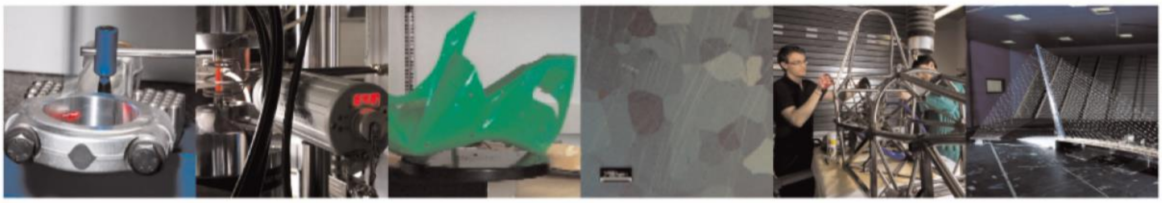

Robust scheduling in a two-machine re-entrant flow shop to minimise the value-at-risk of the makespan: branch-and-bound and heuristic algorithms based on Markovian activity networks and phase-type distributions

Liu, L.; Urgo, M.

This is a post-peer-review, pre-copyedit version of an article published in ANNALS OF OPERATIONS RESEARCH. The final authenticated version is available online at: http://dx.doi.org/10.1007/s10479-023-05647-1

DIPARTIMENTO DI MECCANICA ■ POLITECNICO DI MILANO
via G. La Masa, 1 ■ 20156 Milano ■ EMAIL (PEC): pecmecc@cert.polimi.it
http://www.mecc.polimi.it
Rev. 0

Robust scheduling in a two-machine re-entrant flow shop to minimise the value-at-risk of the makespan: branch-and-bound and heuristic algorithms based on Markovian Activity Networks and phase-type distributions

Lei Liu[1], Marcello Urgo[2]
1, Nottingham University Business School China, University of Nottingham, Ningbo, China
2,Department of Mechanical Engineering, Politecnico di Milano, Milano, Italy

Nov 2023

Abstract

This paper addresses a two-machine re-entrant flow shop scheduling problem with stochastic processing times where each job is expected to require a rework phase, flowing twice within the whole system. Due to the stochastic characteristics of the addressed problem, the proposed approach aims to devise robust schedules, i.e., schedules that are less sensitive to the occurrence of uncertain events, specifically, to the variability of the processing times. Two classes of approaches are proposed: the first is a branch-and-bound algorithm capable of solving the problem optimally, although with limitations regarding the size of the scheduling instances; the second is heuristic algorithms that can be applied to medium/large instances. For both approaches, the goal is to minimise the value-at-risk associated with the makespan, to assist decision-makers in balancing expected performance and mitigating the impact of extreme scenarios. A Markovian Activity Network (MAN) model is exploited to estimate the distribution of the makespan and evaluate its value-at-risk. Phase-type distributions are used to cope with general distributions for the processing times while exploiting a Markovian approach. A set of computational experiments is conducted to demonstrate the effectiveness and performance of the proposed approaches.

Key words: Flow shop; Stochastic scheduling; Re-entrant flow shop; Markovian activity networks; Risk measure

1 Introduction

Flow shop scheduling models are well-known and established approaches for planning a wide range of manufacturing systems where a set of jobs undergo a fixed sequence of operations. Furthermore, to cope with the uncertainty affecting the characteristics of real processes and systems, stochastic scheduling approaches have been proposed supporting the use of random variables to model uncertain factors, e.g., processing times, routings, etc. This class of approaches are especially relevant to match the characteristics of re-manufacturing processes where, due to the unpredictable conditions of used parts to be re-manufactured, operations could require different processing times and/or reworks. Specifically, the motivation for the proposed approach stems from the repairing process of turbines for power generation. In this class of processes, blades are disassembled and re-manufactured by removing the worn parts, adding new material, and restoring the original shape. Within this process, the two most critical operations are the addition of the missing material through an additive manufacturing process and a subsequent grinding process to obtain the final desired shape. Furthermore, due to the variable degree of wear, blades quite always require a further rework by repeating the same sequence of operations, thus competing for the same resources (Liu and Urgo 2022b).

The described process can be modelled as a two-machine re-entrant stochastic flow shop, where jobs are processed in two stations in sequence and, at the end of the process, they enter the system again, repeating the same routing. The deterministic two-machine re-entrant flow shop scheduling problem has received considerable attention (Choi and Kim 2007, 2009; Jeong and Kim 2014), but its stochastic version has been less addressed in the literature. Among the wide range of possible sources of uncertainty, the processing times of manufacturing activities are the most relevant due to their impact on the production schedules. In stochastic scheduling approaches, Random variables and their associated probability distributions are used to model the pertinent sources of uncertainty. Furthermore, most stochastic scheduling approaches aim to optimise a statistic of the objective criteria, e.g., the expected value. Nevertheless, to devise robust schedules, different optimisation criteria should be used to mitigate the impact of extreme events, e.g., indicators able to measure the risk associated with a given schedule.

In this paper, we consider a two-machine re-entrant flow shop scheduling problem with stochastic processing times, and the objective is to minimise the value-at-risk of the makespan. Branch-and-bound and heuristic algorithms are proposed. At the same time, a Markovian Activity Network model is used to estimate the distribution of the objective function, under the hypothesis that processing times follow general phase-type distributions.

The paper is organised as follows: Sect. 2 reviews relevant literature, Sect. 3 describes the addressed scheduling problem and the risk measure used, Sect. 4 presents the proposed branch-and-bound approach. In contrast, heuristic algorithms are presented in Sect. 5, and the results of the experiments are reported in Sect. 6. Finally, Sect. 7 provides the final considerations and conclusions.

2 Literature Review

Re-entrant flow shops, as a special case of flow shops, have been attracting significant attention in the literature (Drobouchevitch and Strusevich 1999), due to their capability of modelling relevant characteristics of real manufacturing problems, as well as for their intrinsic solving difficulty. This class of scheduling problems is further classified according to the characteristic of the routing. The simplest case, i.e., the (1,2,1)-re-entrant flow shop where jobs are processed in the flow shop and then return to the first machine only, is already NP-hard (Emmons and Vairaktarakis 2012). Re-entrant flow shop scheduling models have been proposed for application in the semiconductor industry (Graves et al. 1983) together with a simple and effective scheduling algorithm to minimise the average throughput time. Demirkol and Uzsoy (2000) suggested decomposition methods to minimise the maximum lateness in a re-entrant flow shop with sequence-dependent setup times. Pan and Chen (2003) show that the re-entrant permutation flow shop scheduling problem to minimise the makespan is NP-hard in the strong sense, even for the two-machine case, and propose a mixed integer programming formulation and heuristic algorithms. Choi and Kim (2007, 2009); Jeong and Kim (2014); Choi and Kim (2008) addressed the two-machine re-entrant flow shop scheduling problem to minimise makespan- and tardiness-related objective functions, also extending them to the m-machine version of the problem. Yu and Pinedo (2020) studied two special cases of the ordered re-entrant flow shop, machine-ordered and proportionate flow shops, proposing a dispatching rule to minimise the makespan.

Stochastic scheduling approaches have been proposed to match the characteristics of real scheduling problems, where the occurrence of uncertain events is frequent, capable of modelling the uncertainty through random variables and the associated probability distributions. A review of the works addressing stochastic two-machine flow shop scheduling problems can be found in Gourgand et al. (2000). Within this corpus of works, a special case is represented by modelling processing times with exponential distributions. In such cases, the Talwar rule has been proposed (Talwar 1967) and proved optimal to minimise the expected makespan (Cunningham and Dutta 1973). Within the class of stochastic two-machine flow shop scheduling problems, different objective functions can be pursued: minimisation of the expected maximum completion time, the optimisation of the expectation-variance (De et al. 1992) and minimax regret (Kouvelis et al. 2000). The advantages and limitations linked to the use of these objective functions have been addressed in Tolio and Urgo (2013); Manzini and Urgo (2015); Bertsimas and Sim (2004); Tetenov (2012); Manzini and Urgo (2018) and Urgo (2019). Risk measures derived from applications in the financial area have also been proposed for scheduling problems to pursue robustness. Examples are the value-at-risk and conditional value-at-risk (Filippi et al. 2020; Dixit and Tiwari 2020). Sarin et al. (2014) proposed a scenario-based mixed-integer program formulation to minimise the conditional value-at-risk of the total weighted tardiness for both a single and parallel machine scheduling problem. Tolio et al. (2011); Atakan et al. (2016); Chang et al. (2017); Urgo and Váncza (2019); Kasperski and Zieliński (2019) presented approaches to optimise the value-at-risk of different objective functions within the class of single-machine scheduling problems. Meloni and Pranzo (2020) addressed the minimisation of the conditional value-at-risk of the makespan for a resource-constrained project scheduling problem where the processing time of activities is modelled through an interval in the integer domain.

Nevertheless, in stochastic scheduling problems where the processing times of jobs are modelled through general probability distributions, the main difficulty resides in estimating the distribution of the objective function (Dodin 1985, 1996). Sarin et al. (2010) suggested a method utilising a finite mixture model to estimate any kind of processing time distribution by employing a convex combination of normal distributions, yielding highly favourable outcomes in numerous instances. The mean and variance of the makespan are then computed accordingly through an approximation based on the Clark equation (Clark 1961). However, numerous paths are available in the two-machine re-entrant flow shop scheduling problem, ranging from the first sequenced activity on the first machine to the last activity on the second machine. These paths have at least a few activities in common, leading to correlated completion time distributions. Thus, this approximation may lead to significant errors (Sarin et al. 2010). Furthermore, Markovian Activity Networks (MAN) model has been proposed to support the exact estimation of this distribution considering that the job processing times adhere to an exponential distribution (Kulkarni and Adlakha 1986). To overcome this limitation, extensions have been proposed to cope with generally distributed processing times by approximating them through phase-type distributions (Urgo 2014; Angius et al. 2021). Using this class of models, the distributions of the makespan can be estimated. Based on this, related risk measures can be calculated to support developing a robust schedule. A branch-and-bound approach supported by MAN and phase-type distributions has been proposed for a stochastic two-machine permutation flow shop scheduling problem, without re-entrant flows Liu and Urgo (2023). In contrast, a preliminary version of this approach considering re-entrant flows has been proposed by Liu and Urgo (2022a).

As mentioned, the stochastic version of the re-entrant flow shop scheduling problems has been scarcely addressed in the literature. Dugardin et al. (2010) addressed a stochastic multi-objective re-entrant hybrid flow shop scheduling problem using an approach based on discrete event simulation. Lee et al. (2011) presented a genetic algorithm to solve a stochastic re-entrant flow shop scheduling problem to minimise the weighted tardiness and the makespan.

Various heuristic approaches have been proposed to tackle both the deterministic and stochastic versions of the scheduling problems (Juan et al. 2023). The Iterated Local Search (ILS) framework (Lourenço et al. 2019) has successfully solved the deterministic flow shop scheduling problem. Specifically, the NEH constructive heuristic (Nawaz et al. 1983), along with the iterated greedy(IG) algorithm (Ruiz and Stützle 2007), is regarded as the most effective method (Benavides and Vera 2022). Baker and Altheimer (2012) introduced three heuristics, namely CDS/Johnson, CDS/Talwar and NEH, to address the stochastic m-machine flow shop scheduling problem. These heuristics were compared in terms of efficiency, revealing no clear dominance among them. Additionally, Wang et al. (2005a,b) proposed genetic algorithms to minimise the expected makespan when processing times follow a uniform distribution. By adapting the iterated greedy and NEH algorithms to suit the characteristics of the two-machine re-entrant flow shop scheduling problem with stochastic processing times, they can be coupled with the estimation of the distribution of the objective function using Markovian Activity Networks (MAN).

3 Problem formulation

In a re-entrant two-machine permutation flow shop, jobs are processed on the two machines (M_a, M_b) in series, and after their processing, a rework is required on both M_a and M_b. Following the formalisation introduced in Choi and Kim (2007), jobs are classified into two sets N and N'. Jobs in N visit the machines for the first time and are called *first-pass jobs*. In contrast, the jobs in N' are the ones visiting the machines for the second time and are called *second-pass jobs*. Therefore, given n jobs to be processed in the flow shop, $2n$ jobs must be considered, i.e., n first-pass jobs and n second-pass jobs. A precedence relation has to be defined between a first-pass job and its corresponding second-pass job, i.e., a second-pass job can only be processed on M_a after the corresponding first-pass job has been completed on M_b. A job is considered completed when its second-pass job is completed on M_b. A permutation flow shop problem is considered; thus, the sequence of jobs on M_a and M_b is the same. The described scheduling problem matches a subset of real re-manufacturing processes, often entailing additional steps before (e.g., cleaning, inspection, etc.) and after (inspection, assembling, etc.) the considered two. Additional precedence constraints are imposed to match the constraints derived by process steps not included in the current formalisation. A rework is usually decided after an inspection, triggering the definition of process parameters for the rework itself. As inspecting the part reasonably requires time, the following assumption is defined:

- a second-pass job can be processed at least two jobs after the corresponding first-pass one unless no other jobs to be processed are available.

The processing time of a job $j \in N \cup N'$ on machine $M_i, i = a, b$, denoted as p_{ij}, is modelled as an independent random variable following a general phase-type distribution. No limitations or constraints are imposed on the number of phases and structure, which allows for its use in approximating any positively valued distribution (Bladt and Yslas 2022). Due to the uncertainty affecting processing times, the makespan is also a random variable depending on p_{ij}, as well as on scheduling decisions. The proposed scheduling approach aims to mitigate the impact of longer processing times on the makespan. Thus, the minimisation of the value-at-risk (VaR) of the makespan is used as the objective function.

Definition 3.1. The value-at-risk α (VaR_α) of a performance indicator z associated with decisions $\boldsymbol{x}$ can be defined as:

$$\zeta_\alpha(\boldsymbol{x}) = min\{\zeta | F_z(\boldsymbol{x}, \zeta) \geq 1 - \alpha\} \tag{1}$$

where F is the cumulative distribution function of z and α the risk level.

In the two-machine re-entrant flow shop scheduling problem under investigation, the decision vector $\boldsymbol{x}$ defines the sequencing of the jobs, while a vector of random variables $\boldsymbol{p} = \{p_{a,1}, ..., p_{b,2n}\}$ models the processing times associated to the jobs. These random variables are governed by a probability measure $\mathbb{P}$ and are independent of sequencing decisions in $\boldsymbol{x}$. If the considered performance indicator is the makespan, $z = C_{max}$, for a given schedule $\boldsymbol{x}$, the cumulative density function (cdf) for the makespan is defined as:

$$F_{C_{max}}(\boldsymbol{x}, \zeta) = P(C_{max}(\boldsymbol{x}) \leq \zeta | \boldsymbol{x}) \tag{2}$$

Then, the VaR_α of C_{max}, associated with a schedule defined by $\boldsymbol{x}$, is defined according to the following equation:

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

The described problem can be defined as $F2/re-entrant/\boldsymbol{p}_{ij}/VaR_{C_{max}}$ (Emmons and Vairaktarakis 2012). Furthermore, since the sequencing decision vector $\boldsymbol{x}$ is independent of the values of stochastic variables in $\boldsymbol{p}$, and the makespan C_{max} is a regular scheduling objective function (Pinedo 2016), the value of $C_{max}(\boldsymbol{x})$ is non-decreasing with respect to the scheduling of a new job or the introduction of additional precedence constraints. As a result, the objective function value (VaR) for a partial schedule serves as a lower bound for the objective function value of schedules that include extra jobs (Ma and Wong 2010). Table 1 summarises the decision variables and parameters modelled to address the considered scheduling problem.

Table 1 Parameters and decision variables.

Notations	
$\boldsymbol{x}$	decision vector
α	risk level
$\zeta_\alpha(\boldsymbol{x})$	VaR_α value associated with decision $\boldsymbol{x}$
p_{ij}	processing time of job j on machine i, $j = 1, ..., 2n$, $i = a, b$
$\boldsymbol{p}$	vector of processing time variables, $\boldsymbol{p} = \{p_{a,1}, ..., p_{b,2n}\}$
$C_{max}(\boldsymbol{x})$	makespan associated to sequence decisions $\boldsymbol{x}$
$F_{C_{max}}(\boldsymbol{x}, \zeta)$	cumulative density function (cdf) of the makespan

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

4 Branch-and-bound algorithm

The optimisation of the value-at-risk associated with the makespan is accomplished by employing a branch-and-bound algorithm that relies on the following fundamental components:

1. *Initial bound*: Leveraging available heuristic approaches, an initial upper bound is determined.
2. *Branching scheme and search strategy*: A branching scheme is established to generate the nodes in the branching tree, while the depth-first strategy is employed to facilitate the search for the optimal solution.
3. *Evaluation of the nodes*: A Markovian Activity Network is constructed for each node in the tree to estimate the distribution of the makespan. A lower bound is derived for nodes representing partial solutions (schedules).

The proposed branch-and-bound approach is derived from the one presented in Liu and Urgo (2023), designed for a similar scheduling problem considering single processing of the jobs in the flow shop. Moving to a re-entrant stochastic flow shop problem entails adapting the branch-and-bound approach to the specific characteristics of this problem.

The approach proposed in Liu and Urgo (2023) takes advantage of a dominance rule among schedules (either partial or complete) to speed up the pruning of branches. When considering a re-entrant flow shop scheduling problem, this dominance rule cannot be used anymore. It relies on the hypothesis that the first machine in the shop is never idle. This cannot be guaranteed with re-entrant jobs. In fact, as a second-pass job is scheduled on the first machine, its processing can start only if it has completed the first pass in the shop (i.e. if the corresponding first-pass job has been completed), thus, forcing the first machine to wait and remain idle.

Secondly, the generation scheme of the Markovian Activity Network has to match the characteristics of the re-entrant flow shop problem considering additional precedence relations between first- and second-pass jobs.

4.1 Initial upper bound

Following Sect. 2, the rule introduced in Talwar (1967) offers an optimal schedule for a stochastic two-machine flow shop scheduling problem which involves exponentially distributed processing times and aims to minimise the expected makespan. The rule mentioned above serves as a heuristic approach, applicable even when processing times are generally distributed (Baker and Trietsch 2010). Consequently, this rule is utilised to establish an initial value for the objective function in the two-machine re-entrant flow shop scheduling problem, and it serves as the initial incumbent solution for the branch-and-bound algorithm, thus providing a foundation for subsequent optimisation (Emmons and Vairaktarakis 2012).

According to Talwar (1967), for each job $j \in N \cup N'$, The respective expected values of the processing times on the two machines (M_a and M_b) are denoted as $E(j_a)$ and $E(j_b)$. The initial solution can be defined by arranging the jobs in decreasing order of the difference between the multiplicative inverses of $E(j_a)$ and $E(j_b)$:

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

Note that if the resulting schedule conflicts with the constraints for sequencing second-pass jobs, they are shifted towards the right until all conflicts are solved.

4.2 Branching scheme and search strategy

As described in Sect. 3, a solution for the scheduling problem is defined by $\boldsymbol{x}$. Specifically, $\boldsymbol{x}_k$ represents the index assigned to the job located in the k-th position of the sequence. A branching scheme is defined to support the proposed branch-and-bound algorithm, considering the sequencing of both first-pass and second-pass jobs. A forward branching scheme is exploited in this study, sequencing the jobs starting from the first job in the schedule, starting from the root node (level 0). From this node, $2n$ branches depart, one for each job in the list that can be the next in the sequence. Considering a node at the $k-1$ level of the branching tree, the partial schedule provides the sequence of the first $k-1$ jobs while $2n-k+1$ branches are connected to nodes at level k. Due to the need to respect the constraints for sequencing second-pass jobs, nodes in the branching tree representing partial schedules violating these constraints are pruned.

Concerning the search strategy for the branching tree, it must be noticed that the proposed lower bound might not be so tight for incomplete schedules with a small number of scheduled jobs. Furthermore, since no upper bound has been defined, apart from the initial upper bound described in Sect. 4.1, it is crucial to drive the branch-and-bound towards a new solution to the problem as soon as possible. To this aim, the depth-first strategy is exploited, which aims to swiftly reach a leaf node while efficiently pruning non-leaf nodes to enhance the exploration of the branch tree.

4.3 Evaluation of leaf nodes

Within the branching tree, the leaf nodes correspond to complete schedules encompassing all $2n$ jobs. The duration of the makespan is contingent upon the critical path's length within the activity network. However, when stochastic processing times are taken into account, there exist multiple potential paths that could be critical (Dodin 1985), thereby making the calculation of the makespan's distribution estimation inherently challenging (Dodin 1996). A Markovian Activity Network model is employed to tackle this issue, wherein the processing times are represented by phase-type distributions, facilitating the estimation process.

The two-machine re-entrant flow shop scheduling problem is abstractly depicted as an acyclic-directed graph denoted as $G = (V, A)$. The set of arcs A symbolise activities, while the set of nodes V represent states that illustrate the progression in activity execution. Within this framework, as outlined by the model proposed by Kulkarni and Adlakha (1986), activities at a given time t can assume one of the following states:

- Active: The activity is presently being executed and can be represented as (j);
- Dormant: The activity has been completed, yet an incomplete activity is connected to the same destination node. This condition can be expressed as (j^*);
- Idle: The activity is neither active nor dormant.

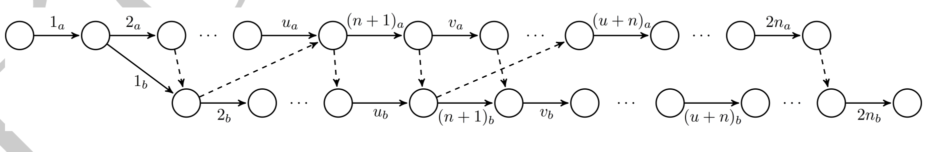

Figure 1 AoA activity network for a complete schedule.

The AoA network modelling a two-machine re-entrant flow shop scheduling problem is shown in Fig. 1, with u and v being arbitrary first-pass jobs. Drawing upon this network, the set of states that model the execution of activities can be derived. Commencing from the state denoting the processing of the first job on machine a as (1_a), a transition occurs once activity 1_a concludes. This transition leads to a state where two activities are simultaneously underway: the first job on machine b and the second job on machine a, indicated as $(1_b, 2_a)$. As the execution of these activities progresses, a subsequent transition becomes possible, leading to one of two independent states. The first state, $(1_b^*, 2_a)$, signifies the completion of the first job on machine b while the second job on machine a is still being processed. The second state, $(1_b, 3_a)$, denotes the completion of the second job and the initiation of processing the third job in the sequence on machine a, while the first job is still being processed on machine b. This pattern persists until the system reaches an absorbing state, representing the comprehensive processing of all jobs on the two machines.

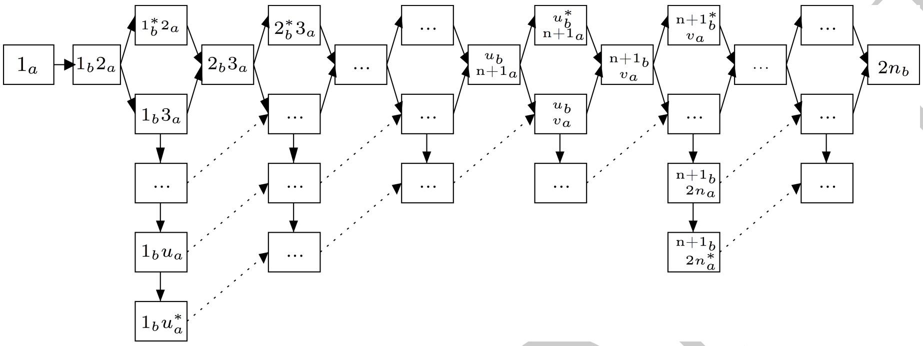

Figure 2 States generation scheme.

Based on the assumption of exponentially distributed processing times, the aforementioned approach gives rise to a continuous-time Markov chain (CTMC) which can be characterised by an initial probability vector and an infinitesimal generator matrix (Kulkarni and Adlakha 1986). As exemplified in Fig. 2, the CTMC defines a general structure of the state-space associated with the considered scheduling problem, which can be subsequently utilised in all the leaf nodes to consider the specific schedule to be evaluated. The makespan of this schedule, i.e., the time spanning from when the first job begins being processed on machine a to the completion of the last job on machine b, is the time to absorption of the CTMC. Furthermore, to expand upon the approach introduced in Kulkarni and Adlakha (1986), which solely applies to exponentially distributed processing times, the infinitesimal generator of the CTMC incorporating phase-type distributions for the processing times of the jobs, can be derived through a Kronecker algebra technique (Angius et al. 2021). As a result, the distribution of the associated makespan can be computed as follows:

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

Here, the symbol β represents the initial probability vector, T refers to the infinitesimal generator matrix excluding the absorbing state, and $\mathbf{1}$ represents a vector consisting of all ones(Ross et al. 1996; Urgo 2014). The exponential of the matrix operator can be calculated using the Krylov subspace method (Sidje and Stewart 1999).

The quantile of this distribution corresponding to the VaR_α is obtained through a root finding method, specifically the bracket and solve method (Boost 2020), 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.

Under the hypothesis that the processing times of the jobs is phase-type distributed, the described approach provides an exact calculation of the value-at-risk of the makespan. The only possible source of approximation arises from using phase-type distributions to approximate general distributions. However, it is worth noting that the set of phase-type distributions is dense in the field of all positive-valued distributions (Bladt and Yslas 2022), and by increasing the number of phases, the accuracy of the fitting can be enhanced, providing the possibility to improve the approximation as needed.

4.4 Evaluation of nodes representing partial schedules

To obtain bounds for the objective function under consideration, it is necessary to evaluate nodes that represent partial schedules, i.e., schedules with only a subset of the jobs sequenced.

According to the branching scheme described in Sect. 4.2, in a non-leaf node of the tree, k jobs have already been sequenced, while the sequencing of the remaining $2n - k$ jobs has not been decided yet. For the k assigned jobs, a similar approach as described in Sect. 4.3 is employed to determine the initial and final segments of the corresponding activity network. On the contrary, for the jobs yet to be sequenced, their processing times on the two machines are represented by two dummy activities (r_a and r_b). These dummy activities have processing times equivalent to the sum of processing times of unscheduled jobs' activities on the respective machines. Thus, possible precedence relations between these activities are omitted. Fig. 3 illustrates the resulting AoA activity network. To establish the initial state space, analyse the CTMC incorporating phase-type distributed processing times, and estimate the VaR, the identical approach as described in Sect. 4.3 is applied.

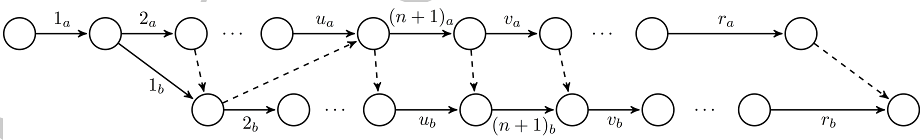

Figure 3 AoA activity network for a partial schedule.

As addressed in Sect. 3, the VaR of the makespan is a regular objective function. Thus, when a new job is scheduled, its VaR value cannot decrease. Consequently, the calculated VaR serves as a lower bound of the VaRs of all the nodes in the branches originating from the examined node (Ma and Wong 2010). If the lower bound associated with a partial schedule exceeds or equals the incumbent solution, the corresponding node in the tree is pruned.

5 Heuristic algorithm

The branch-and-bound algorithm described in Sect. 4 is designed to solve the addressed problem optimally. Still, it is expected to be inefficient in solving medium-/large-scale instances. Thus, heuristic algorithms, i.e., iterated greedy heuristic (IG) (Ruiz and Stützle 2007) and NEH heuristic (Nawaz et al. 1983), are proposed to solve larger instances.

5.1 Iterated greedy heuristic algorithm

Starting from the same initial solution used for the branch-and-bound approach (see Sect. 4.1), the IG algorithm is used to generate a sequence of solutions through the iteration of a greedy insertion heuristics with two main phases: removal and insertion. The removal procedure is applied to a permutation $\boldsymbol{x}$ of $2n$ jobs, and it randomly chooses a first-pass job, and its corresponding second-pass job, these two jobs are then removed from $\boldsymbol{x}$, obtaining a sub-sequence $\boldsymbol{x}_s$. The insertion phase starts from $\boldsymbol{x}_s$ and inserts the two removed jobs into all the possible positions. The best sequence $\boldsymbol{x'}$ is the one that yields the smallest value of the $VaR_{C_{max}}$. The evaluation of the feasibility of the sequence is operated in a way similar to the evaluation of the leaf nodes described in Sect. 4.3 by exploiting the Markovian Activity Network (MAN) approach. This process is iterated until the termination criterion, i.e., the number of iterations or improvement of the objective function, is reached. The outline of the proposed insertion algorithm is provided in Algorithm 1.

Algorithm 1: Iterated greedy (IG) heuristic algorithm

```
input: iter := 0, initial schedule x, improve := true, number of iterations
       T
while iter ≤ T OR improve=true do
    improve:=false
    Randomly remove a first-pass job k and the corresponding
    second-pass job (k + n) from x (no repetition)
    x' :=best permutation obtained by inserting k and (k + n) in any
    possible positions in x
    if VaR_{C_max(x')} < VaR_{C_max(x)} then
        x := x'
        improve :=true
    end
    iter := iter+1
end
```

The IG heuristic is closely related to the Iterated Local Search framework (Lourenço et al. 2019) but, rather than iterating over a local search operation, e.g., shifting and swapping, it iterates over removal and insertion operations only (Ruiz and Stützle 2007). Due to the complexity of estimating the value-at-risk associated with a solution, the IG heuristic algorithm is preferred to an Iterated Local Search approach since it is expected to perform better.

5.2 NEH heuristic algorithm

Together with the iterated greedy (IG) algorithm, the NEH constructive heuristic is considered among the best heuristic methods proposed for the permutation flow shop scheduling problem to minimise the makespan (Benavides and Vera 2022). The NEH constructive heuristic was proposed by Nawaz et al. (1983)

and consists of two main steps. The first one takes an initial schedule of jobs as input and takes the first two sequenced jobs. The best way of sequencing these two jobs is selected, grounding on the considered objective function, and constitutes the initial partial solution. The second step takes the first job in the initial schedule that has not been included in the partial solution. Then it evaluates the impact of inserting it in all the possible positions in the partial schedule, selecting the one that leads to the best partial solution. This step is iterated until a complete schedule is obtained. The application of the NEH constructive heuristic to the considered stochastic two-machine re-entrant flow shop scheduling problem is described in Algorithm 2.

Algorithm 2: NEH constructive heuristic algorithm

Input: initial full schedule $\boldsymbol{x}_{ini}$, initial global VaR value $VaR_g = +\infty$, initial updated schedule $\hat{\boldsymbol{x}}$=null

Step 1: The first two jobs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are taken, and the two possible partial schedules starting from them, $[\boldsymbol{x}_1, \boldsymbol{x}_2,...]$ and $[\boldsymbol{x}_2, \boldsymbol{x}_1,...]$, are evaluated. If the VaR value of the partial schedule is smaller than the global VaR value VaR_g, VaR_g is updated and $\hat{\boldsymbol{x}}$ is updated to the associated partial schedule. The first two jobs from $\boldsymbol{x}_{ini}$ are removed to update the schedule of the unassigned jobs $\overline{\boldsymbol{x}}$.

while *size of* $\hat{\boldsymbol{x}} <$ *size of* $\boldsymbol{x}_{ini}$ **do**

Choose the first job from $\overline{\boldsymbol{x}}$ and insert it into each of the possible positions in $\hat{\boldsymbol{x}}$ to get new partial schedule solutions. Check that these solutions do not conflict with the relations between first-pass and second-pass jobs, then select the best one in terms of their VaR. Update VaR_g, $\hat{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$.

end

Output: Final full schedule $\hat{\boldsymbol{x}}$ and the associated VaR value VaR_g.

6 Computational results

The branch-and-bound algorithm and heuristic algorithms were implemented using the C++ programming language, effectively utilising the Eigen library (Guennebaud et al. 2010) for evaluating the absorption time of the Markov activity network. Additionally, the branch-and-bound algorithm made use of the BoB++ library (Djerrah et al. 2006). A comprehensive set of experiments and comparisons was designed and conducted to assess the performance and effectiveness of these algorithms. All experiments were executed on a Windows 7 workstation equipped with a 2.6 GHz Intel Xeon processor and 64 GB of RAM. A CPU time limit of 7200 seconds was set for the experiments.

6.1 Generation of the test instances

Small- and medium-sized test instances have been considered to support the assessment of the performance of the branch-and-bound and heuristic approaches. Small-sized instances contain 5, 6 or 8 jobs, corresponding to 10, 12 and 16 jobs, respectively, when including first-pass and second-pass jobs. Medium-sized instances comprise 10, 15, 20 or 25 jobs, resulting in a total number of 20, 30, 40 or 50 jobs.

The processing times of the jobs are modelled with phase-type distributions, randomly generated using the BuTools library (Horvath and Telek 2017)starting from the desired mean value and number of phases (Butools 2018). The mean value is randomly sampled from three uniform distributions: [0, 20], [30, 50], and [60, 80]. This applies for all but 25(50)-job instances which are derived from the Taillard dataset (Taillard 1993), specifically from the 50-job and 5-machine instances, by only considering two machines only. Furthermore, the deterministic processing times in the Taillard dataset are used as the mean value of the random processing times.

The number of phases is randomly chosen between 1 and 4 for generating the distributions for all the instances. It is important to note that the generation approach provided by the BuTools library does not allow explicit control of higher-order moments such as the variance and the skewness.

Different risk levels (α) were exploited to conduct the experiments, specifically 5%, 10%, and 20%. A total of 420 instances were generated by creating 20 test instances for each combination of the number of jobs n and risk level α.

6.2 Analysis of the branch-and-bound approach

The first objective of the experiments is to assess the performance of the branch-and-bound approach. This has been carried out on small-sized instances, with 5, 6, and 8 first-pass jobs, for a total of 180 test instances. The branch-and-bound algorithm was able to find the optimal solution for 95.9% of the instances within a time limit of two hours. Specifically, it was able to solve all 5- and 6-job instances and 87.7% of the 8-job ones.

The results of these experiments are reported in Table 2. The table reports the statistics for the solution time and number of evaluated nodes. The first three columns also provide an indication of the number of first-pass jobs, the total number of jobs (in parentheses), the fraction of the total number of instances that the algorithm was able to solve and the specific risk level α.

On average, the proposed branch-and-bound approach was able to solve the problem instances in about 1500 seconds (i.e., 25 minutes), with the actual time ranging from a minimum of less than one second, up to two hours for those instances where the algorithm was not able to find the optimal solution. With respect to the number of evaluated nodes, on average, about 60 thousand of them were analysed, which corresponds to (1.78%) of the total number of nodes in the branching tree.

The results for the solution time clearly show, as expected) a dependence on the number of jobs. Nevertheless, the analysis of the impact of the risk level is less explicit. In fact, for a given dimension of the scheduling problem, the value of α seems to impact the solution time. Furthermore, the analysis of the number of evaluated nodes also points to the possibility that this dependence could be due to the need to explore a larger number of nodes to reach the optimal solution.

A two-factor ANOVA analysis has been carried out to investigate the possible effect of the factors highlighted above (number of jobs and risk level) on the performance of the branch-and-bound algorithm. The results of this analysis are reported in Table 3, showing that the solution time significantly depends on

Table 2 Branch-and-bound approach results.

# jobs*	# optimal (%)	risk level	solution time (s)			evaluated nodes			
			mean	min	max	mean	min	max	(%)
5 (10)	100%	5	4.4	0.7	10.8	576	26	1593	1.49
		10	6.0	1.2	14.6	893	381	2111	2.32
		20	6.6	1.8	15.9	605	259	1723	1.57
		ALL	5.7	0.7	15.9	691	26	2111	1.79
6 (12)	100%	5	49.4	2.6	153.5	4412	37	13189	0.88
		10	97.6	8.8	240.3	9908	706	25702	1.98
		20	81.9	13.4	241.4	8362	930	28933	1.67
		ALL	76.4	2.6	241.4	7561	37	28933	1.51
8 (16)	87.7%	5	3611.1	582.8	6636	158307	25809	474834	0.13
		10	4805.2	824.8	7200	184013	20074	544436	0.15
		20	4902.3	693.7	7200	176301	20226	474290	0.14
		ALL	4439.5	582.8	7200	172823	20074	544436	0.14
ALL	95.9%		1507.2	0.7	7200	60358	26	544436	1.14

* the number in () denotes the total number of jobs, including re-entrant ones.

the number of jobs. On the contrary, the risk level and the interaction between these two factors are not significant. It must be noticed that the experiments have been completely randomised, using different randomly generated instances for each experiment within the same combination of levels for the factors. Thus blocking effects linked to the instances are not considered. On the contrary, some of the characteristics of the instances can impact the performance.

Table 3 ANOVA table for the solution time of the approach.

	Df	Sum Sq	F-value	P-value
number of jobs	2	$2.70 * 10^8$	321.77	$1.09 * 10^{-58}$
risk level	2	$3.82 * 10^5$	0.45	0.63
number of jobs:risk level	4	$6.33 * 10^5$	0.37	0.82
residuals	171	$7.17 * 10^7$	-	-

$R^2 = 0.79$, $R^2_{adj} = 0.78$

Further analyses have been operated considering the total number of nodes evaluated during the search, and the average time needed to evaluate a single node. The number of evaluated nodes is expected to identify situations where multiple schedules have very similar objective function values. This reduces the capability of the lower bound to prune branches and forces the algorithm to evaluate a larger number of nodes and solutions. In contrast, the average time to evaluate a node is expected to identify the impact of the computational effort to assess the distribution of the objective function and the associated VaR. The total number of jobs surely impacts this, although other characteristics of the instances could play a role.

A comparison was carried out to check for the possible influence of the number of jobs on the average number of nodes to be evaluated in the optimisation and the average evaluation time per node. The results are reported in Fig. 4, showing that instances with a larger number of jobs will take longer to be solved due to a larger number of nodes to be evaluated in the search, as well as a higher time needed to evaluate a single node, reasonably due to the requirement to handle larger infinitesimal generator matrices for the evaluation of both partial and complete schedules. With respect to both the indicators, the graphs in Fig. 4 also shows a very wide range of values and the presence of outliers. Thus, also with respect to them, the resulting characteristics of the instances have a significant impact and can affect the performance unpredictably.

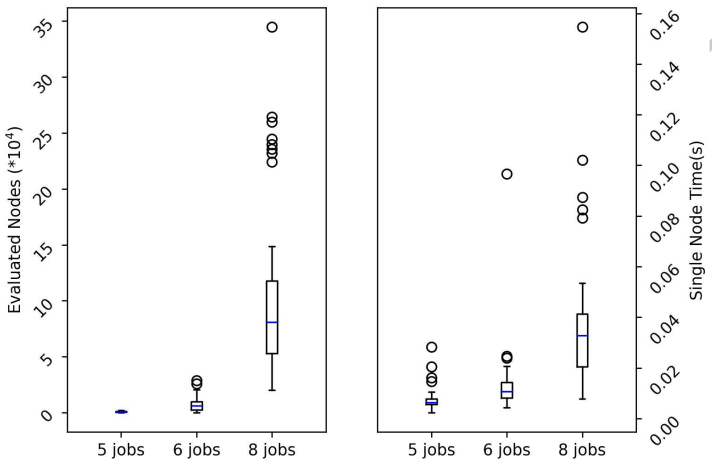

Figure 4 Number of evaluated nodes and time to solve a single node for the different dimensions of the instances.

Returning to the results of the ANOVA in Table 3, there is no statistical evidence to state that the solution time is affected by the different risk levels. With respect to this, Fig. 5 provides a box plot of the solution time with respect to the number of evaluated nodes, considering the different risk levels and confirming, also visually, that no clear difference emerges.

An additional investigation has been executed to assess the effectiveness of the initial solution obtained through the Talwar rule (Sect. 4.1). Firstly, the number of times the incumbent solution is updated is reported in Fig. 6(a), which is consistently below 10. Thus the initial solution accelerates the branch-and-bound algorithm when seeking the optimal solution. Nevertheless, the contribution of the branch-and-bound algorithm remains significant. Fig. 6(b) reports the percentual improvement from the initial to the optimal solution, showing that the branch-and-bound algorithm can improve the initial solution of 6.6% on average, with a minimum and maximum improvement of 0.0% and 22.6%, respectively.

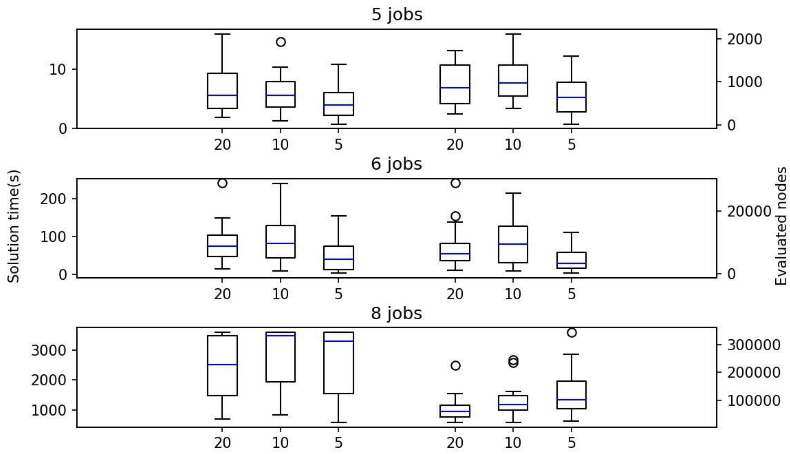

Figure 5 Box plot of the solution time and the number of evaluated nodes in relation to the considered risk levels.

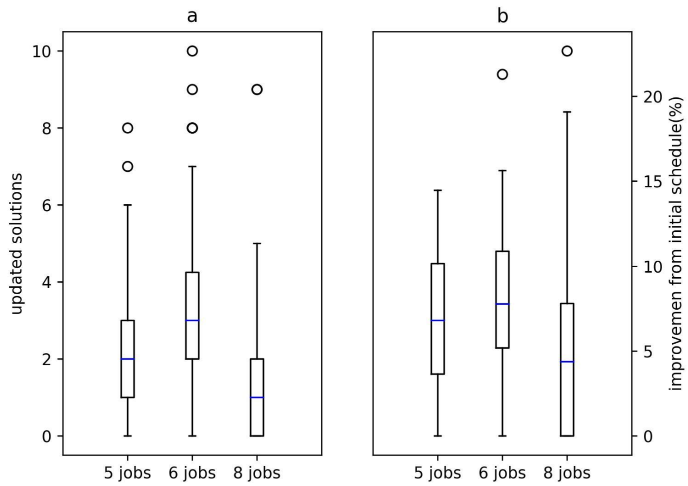

Figure 6 Updated solutions(a) and improvement(b) from the proposed initial solution.

Table 4 Performance of the algorithms with respect to the initial solution.

# jobs*	Δ%											
	B&B				IG				NEH			
	mean	min	max	StDev	mean	min	max	StDev	mean	min	max	StDev
5 (10)	7.2	0.0	14.5	3.8	6.7	0.0	14.4	3.8	3.5	0.0	7.2	2.1
6 (12)	8.2	0.0	21.3	4.3	7.4	0.0	15.6	4.0	3.6	0.0	7.9	3.6
8 (16)	7.5	0.7	22.7	5.0	7.2	0.0	25.6	4.7	3.9	2.9	6.1	1.3
10 (20)	9.1	4.5	18.3	5.3	7.1	1.1	15.5	3.6	2.9	0.3	6.1	2.5
15 (30)	6.1	4.9	8.3	1.6	6.6	0.0	15.5	3.3	3.5	0.0	14.8	6.4
20 (40)	3.7	1.6	5.2	1.4	7.6	0.0	15.0	3.6	4.8	1.7	7.5	1.8
25 (50)	2.6	1.1	6.1	2.1	8.1	1.7	19.6	7.0	7.6	3.2	13.9	3.2

As indicated during the initial analysis of the outcomes, while solving instances with $n = 8$ and $2n = 16$, the branch-and-bound algorithm failed to find the optimal solution within the specified time limit in approximately 12.3% of the instances. Furthermore, an additional examination was conducted to evaluate the quality of the obtained incumbent solution obtained from the branch-and-bound algorithm, together with the investigation and comparison of alternative heuristic approaches, i.e., IG and NEH. Besides considering small-sized instances not solved optimally, this analysis also considers larger instances where the branch-and-bound approach can never reach the optimum within a given time limit of two hours.

To support this analysis, the performance indicator Δ% is used (Eq. 7) to measure the percentual improvement obtained by the algorithm with respect to the initial solution value S_0.

$$\Delta\% = \frac{S_0 - Output_{algo}}{Output_{algo}} \tag{7}$$

The results in Table 4 show that the proposed branch-and-bound algorithm can improve the initial solution by 6.4%, demonstrating its effectiveness even on larger instances, where the capability to achieve the optimality cannot be guaranteed.

With respect to heuristic approaches, the IG heuristic was able to improve the VaR of the initial solution of 7.2% on average, ranging from a minimum improvement of 0.0% to a maximum of 25.6%. The NEH heuristic improved the VaR of the initial solution of 4.2% on average, ranging from a minimum improvement of 0.0% to a maximum of 14.8%.

The improvement based on the initial solutions demonstrates that the branch-and-bound approach dominates heuristic ones for smaller instances (up to 20 jobs in total). In contrast, as the dimension of the instances increases, heuristic approaches can better improve the solution. Furthermore, the IG heuristic clearly dominates the NEH one.

With respect to the computation time, the complexity of the IG heuristic is

$O(T*[2n]^2)$, where T represents the number of iterations (Algorithm 1). Setting the number of iterations (T) to 10, the IG approach was able to converge in less than 1 second for both 5- and 6-job instances and within 20 seconds for 8-job instances. For larger instances of up to 25 jobs, convergence could be achieved within 2 hours. Regarding the NEH heuristic, its complexity is $O([2n]^2)$, where $2n$ is the total number of jobs (including re-entrant jobs). However, due to the partial schedule evaluation in the MAN, specifically the Kronecker product operation, a larger infinitesimal generator matrix is derived, which requires higher calculation times. Additionally, as a constructive heuristic, the NEH algorithm cannot be terminated within a specific time limit.

6.3 Comparison with alternative robust scheduling approaches

To evaluate the benefits of scheduling to minimise the value-at-risk of the makespan, alternative robust scheduling approaches have been implemented and compared, i.e., approaches minimising the maximum (Levorato et al. 2022; Juvin et al. 2023) and expected processing times (Levorato et al. 2022).

Two representative instances with 5(10) and 8(16) jobs have been randomly selected among the ones generated according to Sect. 6.1. The details for these instances are reported in Appendix A. Since the support of phase-type distributions is not bounded, minimum and maximum values are not available, entailing difficulties in using approaches considering maximum processing times. To this aim, the 0.01% and 99.99% quantiles are considered as the minimum and maximum values, respectively. To minimise the maximum makespan, a global budget Γ, which denotes the maximum number of operations whose uncertain processing times can reach their worst-case values, is randomly chosen from $[80\%, 100\%]$ as the total number of activities (Juvin et al. 2023).

For each representative instance, three optimal schedules are derived using the three different objective functions, i.e., minimising the VaR, the minimax, and the expected value of the makespan. The latter is operated by using a deterministic problem where the processing times are equal to their expected values. The three approaches led to different schedules for each instance h, i.e., $\boldsymbol{h}_{\text{i,VaR}}$, $\boldsymbol{h}_{\text{i,minimax}}$, and $\boldsymbol{h}_{\text{i,expval}}$. Specifically, for each of the two instances considered, the three optimal schedules are:

$$\boldsymbol{x}_{1,\text{VaR}} = \{1 \rightarrow 3 \rightarrow 5 \rightarrow 1' \rightarrow 2 \rightarrow 3' \rightarrow 4 \rightarrow 5' \rightarrow 2' \rightarrow 4'\}$$
$$\boldsymbol{x}_{1,\text{minimax}} = \{5 \rightarrow 1 \rightarrow 3 \rightarrow 5' \rightarrow 2 \rightarrow 1' \rightarrow 4 \rightarrow 3' \rightarrow 2' \rightarrow 4'\}$$
$$\boldsymbol{x}_{1,\text{expval}} = \{5 \rightarrow 1 \rightarrow 3 \rightarrow 5' \rightarrow 2 \rightarrow 4 \rightarrow 3' \rightarrow 1' \rightarrow 2' \rightarrow 4'\}$$

$$\boldsymbol{x}_{2,\text{VaR}} = \{4 \rightarrow 8 \rightarrow 7 \rightarrow 5 \rightarrow 4' \rightarrow 2 \rightarrow 8' \rightarrow 7' \rightarrow 6 \rightarrow 1 \rightarrow 3 \rightarrow 5' \rightarrow 2' \rightarrow 6' \rightarrow 1' \rightarrow 3'\}$$
$$\boldsymbol{x}_{2,\text{minimax}} = \{8 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 6 \rightarrow 5' \rightarrow 7' \rightarrow 4' \rightarrow 1 \rightarrow 2 \rightarrow 8' \rightarrow 3 \rightarrow 6' \rightarrow 2' \rightarrow 3' \rightarrow 1'\}$$
$$\boldsymbol{x}_{2,\text{expval}} = \{4 \rightarrow 5 \rightarrow 8 \rightarrow 6 \rightarrow 4' \rightarrow 5' \rightarrow 8' \rightarrow 2 \rightarrow 3 \rightarrow 6' \rightarrow 7 \rightarrow 1 \rightarrow 2' \rightarrow 3' \rightarrow 7' \rightarrow 1'\}$$

For each instance, 1000 scenarios were randomly generated and the different schedules were tested to evaluate the associated makespan. Table 5 displays the number of times each schedule achieved the minimum makespan over the considered scenarios, demonstrating a clear dominance of the minimisation of the VaR. Moreover, to quantitatively assess the difference between the minimisation of the VaR and the alternative approaches, a performance indicator 8 is defined with $\boldsymbol{x}_{\text{alt}}$ being the optimal schedule obtained with the minimax or deterministic expected value approaches. The results of this analysis are presented in Table 6.

$$\Delta_{alt}\% = \frac{C_{max}(\boldsymbol{x}_{\text{alt}}) - C_{max}(\boldsymbol{x}_{\text{VaR}})}{C_{max}(\boldsymbol{x}_{\text{VaR}})} \tag{8}$$

Table 5 Number of times the different schedules obtained the smallest makespan over the considered 1000 scenarios.

	VaR approach	Expected value approach	Minimax approach
Instance 1	633/1000	213/1000	154/1000
Insstance 2	511/1000	172/1000	317/1000

Table 6 Improvement of VaR approach with respect to alternative robust scheduling approaches

	$\Delta_{minimax}\%$				$\Delta_{expval}\%$			
	mean	min	max	median	mean	min	max	median
Instance 1	4.1	-19.5	33.2	7.5	3.9	-17.1	36.8	6.1
Instance 2	1.5	-15.2	37.1	6.7	6.2	-20.9	31.5	4.9

Across all the 1000 scenarios considered, the proposed VaR approach shows an average improvement ranging from 1.5% to 6.2% compared to the alternative approaches. It is worth noting that in certain extreme scenarios, the VaR approach can provide protection by improving the performance exceeding 30%. This demonstrates the efficiency of the VaR approach in assisting decision-makers in balancing expected performance and mitigating the impact of extreme scenarios.

Finally, Fig. 7 reports the graphs of the cumulative distribution functions (CDF) of the makespan for the schedules obtained with different alternative approaches, over the 1000 sampled scenarios. Specifically, the cdf of the makespan for the schedule minimising the VaR is stochastically smaller (first-order stochastic dominance) than the cdfs of the schedules minimising the other two alternative objective functions (Pinedo 2016). Thus, based on these analyses, the proposed minimisation of the VaR provides better results compared to the two alternative objective functions.

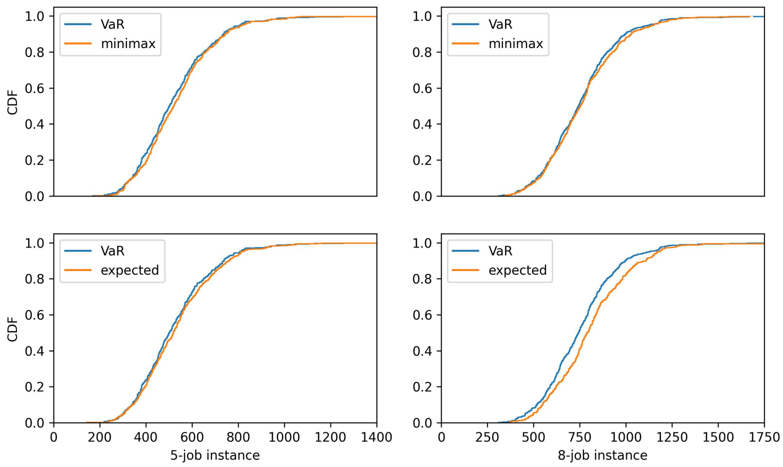

Figure 7 Comparison of the performances between VaR approach and alternative approaches on 1000 scenarios.

7 Conclusions

In this paper, the two-machine re-entrant flow shop scheduling problem with stochastic processing times has been investigated. The aim was the development of solution approaches able to devise a robust schedule, capable of protecting against the occurrence of unfavourable events, using the minimisation of the value-at-risk as a criterion for robustness. To estimate the makespan distribution, a Markovian Activity Network (MAN) approach was employed, leveraging phase-type distributions to align with the realistic distributions observed in industrial processes.

The proposed branch-and-bound approach demonstrated reasonable performance for small-scale problems, although being constrained by the increasing computational load as the dimension of the scheduling instances increases. Heuristics have been proposed to solve larger-scale problems, whose efficiency and effectiveness are demonstrated through a set of experiments.

The first direction for future development is enhancing computational efficiency. Compared to a similar branch-and-bound approach developed for the non-re-entrant version of this scheduling problem (Liu and Urgo 2023), the proposed one demonstrated reduced performance. This result opens the way to further investigations on the characteristics and complexity of the two problems. Looking at this from the point of view of the total number of possible solutions, the total number of alternative schedules for the re-entrant flow shop problem (e.g., in the case of $n+n$ jobs) is lower than the total number of possible schedules for its non-re-entrant version (i.e. with $2n$ jobs). This is due to the additional constraints linking the processing of first- and second-pass jobs, causing some schedules to be infeasible. In contrast, while a specific dominance rule can be applied to the problem non-re-entrant problem (Liu and Urgo 2023), it cannot unfortunately be used for the re-entrant problem (Sect. 4). Finally, both the Markov Chain models and the proposed lower bounds for the two problems differ, which might impact computational performance and, consequently, the overall solution time.

Further in this direction, the possible dominance between the two machines in the flow shop surely causes some problem instances to be extremely difficult to solve. This has been addressed for the deterministic version of the problem (Emmons and Vairaktarakis 2012), for which a machine dominance criterion has been defined. A similar criterion is not available for the stochastic version of the problem and, understanding the possible dominance among machines could surely guide the search for the optimal solution and reduce the solution time.

Finally, within this paper, the first-pass and second-pass jobs are treated as independent entities, with no consideration given to their similarity (Sect. 3). Consequently, exploring the correlation between these two task sets could offer valuable insights and enhancements for the proposed approach.

Future advancements of the proposed approach will also be focused on extending its applicability to a broader range of scheduling problems.

Data availability statement

The data that support the findings of this study are openly available in Figshare: Stochastic 2-m re-entrant flow shop scheduling instances.

Conflict of interest

The authors declare that they have no conflict of interest.

References

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., BÃijlbÃijl, K., and Noyan, N. (2016). Minimizing value-at-risk in single-machine scheduling. *Annals of Operations Research*, 248(1-2):25–73.

Baker, K. R. and Altheimer, D. (2012). Heuristic solution methods for the stochastic flow shop problem. *European Journal of Operational Research*, 216(1):172–177.

Baker, K. R. and Trietsch, D. (2010). Three heuristic procedures for the stochastic, two-machine flow shop problem. *Journal of Scheduling*, 14(5):445–454.

Benavides, A. J. and Vera, A. (2022). The reversibility property in a job-insertion tiebreaker for the permutational flow shop scheduling problem. *European Journal of Operational Research*, 297(2):407–421.

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

Bladt, M. and Yslas, J. (2022). Heavy-tailed phase-type distributions: a unified approach. *Extremes*, 25(3):529–565.

Boost (2020). Boost C++ Libraries. `http://www.boost.org/`.

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

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.

Choi, S.-W. and Kim, Y.-D. (2007). Minimizing makespan on a two-machine re-entrant flowshop. *Journal of the Operational Research Society*, 58(7):972–981.

Choi, S.-W. and Kim, Y.-D. (2008). Minimizing makespan on an m-machine re-entrant flowshop. *Computers & Operations Research*, 35(5):1684–1696.

Choi, S.-W. and Kim, Y.-D. (2009). Minimizing total tardiness on a two-machine re-entrant flowshop. *European Journal of Operational Research*, 199(2):375–384.

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

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.

Demirkol, E. and Uzsoy, R. (2000). Decomposition methods for reentrant flow shops with sequence-dependent setup times. *Journal of Scheduling*, 3(3):155–177.

Dixit, V. and Tiwari, M. K. (2020). Project portfolio selection and scheduling optimization based on risk measure: a conditional value at risk approach. *Annals of Operations Research*, 285(1-2):9–33.

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

Drobouchevitch, I. G. and Strusevich, V. A. (1999). A heuristic algorithm for two-machine re-entrant shop scheduling. *Annals of Operations Research*, 86(0):417–439.

Dugardin, F., Yalaoui, F., and Amodeo, L. (2010). New multi-objective method to solve reentrant hybrid flow shop scheduling problem. *European Journal of Operational Research*, 203(1):22–31.

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

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.

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.

Graves, S. C., Meal, H. C., Stefek, D., and Zeghmi, A. H. (1983). Scheduling of re-entrant flow shops. *Journal of Operations Management*, 3(4):197–207.

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

Horvath, G. and Telek, M. (2017). BuTools 2: a rich toolbox for markovian performance evaluation. In *Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools*. ACM.

Jeong, B. and Kim, Y.-D. (2014). Minimizing total tardiness in a two-machine re-entrant flowshop with sequence-dependent setup times. *Computers & Operations Research*, 47:72–80.

Juan, A. A., Keenan, P., Martí, R., McGarraghy, S., Panadero, J., Carroll, P., and Oliva, D. (2023). A review of the role of heuristics in stochastic optimisation: From metaheuristics to learnheuristics. *Annals of Operations Research*, 320(2):831–861.

Juvin, C., Houssin, L., and Lopez, P. (2023). Constraint programming for the robust two-machine flow-shop scheduling problem with budgeted uncertainty. In *20th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*.

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.

Lee, C. K., Lin, D., Ho, W., and Wu, Z. (2011). Design of a genetic algorithm for bi-objective flow shop scheduling problems with re-entrant jobs. *The International Journal of Advanced Manufacturing Technology*, 56(9):1105–1113.

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. (2022a). A branch and bound approach for stochastic 2-machine flow shop scheduling with rework. In *18th International Conference on Project Management and Scheduling*.

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

Liu, L. and Urgo, M. (2023). A branch-and-bound approach to minimise the value-at-risk of the makespan in a stochastic two-machine flow shop. *International Journal of Production Research*.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2019). Iterated local search: Framework and applications. In *Handbook of metaheuristics*, pages 129–168. Springer.

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.

Manzini, M. and Urgo, M. (2015). Makespan estimation of a production process affected by uncertainty: Application on mto production of nc machine tools. *Journal of Manufacturing Systems*, 37:1–16.

Manzini, M. and Urgo, M. (2018). A risk based approach to support the supplying of components in a mto assembly process. *Journal of manufacturing systems*, 46:67–78.

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.

Nawaz, M., Enscore Jr, E. E., and Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91–95.

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

Pan, J.-H. and Chen, J.-S. (2003). Minimizing makespan in re-entrant permutation flow-shops. *Journal of the operational Research Society*, 54(6):642–653.

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

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.

Sidje, R. B. and Stewart, W. J. (1999). A numerical study of large sparse matrix exponentials arising in markov chains. *Computational Statistics & Data Analysis*, 29(3):345–368.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *european journal of operational research*, 64(2):278–285.

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 VÃąncza, 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.

Wang, L., Zhang, L., and Zheng, D.-Z. (2005a). A class of hypothesis-test-based genetic algorithms for flow shop scheduling with stochastic processing time. *The International Journal of Advanced Manufacturing Technology*, 25(11):1157–1163.

Wang, L., Zhang, L., and Zheng, D.-Z. (2005b). Genetic ordinal optimisation for stochastic flow shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 27(1):166–173.

Yu, T.-S. and Pinedo, M. (2020). Flow shops with reentry: Reversibility properties and makespan optimal schedules. *European Journal of Operational Research*, 282(2):478–490.

Appendix A

For the representative 5(10)-job and 8(16)-job instances in Sect. 6.3, the phase-type distributions of job processing times on each machine are reported in Table 7 and 8. Each distribution is denoted by an initial vector $\boldsymbol{\beta}$ and a matrix T (Neuts 1994).

Table 7 Distributions of the processing times for the representative 5-job instances.

job j		distribution p_{ja}	distribution p_{jb}
1	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.354 & 0.125 \\ 0.151 & -0.151 \end{bmatrix}$	[-0.028]
2	β	[1 0]	[1 0]
	T	$\begin{bmatrix} -0.036 & 0.033 \\ 0 & -0.048 \end{bmatrix}$	$\begin{bmatrix} -0.24 & 0.044 \\ 0.175 & -0.175 \end{bmatrix}$
3	β	[1 0]	[1 0]
	T	$\begin{bmatrix} -0.03 & 0.012 \\ 0.032 & -0.032 \end{bmatrix}$	$\begin{bmatrix} -0.053 & 0.044 \\ 0 & -0.039 \end{bmatrix}$
4	β	[1 0 0 0]	[1]
	T	$\begin{bmatrix} -0.503 & 0.171 & 0.149 & 0.048 \\ 0.214 & -0.214 & 0 & 0 \\ 0 & 0.129 & -0.129 & 0 \\ 0 & 0 & 0.028 & -0.028 \end{bmatrix}$	[-0.05]
5	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.252 & 0.134 \\ 0 & -0.015 \end{bmatrix}$	[-0.166]
1'	β	[1]	[1 0]
	T	[-0.027]	$\begin{bmatrix} -0.185 & 0.185 \\ 0.485 & -0.572 \end{bmatrix}$
2'	β	[1]	[1 0]
	T	[-0.025]	$\begin{bmatrix} -0.259 & 0.166 \\ 0.084 & -0.084 \end{bmatrix}$
3'	β	[1]	[1 0]
	T	[-0.051]	$\begin{bmatrix} -0.043 & 0.013 \\ 0.05 & -0.05 \end{bmatrix}$
4'	β	[1]	[1]
	T	[-0.022]	[-0.21]
5'	β	[1]	[1 0 0 0]
	T	[-0.013]	$\begin{bmatrix} -0.128 & 0.044 & 0.012 & 0.03 \\ 0.061 & -0.061 & 0 & 0 \\ 0.051 & 0 & -0.051 & 0 \\ 0 & 0 & 0.088 & -0.088 \end{bmatrix}$

Table 8 Distributions of the processing times for the representative 8-job instances.

job j		distribution p_{ja}	distribution p_{jb}
1	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.313 & 0.292 \\ 0 & -0.224 \end{bmatrix}$	[-0.057]
2	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.037 & 0.026 \\ 0 & -0.025 \end{bmatrix}$	[-0.057]
3	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.057 & 0.014 \\ 0 & -0.03 \end{bmatrix}$	[-0.028]
4	β	[1]	[1 0]
	T	[-0.2]	$\begin{bmatrix} -0.047 & 0.014 \\ 0 & -0.032 \end{bmatrix}$
5	β	[1]	[1 0]
	T	[-0.04]	$\begin{bmatrix} -0.024 & 0.024 \\ 0.049 & -0.504 \end{bmatrix}$
6	β	[1]	[1 0]
	T	[-0.013]	$\begin{bmatrix} -0.081 & 0.063 \\ 0 & -0.117 \end{bmatrix}$
7	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.077 & 0.034 \\ 0 & -0.007 \end{bmatrix}$	[-0.148]
8	β	[1 0 0 0]	[1 0]
	T	$\begin{bmatrix} -0.052 & 0.041 & 0 & 0.01 \\ 0 & -0.068 & 0.007 & 0 \\ 0 & 0 & -0.072 & 0.032 \\ 0 & 0 & 0.064 & -0.064 \end{bmatrix}$	$\begin{bmatrix} -0.033 & 0.024 \\ 0 & -0.017 \end{bmatrix}$
1'	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.124 & 0.078 \\ 0.07 & -0.07 \end{bmatrix}$	[-0.23]
2'	β	[1]	[1]
	T	[-0.014]	[-0.035]
3'	β	[1]	[1 0]
	T	[-0.032]	$\begin{bmatrix} -0.896 & 0.497 \\ 0.187 & -0.187 \end{bmatrix}$
4'	β	[1]	[1 0]
	T	[-0.061]	$\begin{bmatrix} -0.028 & 0.004 \\ 0.016 & -0.016 \end{bmatrix}$
5'	β	[1]	[1 0]
	T	[-0.015]	$\begin{bmatrix} -0.018 & 0.006 \\ 0 & -0.072 \end{bmatrix}$
6'	β	[1]	[1 0]
	T	[-0.06]	$\begin{bmatrix} -0.243 & 0.187 \\ 0.075 & -0.075 \end{bmatrix}$
7'	β	[1 0]	[1]
	T	$\begin{bmatrix} -0.138 & 0.114 \\ 0 & -0.075 \end{bmatrix}$	[-0.014]
8'	β	[1]	[1 0 0 0]
	T	[-0.13]	$\begin{bmatrix} -0.052 & 0.041 & 0 & 0.01 \\ 0 & -0.068 & 0.007 & 0 \\ 0 & 0 & -0.072 & 0.032 \end{bmatrix}$