & 0.178 \\ 0 & -0.896 & 0.294 & 0.204 \\ 0 & 0 & -0.097 & 0 \\ 0 & 0.055 & 0 & -0.055 \end{bmatrix}$