Computational Management Science
https://doi.org/10.1007/s10287-020-00386-1

ORIGINAL PAPER

Stochastic single machine scheduling problem as a multi-stage dynamic random decision process

Mina Roohnavazfar[1,2] · Daniele Manerba[3] · Lohic Fotio Tiotsop[1] · Seyed Hamid Reza Pasandideh[2] · Roberto Tadei[1]

Received: 29 December 2019 / Accepted: 23 December 2020

Abstract
In this work, we study a stochastic single machine scheduling problem in which the features of learning effect on processing times, sequence-dependent setup times, and machine configuration selection are considered simultaneously. More precisely, the machine works under a set of configurations and requires stochastic sequence-dependent setup times to switch from one configuration to another. Also, the stochastic processing time of a job is a function of its position and the machine configuration. The objective is to find the sequence of jobs and choose a configuration to process each job to minimize the makespan. We first show that the proposed problem can be formulated through two-stage and multi-stage Stochastic Programming models, which are challenging from the computational point of view. Then, by looking at the problem as a multi-stage dynamic random decision process, a new deterministic approximation-based formulation is developed. The method first derives a mixed-integer non-linear model based on the concept of accessibility to all possible and available alternatives at each stage of the decision-making process. Then, to efficiently solve the problem, a new accessibility measure is defined to convert the model into the search of a shortest path throughout the stages. Extensive computational experiments are carried out on various sets of instances. We discuss and compare the results found by the resolution of plain stochastic models with those obtained by the deterministic approximation approach. Our approximation shows excellent performances both in terms of solution accuracy and computational time.

Keywords Single machine scheduling · Stochastic sequence-dependent setup times · Stochastic processing times · Learning effect · Deterministic approximation · Multi-stage stochastic programming

✉ Daniele Manerba
daniele.manerba@unibs.it

Extended author information available on the last page of the article

Published online: 03 February 2021

1 Introduction

Single machine scheduling is a decision-making process that plays a critical role in all manufacturing and service systems. This problem has been extensively investigated for a long time because of its practical importance in developing scheduling theory in more complex job shops and integrated processes. The flow time, total tardiness, or makespan minimization are the main frequently used criteria. Here, the same machine can be used for different jobs, but the efficiency of processing them depends on the configuration used. In general, switching from one configuration to another one implies a so-called setup time.

In machine scheduling problems, the setup times are considered either sequence-independent or sequence-dependent. In the former case, the setup times are negligible or assumed to be a part of job processing times while, in the latter case, the setup times depend on not only the job currently being scheduled but also the last scheduled job. Sequence-dependent setup time between two different activities is encountered in many industries such as the printing industry, paper industry, automotive industry, chemical processing, and plastic manufacturing industry. Dudek et al. (1974) reported that 70% of industrial activities include sequence-dependent setup times.

Another realistic aspect to consider is that the efficiency of workers or machines increases depending on the time spent by the jobs or repetition of activities in many manufacturing and service industries. Therefore, the actual processing time of a job could be shorter if it is scheduled at the end of a queue. This phenomenon, known as *learning effect*, has been observed in various practical situations in several branches of industry and for a variety of activities (Yelle 1979; Gawiejnowicz 1996). Depending on the production system, learning effects can be based on position or on the sum of processing times. In the former case, they depend only on the number of jobs being processed, while in the latter case, they depend on the sum of the processing times of the already processed jobs. In this study, single machine scheduling with position-dependent learning effects is considered.

It is also essential to notice that the manufacturing and service systems operate under uncertainty in many realistic situations. The uncertain environment stems from a variety of random events, such as machine breakdown, job cancellation, rush orders, and inaccurate expected job information. The majority of the works in the literature on stochastic single machine scheduling mainly consider uncertain job processing time. However, in some real-world situations, the setup times may also be uncertain due to some random factors like crew skills, tools and setup crews, or unexpected breakdown of fixtures and tools. Although it is a typical case in several industries, we found only a few studies that consider stochastic sequence-dependent setup times (see, e.g., Allahverdi 2015). Some real-life examples include various practices such as the use of checklists during the setup of the machine, the use of documented standard operating procedures, the use of devices to reduce operator errors during the setup, and the use of training. In some cases, setup times increase due to unplanned maintenance, which in general needs more setup activities such as the replacement of worn-out tools. These examples indicate that variability in setup time is a real concern in practice.

In this study, we consider a scheduling problem on a single machine with a set of configurations that can perform the jobs in various processing times. Each job

has a deterministic processing time, which has been affected by the job-dependent learning effect and the selected machine configuration. The deterministic setup time of switching the machine from a configuration to another is sequence-dependent. A random variable associated with the job attributes, including both processing time and setup time between the machine configurations, is defined. The problem objective is to determine the sequence of jobs and choose the configuration to process each job such that the makespan is minimized. Our proposed problem belongs to $\mathcal{NP}$-hard class since the deterministic single machine scheduling with sequence-dependent setup times is $\mathcal{NP}$-hand as well (Baker and Trietsch 2009). To the best of our knowledge, there is only one paper which studies a quite similar problem. The work is proposed by Soroush (2014), which addresses the single machine scheduling in which processing times, setup times, and reliabilities are sequence-dependent random variables. These attributes are also subjected to different job-dependent and position-based learning effects. They explored various scenarios of the problem when the cost functions are linear, exponential, and fractional in different pairs of criteria consisting of the makespan, total absolute variations in completion times, total waiting time and their weighted counterparts, and the total reliability/unreliability. It has been proved that the scenarios can be modeled as quadratic assignment problems, which are solvable exactly or approximately. Also, the special cases with sequence-independent setup times and either position-independent or sequence-independent reliabilities can be solved optimally in polynomial time.

We address the above problem by using three methods: a two-stage Stochastic Programming (SP) model, a multi-stage SP one, and a Deterministic Approximation (DA)-based approach. SP is one of the main existing paradigms to deal with uncertain data and assumes that the random input data follow given probability distributions and pursues optimality in the average sense, adopting a risk-neutral perspective. However, it is difficult to obtain the probabilistic distribution of input data in practice. Even if an estimate of such a distribution is available, many scenarios are necessarily needed to approximate it accurately. Unfortunately, as the number of scenarios increases the problem becomes more complicated since the number of decision variables and constraints grows up. To deal with these drawbacks, we model the problem as a multi-stage dynamic random decision process where the knowledge of the probabilistic distribution of uncertain data is not needed. According to this perspective, the problem is seen as a multi-stage decision process in which jobs and configurations are chosen step by step to achieve an optimal sequence eventually. According to Tadei et al. (2020), the probability distribution of the best choice in terms of the next processed job and of the machine configuration can be asymptotically approximated by using results from the Extreme Value Theory (Galambos 1994). In turn, the makespan of the process can be analytically derived. From now on, we will refer to this framework to the Extreme Value Theory-based Deterministic Approximation (EVTDA). Using this approach, the problem is first reformulated as a mixed-integer non-linear programming model. Then, a simplification of such a model is provided to achieve a linear formulation. More precisely, it is shown that the optimal scheduling can be found by efficiently solving a linear shortest path problem in a multi-stage framework.

The contribution of this study is twofold. First, two-stage and multi-stage SP formulations are derived to model a single machine scheduling involving both stochastic

sequence-dependent setup times and stochastic position-dependent processing times, affected by learning effect. Second, approximated makespan and optimal solutions are found by proposing an innovative deterministic model which, for the first time in the literature, embeds the calculation of estimators and parameters coming from the EVTDA approach. This approach provides a powerful decision support tool that overcomes the computational burden of solving fat stochastic programs that depend on the number of scenarios considered and can be derived even when the probability distribution of the uncertainties is unknown.

This paper is organized as follows. In Sect. 2, a literature review of the problem is given. Sect. 3 describes the problem and formulates it using the two-stage and multi-stage SP models. In Sect. 4, we propose a solution approach based on a deterministic approximation recently introduced in the literature. We first derive the approximation, and, based on that, we then formulate the problem as a non-linear model. Eventually, by defining a new accessibility measure, the model is converted into a shortest path problem. In Sect. 5, the computational results over several simulated instances are proposed. Finally, conclusions are given in Sect. 6.

2 Literature review

The first research on job scheduling problems was performed in the mid-1950s. Since then, thousands of papers on different scheduling problems have appeared in the literature. In the manufacturing industries, the machine environment is generally considered as the resource of scheduling problems. The production system sometimes includes a machine bottleneck, which affects, in some cases, all the jobs. Since the management of this bottleneck is crucial, the single machine scheduling problem has been gaining importance for a long time. Here, we explored the scheduling literature within the single machine scheduling problems. The surveys by Yen and Wan (2003), Pinedo (2012), Adamu and Adewumi (2014), and the work proposed by Leksakul and Techanitisawad (2005) have detailed the literature on the theory and applications about this problem in the past several decades.

The majority of papers assumed sequence-dependent setup times, which occur in many different manufacturing environments. Angel-Bello et al. (2011) addressed the single machine scheduling with sequence-dependent setup times and maintenance with the aim of makespan minimizations. They developed a MIP model, as well as a linear relaxation of it and an efficient heuristic approach to solve large instances. Kaplanoglu (2014) addressed the case of dynamic job arrivals. He developed a collaborative multi-stage optimization approach. Bahalke et al. (2010) proposed a tabu search and genetic algorithm to deal with the single machine scheduling with sequence-dependent setup times and deteriorating jobs. Stecco et al. (2008) deal with the single machine scheduling where the setup time is not only sequence-dependent but also time-dependent. They developed a branch-and-cut algorithm that solves the instances up to 50 jobs. Ying and Bin Mokhtar (2012) addressed this problem with the secondary objective of minimizing the total setup time where jobs dynamically arrive. They proposed a heuristic algorithm based on the dynamic scheduling system. There are numerous studies on single machine scheduling with sequence-dependent setup

times and various performance measures, such as minimizing total flow time, tardiness, lateness, or waiting time. We refer the reader to Allahverdi et al. (1999) and Allahverdi (2015) for a comprehensive survey of the models, applications, and algorithms.

As Allahverdi (2015) emphasizes, the literature on stochastic sequence-dependent setup times is scarce. However, in some real-world situations, setup times may be uncertain as a result of random factors such as crew skills, tools and setup crews, and unexpected breakdown of fixtures and tools. In the literature of single machine scheduling, we found only a few papers that consider sequence-dependent setup times as random variables. Lu et al. (2012) addressed a robust single machine scheduling problem with uncertain job processing times and sequence-dependent family setup times. They formulated the problem as a robust constrained shortest path problem and solved by a simulated annealing-based heuristic. The objective was minimizing the absolute deviation of total flow time from the optimal solution under the worst-case scenario. Also, the interval data were used to generate uncertain parameters of sequence-dependent setup times. Ertem et al. (2019) focused on single machine scheduling with stochastic sequence-dependent setup times to minimize the total expected tardiness. They proposed a two-stage SP and the sample average approximation (SAA) method to model and solve the problem. The genetic algorithm is used to solve more significant size problems. Soroush (2014) deals with position and sequence-dependent setup times in the single machine scheduling under uncertain job attributes, including processing time and setup times.

Many researchers have been focused on various scheduling problems with learning effect on processing times. Biskup (1999) demonstrated that the makespan minimization on single machine scheduling with position-based learning could be optimally solved in polynomial time by using the shortest processing time (SPT) rule. Since then, many researchers have focused on scheduling with a position-based learning model and various performance measures. The most well-known ones include those of Mosheiov (2001), Lee et al. (2004), Zhao et al. (2004), and Kuo and Yang (2007), Mustu and Eren (2018). A comprehensive review of different kinds of learning effects is proposed by Azzouz et al. (2018). Some extensions of the basic position-based learning model have been presented, including the consideration of job-dependent position-based learning effects (Yin et al. 2010; Cheng et al. 2008), autonomous position-based and induced learning effects (Zhang et al. 2013; Huo et al. 2018), position-based learning and deteriorating effects (Toksari and Guner 2009; Cheng et al. 2011; Sun 2009), and both position-based and sum-of-the-processing-time-based learning effects (Yang and Yang 2011; Cheng et al. 2013).

In the above studies, the processing time is assumed to be a deterministic value. However, real-world manufacturing and service systems usually work under uncertain environment due to various random interruptions. The ignorance of uncertainty yields schedules that cannot be readily executed in practice. Most works in the literature of stochastic single machine scheduling mainly study uncertain processing time. Various objective functions have been considered, such as minimizing expected total tardiness, earliness and tardiness penalty costs, expected number of tardy jobs, expected total weighted number of early and tardy jobs, expected value of the sum of a quadratic cost function of idle time and the weighted sum of a quadratic function of job lateness, mean completion time and earliness and tardiness costs, worst-case conditional value

at risk of the job sequence total flow time, total weighted completion time or the total weighted tardiness (Ertem et al. 2019). Hu et al. (2015) used uncertainty theory to study the single machine scheduling problem with deadlines and stochastic processing times with known uncertainty distributions. The aim is to derive a deterministic integer programming model by using the operational law for inverse uncertainty distributions to maximize the expected total weight of batches of jobs. Pereira (2016) addressed single machine scheduling under a weighted completion time performance metric in which the processing times are uncertain, but can only take values from closed intervals. The objective is to minimize the maximum absolute regret for any possible realization of the processing times. An exact branch-and-bound method to solve the problem has been developed. Seo et al. (2005) studied single machine scheduling to minimize the expected number of tardy jobs. The jobs usually have distributed processing times and a common deterministic due date. They proposed a non-linear integer programming model and some relaxations to approximately solve it.

Limited work exists to address problems with both learning effect and uncertainty on processing time in the single machine context. Li (2016) addressed the single machine scheduling problem with random nominal processing time, or random job-based learning rate, or both, to minimize the expected total flow time and expected makespan. It was shown that the shortest expected processing time (SEPT) rule is optimal for minimizing the expected total flow time or makespan in the position-based learning model with only job processing time being random. For the job-based learning model, it was proved that minimizing the expected total flow time or makespan is equivalent to solve a random assignment problem with uncertain assignment costs. Zhang et al. (2013) studied the single machine scheduling problem with both learning effect and uncertain processing time. They proved that the SEPT rule is optimal for minimizing the expected makespan and maximum lateness. Also, they studied the case with stochastic machine breakdowns.

Concerning the need to take into account uncertain data, researchers have applied various methodologies to achieve optimal solutions, such as Robust Optimization and Stochastic Programming (Maggioni et al. 2017). In Robust Optimization (RO), uncertain data are represented using continuous intervals, and the aim is to optimize the performance measure in the worst-case scenario. Lu et al. (2012) studied the robust single machine scheduling with uncertain processing time and sequence-dependent family setup time represented by interval data. The objective is to minimize the absolute deviation of total flow time from the optimal solution of the worst-case scenario. They formulated the problem as a robust constrained shortest path problem and solved by a simulated annealing algorithm that embeds a generalized label correcting method. Daniels and Kouvelis (1995) addressed the single machine scheduling with the uncertain processing time and objective of total flow time. They used a branch-and-bound algorithm and two surrogate relation heuristics to find robust schedules. Yang and Yu (2002) studied the same problem but with a discrete finite set of processing time scenarios rather than interval data. They developed an exact dynamic programming algorithm and two heuristics to obtain robust schedules. Stochastic Programming (SP) is another approach to tackle machine scheduling in which job attributes (e.g., processing time, release time, setup time, due dates) follow given probability distributions and reach optimality in the average sense. Some studies indicate that the SP models

of single machine scheduling are $\mathcal{NP}$-hard under certain distributional assumptions of job processing time. For example, Soroush (2007) addressed static stochastic single machine scheduling problem in which jobs have random processing times with arbitrary distributions, known due dates with certainty, and fixed individual penalties imposed on both early and tardy jobs. He showed that the problem is $\mathcal{NP}$-hard and developed certain conditions under which the problem is solvable exactly. In the literature, various performance measures, such as flow time (Agrawala et al. 1984; Lu et al. 2010), maximum lateness (Cai et al. 2007), the number of late jobs (van den Akker and Hoogeveen 2008; Seo et al. 2005; Trietsch et al. 2008), weighted number of early and tardy jobs (Soroush 2007), and the total tardiness (Ertem et al. 2019; Ronconi and Powell 2010), have been addressed.

It is well-known that explicitly addressing uncertainty in an optimization problem generally poses significant computational challenges. Therefore, another line of research is to see whether it is possible to incorporate uncertainty in an approximated way and convert the stochastic model into a deterministic one. Tadei et al. (2020) have recently proposed the EVTDA approach to deal with uncertainty in a multi-stage decision-making process, where the knowledge of the probability distribution of uncertain data is not required. Approaches leveraging the Extreme Value Theory for deterministically approximating stochastic optimization problems have empirically proved to be effective (Perboli et al. 2012, 2014; Tadei et al. 2009). Roohnavazfar et al. (2019) relied on such an approach for finding the optimal sequence of choices in a multi-stage stochastic structure and compared the obtained solution with that of the expected value problem. The results pointed out the suitability of the method.

3 Problem definition and mathematical formulation

The problem proposed in this paper aims at sequencing a set of jobs that require to be processed on a single machine so to minimize the makespan. The considered machine can handle one job per time and works under a set of operating modes (from now on simply called *configurations*), which affect the job processing times. No pre-emption is allowed, i.e., a job must be completed without interruptions once it is started to be processed. Sequence-dependent setup times are assumed when the machine switches from a configuration to another one. Finally, taking into account the learning effect, job processing times also depend on the job positions in the sequence. Let us consider the following notation:

- $I = \{1, 2, \ldots, n\}$: set of jobs;
- $R = \{1, 2, \ldots, n\}$: set of positions in the job sequence;
- $F = \{0, 1, 2, \ldots, m\}$: set of possible machine configurations. Note that 0 indicates a dummy initial configuration of the machine;
- $\alpha < 0$: learning effect rate;
- P_i^k: nominal processing time of job i under configuration k;
- $P_{ir}^k := P_i^k \cdot r^\alpha$: deterministic processing time of job i processed at position r under configuration k;
- S_r^{jk}: deterministic sequence-dependent setup time of the machine to switch from configuration j to configuration k at position r.

To achieve a more realistic representation of manufacturing and service systems, a stochastic oscillation associated with the machine setup time and the job processing time is considered. This oscillation can be represented by a random variable, defined over a given probability space. As commonly done in the literature (see, e.g., Birge and Louveaux 2011), we approximate the distribution of the random variables through a sufficiently large set of realizations. Therefore, let us consider the following additional notation:

- L_{ir}^{jk}: set of scenarios of random oscillations related to processing job i at position r under configuration k after switching from configuration j;
- L_r: set of scenarios associated with position r;
- π^l: probability of each scenario $l \in L_{ir}^{jk}$;
- $\tilde{\theta}_{ir}^{jkl}$: realization of the random oscillation of the time when processing job i at position r and configuration k, when the machine is switched from configuration j, under scenario $l \in L_{ir}^{jk}$.

In the following, we provide two different mathematical formulations for the problem described, using the well-known two-stage and multi-stage SP paradigms.

3.1 Two-stage Stochastic Programming formulation

The proposed problem can be formulated as a mixed-integer linear model by using a two-stage SP model in which the first-stage variables are those that have to be decided before the actual realization of the uncertain parameters becomes available. Once the random events occur, the value of the second-stage (or *recourse*) variables can be decided.

The proposed problem contains two different types of operational decisions, namely, assigning jobs to positions and choosing a machine configuration to process each job. In our two-stage SP model, two variables sets corresponding to decisions before and after revealing information must be defined. The first-stage decisions, common to all realizations, represent the assignments of jobs to positions. The second-stage decisions, specific to each realization and dependent on the first-stage decisions, represent the choice of configuration to process each job. Obviously, in the two-stage model, the term *stage* is corresponding to the period before and after revealing information. Therefore, the first stage is completed when all the jobs are assigned to positions. Then the second stage starts as soon as the uncertainties are revealed.

We consider the following variables:

- y_{ir}: boolean variable equal to 1 if job i is assigned to position r, 0 otherwise;
- x_{ir}^{jkl}: boolean variable equal to 1 if job i is assigned to position r and processed under configuration k after switching from configuration j and scenario l, 0 otherwise.

Then, a two-stage SP model for the problem is as follows:

$$\min_{x} \sum_{i=1}^{n} \sum_{r=1}^{n} \sum_{j=0}^{m} \sum_{k=1}^{m} \sum_{l \in L_{ir}^{jk}} \pi^l (S_r^{jk} + P_{ir}^k + \tilde{\theta}_{ir}^{jkl})\, x_{ir}^{jkl} \tag{1}$$

subject to

$$\sum_{r=1}^{n} y_{ir} = 1 \quad \forall i \in I, \tag{2}$$

$$\sum_{i=1}^{n} y_{ir} = 1 \quad \forall r \in R, \tag{3}$$

$$\sum_{j=0}^{m}\sum_{k=1}^{m} x_{ir}^{jkl} = y_{ir} \quad \forall i \in I, r \in R, l \in L_{ir}^{jk}, \tag{4}$$

$$\sum_{i=1}^{n}\sum_{j=0}^{m} x_{ir}^{jkl} = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{i,r+1}^{kjl} \quad \forall k \in F\backslash\{0\}, r \in R\backslash\{n\}, l \in L_{ir}^{jk}, \tag{5}$$

$$\sum_{k=1}^{m} x_{i1}^{0kl} = y_{i1} \quad \forall i \in I, l \in L_{i1}^{0k}, \tag{6}$$

$$y_{ir} \in \{0, 1\} \quad \forall i \in I, r \in R, \tag{7}$$

$$x_{ir}^{jkl} \in \{0, 1\} \quad \forall i \in I, r \in R, j, k \in F, l \in L_{ir}^{jk}. \tag{8}$$

The objective function (1) expresses the minimum expected makespan. Constraints (2) and (3) ensure that each job must be selected to be processed in exactly one position, and in each position, exactly one job is assigned. Constraints (4) state that the machine is switched from a configuration to another one to process each job (it should be noted that the two consecutive configurations are not necessarily different from each other). These constraints are also linking the two types of decisions. Constraints (5) form the sequence (flow) of configurations to process all jobs over the positions. They establish the fact that it is possible to switch from a certain configuration k to another if and only if the machine is already under configuration k. Without this constraint, the sequence of configurations switch over positions would not be continuous. Constraint (6) indicate that the machine is switched from the initial configuration 0 to one of the configurations to process the job in the first position. Finally, (7) and (8) are binary constraints on the variables.

The proposed two-stage SP formulation, at first glance, might look quite simplistic. However, it can be used to model many practical situations exhaustively. Consider, for instance, the operations scheduling of a crane dedicated to loading/unloading of heavy items, such as containers or pallets. This machine works under different configurations and provides processing times that depend on the actual weight of the item to be managed. If a series of items is to be processed in the future, it may be essential for organizational issues of the company to estimate the makespan of the whole process managing before the items arrive. The precise weight of each item is not known a priori, but only an estimate of it is available. In this case, our two-stage formulation could be particularly suitable since, at the first stage, the decision-maker is asked to establish the processing order (which provides an estimate of the makespan). Then, only when the items arrive, the weight of each of them can be carefully measured and,

therefore, the best configuration under which each item should be managed can be decided.

Despite the applicability of the two-stage model, the strongly periodical structure of the considered problem naturally suggests that a multi-stage SP approach could be more suitable. Therefore, in the following, we also propose to model the uncertainty structure of the problem differently.

3.2 Multi-stage Stochastic Programming formulation

The proposed scheduling problem can also be modeled using a Multi-Stage Stochastic Programming formulation in which the decisions are taken stage by stage, along with the realizations of some random variables. Using the multi-stage SP paradigm, the uncertainty of random oscillations is dealt with a *scenario tree* as a branching structure representing the evolution of realizations over stages. In this context, we define a symmetric and balanced scenario tree. The jobs positions in our proposed problem correspond to stages in the SP approach in which the decisions on choosing jobs and configurations are taken. In a scenario tree, a path of realizations from the root node to a leaf node represents a scenario ω, occurring with probability π^{ω}. We call Ω the entire set of scenarios, i.e., the set of paths of realizations up to any leaf of the scenario tree. Two scenarios $\omega, \acute{\omega}$ are called *indistinguishable* at stage r if they share a common history of realizations until that stage, while after that they are represented by distinct paths. Also, each node o at stage r in the tree can be associated with a scenario group, represented as Ω_r^o, such that two scenarios that belong to the same group have the same realizations up to that stage. Moreover, the set of all the nodes at stage r is depicted as Φ_r.

Let us consider a boolean variable $x_{ir}^{jk\omega}$ equal to 1 if job i is assigned to position r and processed under configuration k after switching from configuration j under scenario ω, and 0 otherwise. Then, a multi-stage SP model for the problem is as follows:

$$\min_{x} \sum_{\omega\in\Omega} \pi^{\omega} \sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{j=0}^{m}\sum_{k=1}^{m}(S_r^{jk} + P_{ir}^{k} + \tilde{\theta}_{ir}^{jk\omega})\, x_{ir}^{jk\omega} \tag{9}$$

subject to

$$\sum_{r=1}^{n}\sum_{j=0}^{m}\sum_{k=1}^{m} x_{ir}^{jk\omega} = 1 \quad \forall i \in I, \omega \in \Omega, \tag{10}$$

$$\sum_{i=1}^{n}\sum_{j=0}^{m}\sum_{k=1}^{m} x_{ir}^{jk\omega} = 1 \quad \forall r \in R, \omega \in \Omega, \tag{11}$$

$$\sum_{i=1}^{n}\sum_{j=0}^{m} x_{ir}^{jk\omega} = \sum_{i=1}^{n}\sum_{t=1}^{m} x_{i,r+1}^{kt\omega} \quad \forall k \in F\backslash\{0\}, r \in R\backslash\{n\}, \omega \in \Omega, \tag{12}$$

$$x_{ir}^{jk\omega} = x_{ir}^{jk\acute{\omega}} \quad \forall i \in I, j, k \in F, r \in R\backslash\{n\}, \omega, \acute{\omega} \in \Omega_r^o, o \in \Phi_r, \tag{13}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{m} x_{i1}^{0k\omega} = 1 \quad \forall \omega \in \Omega, \tag{14}$$

$$x_{ir}^{jk\omega} \in \{0,1\} \quad \forall i \in I, r \in R, j,k \in F, \omega \in \Omega. \tag{15}$$

The objective function (9) expresses the minimum expected makespan. Constraints (10) ensure that each job must be selected to be processed in exactly one position under a switched configuration. In contrast, constraints (11) indicate that exactly one job and switched configurations are assigned to each position. Constraints (12) imposes the continuity of the sequence of configurations switch, as explained for constraint (5). Constraints (13) indicate specific non-anticipativity conditions which ensure that, for any pair of scenarios with the same history of realizations up to stage (position) r, the decisions must be the same. These constraints imply that the decisions taken at any stage do not depend on future realizations of uncertainty, but they are affected by the previous realizations of uncertainty as well as the knowledge of previous decisions. Constraints (14) indicate that the machine is switched from the initial configuration 0 to one of the configurations to process the job assigned to the first position. Finally, (15) are binary conditions on the variables.

4 EVTDA modeling approach

In the previous sections, we proposed a couple of formulations according to the two-stage and multi-stage stochastic programming paradigms. It is important to note that the two models handle uncertainty differently. While the two-stage model has a kind of invest-and-use perspective, the greater flexibility offered by the multi-stage SP allows us to deal with the problem more operationally. In particular, in the two-stage case, the whole sequence is defined under uncertainty. In contrast, in the multi-stage case, the problem is formulated, such as to allow, as new information becomes available, simultaneous optimization of both the job sequence and the configuration choice. However, these two models are computationally demanding. So, in this section, we derive an effective deterministic approximation of the multi-stage SP formulation of the problem that relies on the EVTDA approach, to make possible the resolution of large scale instances.

4.1 Estimation of the expected completion times

To derive a deterministic model that approximates the multi-stage SP one, we interpret the proposed problem as a stochastic multi-stage choice process, i.e., a decision process in which the decision-maker has to choose, stage by stage, the job to process and the configuration to use to minimize the expected time needed to process the remaining jobs, i.e., the *completion time*. Therefore, we aim to obtain at each stage a completion time estimator that depends on the chosen job and configuration. However, obtaining such an estimator is not trivial at all, since the expected completion time at each stage depends on the probability distribution of the oscillations, which is assumed to be

unknown. Hence, we decided to leverage the EVTDA approach to derive such an estimator at each stage. Note that, this way, the estimated expected completion time at the dummy stage corresponds to the expected makespan. As one should expect, the derived estimators (both for the makespan and the intermediate completion times) depend on the sequence of the jobs chosen over the stages and on the configurations used for processing them. Hence, once obtained the expression of these estimators, we derive an innovative non-linear scheduling model that aims at minimizing the makespan and that represents a deterministic approximation of the multi-stage SP formulation.

More precisely, at each stage (i.e., in each position), the decision maker faces a set of choices including the combinations of jobs and machine configurations to choose from for the next stage. At stage r, the choice of the combination made up by job i and configuration k switching from configuration j incurs a total cost $\tilde{t}_{ir}^{jk}$ constituted by the deterministic configurations setup time S_r^{jk}, the processing time P_{ir}^k, and finally the expected completion time T_{ir}^k of the remaining jobs to be processed in future positions.

To derive a deterministic approximation of the expected completion times T_{ir}^k, the EVTDA assumes that the decision-maker has an optimistic vision of the future. Not knowing what will happen in the future, he assumes that he will have to pay as little as possible. Then, from this assumption, it is derived an estimate of the expected future cost, which, however, depends on a parameter whose appropriate calibration allows to mitigate the risk associated with the initial optimistic attitude. Therefore, in our context, it is optimistically assumed that we have to incur in the following stochastic time:

$$\tilde{t}_{ir}^{jk} = \min_{l \in L_{ir}^{jk}} [S_r^{jk} + P_{ir}^k + \theta_{ir}^{jkl} + T_{ir}^k], \tag{16}$$

where θ_{ir}^{jkl} is a random variable, representing the time oscillation, whose realizations are $\{\tilde{\theta}_{ir}^{jkl}\}$. Please note that the completion times T_{ir}^k as well as the makespan are random variables since they depend on the realizations of the random time oscillation in future stages. Still following the optimistic perspective of the EVTDA, the expected completion times after each stage could be expressed as follows:

$$T_{ir}^k = \begin{cases} \mathbb{E}_{\tilde{\theta}} \left[\min_{\acute{j} \in F, h \in \tilde{I}_{ir}} \tilde{t}_{h,r+1}^{k\acute{j}} \right], & i \in I, k \in F \backslash \{0\}, r \in R \backslash \{n\} \\ 0, & r = n \end{cases} \tag{17}$$

where $\tilde{I}_{ir}$ denote the set of jobs that are still to be considered after having processed the job i at position r.

The distribution of the random cost T_{ir}^k is not known because it depends on future realizations of the random oscillations whose probability distribution is unknown. However, under the following assumptions:

1. the random oscillations θ_{ir}^{jkl} are independent and identically distributed (i.i.d.) according to an unknown survival function $F(x) = \mathbb{P}(\theta_{ir}^{jkl} > x)$;

2. $F(x)$ has an asymptotic exponential behavior in its left tail, i.e.,

$$\exists \beta > 0 \text{ such that } \lim_{y \to -\infty} \frac{1 - F(x+y)}{1 - F(y)} = e^{\beta x}; \tag{18}$$

the following two main results can be derived from the EVTDA:

1. T_{ir}^{k} can be approximated by:

$$T_{ir}^{k} = -\frac{1}{\beta}(\ln A_{ir}^{k} + \gamma) \qquad \forall i \in I, k \in F \backslash \{0\}, r \in R \backslash \{n\}, \tag{19}$$

where $\gamma \simeq 0.5772$ is the Euler constant and A_{ir}^{k} is the *accessibility* in the sense of Hansen (1959) to the overall set of choices at position $r+1$ when it is reached from job i and configuration k at position r and it is calculated as

$$A_{ir}^{k} = \sum_{h=1}^{n} \sum_{j=1}^{m} \alpha_{h,r+1}^{kj} \exp^{-\beta\left(S_{r+1}^{kj} + P_{h,r+1}^{j} + T_{h,r+1}^{j}\right)} \quad \forall i \in I, k \in F \backslash \{0\}, r \in R \backslash \{n\}, \tag{20}$$

where $\alpha_{ir}^{kj} = \frac{|L_{ir}^{kj}|}{|L_r|}$ indicates the proportion of the number of available scenarios associated with the job i and configuration k at position r with respect to the total number of scenarios associated with position r;

2. the expected makespan T_0 of all jobs can be estimated by:

$$T_0 = -\frac{1}{\beta}(\ln A_0 + \gamma), \tag{21}$$

where A_0 is the accessibility to all the choices and realizations in the first stage of decision the making process.

To fully understand the role of parameter β, it must be remembered that the entire decision-making process consists of as many stages as there are jobs. At each stage, a choice must be made between different alternatives. Each alternative being made by a couple formed by a job and a configuration under which the job will be processed. Each alternative is therefore associated with a stochastic cost as reported in (16). Being the cost associated with each alternative stochastic, the choice among them is not trivial. The β parameter aims at capturing the diversity in terms of cost and, thus, the probability of choosing among the available alternatives while taking into account the stochastic nature of their costs. It, therefore, allows to model more effectively the effect of uncertainty on the completion time estimator while minimizing any risks associated with the optimistic nature of the EVTDA approach.

It is important to notice that the assumption needed to apply the proposed EVTDA approach are quite mild and, in any case, less restrictive of those required by the SP paradigm. Interesting enough, Fadda et al. (2020) have recently proved that the EVTDA can still be deployed when assumption (18) is relaxed to the following condition:

$$\exists \beta > 0, a_{|L_r|} \text{ such that } \lim_{|L_r| \to +\infty} (F(x + a_{|L_r|}))^{|L_r|} = exp^{-e^{\beta x}}. \tag{22}$$

Assumption (22) is equivalent to ask the unknown distribution of the random oscillations to belong to the *domain of attraction* of a Gumbel distribution. The Gumbel domain of attraction describes distributions with light tails, i.e., probability distributions whose tails decrease exponentially. This assumption enlarges the set of distributions for which the EVTDA approach can be deployed in practice. It can be shown that such assumption is satisfied by widely used distributions such as the Normal, the Gumbel, the exponential, the Weibull, the Logistic, the Laplace, the Log-normal, and any cumulative distribution in the form $1-e^{-p(x)}$, where $p(x)$ is a positive polynomial function. Thus, this is a very mild assumption and, therefore, does not significantly restrict the deployment of the EVTDA approach. It is essential to notice that some empirical results presented in Tadei et al. (2017), Fadda et al. (2020), and Roohnavazfar et al. (2019) have demonstrated the effectiveness of the EVTDA framework even when the distribution of random oscillations does not satisfy the assumption (22) like the Uniform distribution.

The theoretical result underlining the EVTDA approach considers a single β parameter for the whole decision-making process. In practice, however, the problem of how to calibrate this parameter so that it captures the effects of uncertainty in a highly accurate way is still open. Therefore, to obtain a robust solution, we investigate the possibility of using one β parameter for each alternative. Although this approach makes the model less selfish, it allows us to directly model the dispersion of the realizations of the random oscillation associated with each alternative and, therefore, to derive a more robust estimate of the expected completion times. A similar approach has been profitably adopted by Roohnavazfar et al. (2019).

Using one β parameter per alternative, the expected completion times in (19) are now approximated by:

$$T_{ir}^{k} = -\frac{1}{\beta_{ir}^{k}}(\ln A_{ir}^{k} + \gamma) \qquad \forall i \in I, k \in F \backslash \{0\}, r \in R \backslash \{n\}. \tag{23}$$

4.2 EVTDA-based Mixed Integer Non-Linear formulation

Leveraging the deterministic approximation of the completion times in (23), both the sequence of jobs and configurations, as well as the expected optimal makespan, can be determined by using a non-linear model. Let us consider the following decision variables:

- y_{ir}^{k}: boolean variable equal to 1 if job i is processed at position r under configuration k, 0 otherwise;
- A_{ir}^{k}: accessibility of job i and configuration k at position r to the set of available choices at position $r+1$;
- A_0: accessibility to the set of available choices in the first position;
- T_{ir}^{k}: expected total completion time of future schedules when they are reached from job i and configuration k at position r;
- T_0: expected makespan.

Then, the following Mixed Integer Non-Linear model for the problem is proposed as a deterministic approximation of the multi-stage SP formulation of the problem:

$$\min T_0 \tag{24}$$

subject to

$$\sum_{r=1}^{n}\sum_{k=1}^{m} y_{ir}^{k} = 1 \quad \forall i \in I, \tag{25}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{m} y_{ir}^{k} = 1 \quad \forall r \in R, \tag{26}$$

$$T_{ir}^{k} = -\frac{1}{\beta_{ir}^{k}}(\ln A_{ir}^{k} + \gamma) \qquad \forall i \in I, k \in F\backslash\{0\}, r \in R\backslash\{n\}, \tag{27}$$

$$A_{ir}^{k} = \sum_{h=1}^{n}\sum_{j=1}^{m}\left(\alpha_{h,r+1}^{kj} \exp^{-\beta_{ir}^{k}(S_{r+1}^{kj}+P_{h,r+1}^{j}+T_{h,r+1}^{j})}\right)\left(1 - \sum_{\acute{r}=1}^{r}\sum_{\acute{j}=1}^{m} y_{h\acute{r}}^{\acute{j}}\right)$$
$$\forall i \in I, k \in F\backslash\{0\}, r \in R\backslash\{n\}, \tag{28}$$

$$T_{in}^{k} = 0 \quad \forall i \in I, k \in F\backslash\{0\}, \tag{29}$$

$$A_0 = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\alpha_{i1}^{0j} \exp^{-\beta_0(S_1^{0j}+P_{i1}^{j}+T_{i1}^{j})}\right), \tag{30}$$

$$T_0 = -\frac{1}{\beta_0}(\ln A_0 + \gamma), \tag{31}$$

$$y_{ir}^{k} \in \{0, 1\} \qquad \forall i \in I, r \in R, k \in F\backslash\{0\}, \tag{32}$$

$$T_{ir}^{k}, T_0, A_{ir}^{k}, A_0 \geq 0 \quad \forall i \in I, k \in F\backslash\{0\}, r \in R. \tag{33}$$

The objective function (24) expresses the minimum expected makespan. Constraints (25) and (26) ensure that each job must be selected to be processed under one configuration in exactly one position and in each position exactly one job and one configuration are assigned, respectively. Using constraints (27) and (28), the expected completion time and the accessibility of job i and configuration k at position r to the set of available choices at position $r+1$ are computed, respectively. It should be noted that, in computing A_{ir}^{k}, the part $(1 - \sum_{\acute{r}=1}^{r}\sum_{\acute{j}=1}^{m} y_{h\acute{r}}^{\acute{j}})$ indicates that the set of available choices at position r contains the jobs that have not yet been processed at any positions before r. According to constraint (29), the expected completion time of all choices at the last position is equal to zero. The accessibility to all choices in the first position and the expected makespan are calculated in (30) and (31). Clearly, in calculating A_0, the set of available choices include all the combinations of jobs and configurations, since no job has been assigned before. Finally, the binary and non-negative variables are indicated in (32) and (33).

4.3 Linearizing the EVTDA-based formulation

Despite the mixed-integer non-linear formulation presented in the previous section, being a deterministic model not obtained by scenario generation, is less computational demanding than the multi-stage formulation. However, solving it for large instances can still be challenging. Its complexity is mainly due to the non-linearity of the model and the nested structure of some of its constraints. For instance, in constraints (27) and (28), the expected completion time T_{ir}^{k} is computed using the accessibility A_{ir}^{k} that in turn is depending on the expected completion time T_{ir+1}^{k} of the chosen schedule (job and configuration) in the next position. In the following, to make possible an efficient resolution of realistic instances of the problem, a linear model inspired by the non-linear EVTDA-based formulation in (24)–(33) is developed.

As yet mentioned, the constraint (28) expresses the accessibility in the sense of Hansen. In the context of this work, such accessibility measures the attractiveness of a given schedule at the position r by computing a weighted sum of the exponential cost associated to the choices that would still be available at the position $r+1$ is case such schedule is chosen. Unfortunately the deployment of the Hansen's accessibility leads to the deterministic approximation model in (24)–(33) that is computationally challenging for realistic instances. For this reason, by maintaining the EVTDA-based model structure, we consider a different measure of accessibility that allows to obtain a new formulation of the problem that can be effectively linearized. In particular rather than taking a convex sum of the exponential cost of the alternative available at the next position as suggested by the Hansen' measure of accessibility, the new measure assesses the attractiveness of a given schedule at the position r by looking at the exponential cost of the best alternative (lowest processing plus switching time) among those still available at the position $r+1$ if such schedule is chosen. Therefore, the new accessibility measure A_{ir}^{k} which determines the attractiveness of choosing job i and configuration k at position r is computed as:

$$A_{ir}^{k} = \sum_{h=1}^{n}\sum_{j=1}^{m}\left(\exp^{-\beta_{ir}^{k}(S_{r+1}^{kj}+P_{h,r+1}^{j}+T_{h,r+1}^{j})}\right) y_{h,r+1}^{j}$$
$$\forall i \in I, k \in F\backslash\{0\}, r \in R\backslash\{n\}. \tag{34}$$

Let us suppose that the most profitable choice at position $r+1$ is to process a certain job $\bar{h}$ under a certain configuration $\bar{j}$ after processing job i under configuration k at stage r, which means $y_{\bar{h},r+1}^{\bar{j}} = 1$. Hence, the accessibility A_{ir}^{k} in equation (34) is represented as:

$$A_{ir}^{k} = \exp^{-\beta_{ir}^{k}(S_{r+1}^{k\bar{j}}+P_{\bar{h},r+1}^{\bar{j}}+T_{\bar{h},r+1}^{\bar{j}})} \qquad \forall i \in I, k \in F\backslash\{0\}, r \in R\backslash\{n\}. \tag{35}$$

So, the expected completion time T_{ir}^{k} can be reformulated as:

$$T_{ir}^{k} = -\frac{1}{\beta_{ir}^{k}}(\ln A_{ir}^{k} + \gamma)$$

$$= S_{r+1}^{k\bar{j}} + P_{\bar{h},r+1}^{\bar{j}} + T_{\bar{h},r+1}^{\bar{j}} - \frac{\gamma}{\beta_{ir}^{k}} \quad \forall i \in I, k \in F\backslash\{0\}, r \in R\backslash\{n\}. \tag{36}$$

Using, this new accessibility measure, the following simpler non-linear EVTDA-based model can be written as:

$$\min T_0 \tag{37}$$

subject to

$$T_{ir}^{k} = \sum_{h=1}^{n}\sum_{j=1}^{m}\left(S_{r+1}^{kj} + P_{h,r+1}^{j} + T_{h,r+1}^{j} - \frac{\gamma}{\beta_{ir}^{k}}\right) y_{h,r+1}^{j}$$
$$\forall i \in I, k \in F\backslash\{0\}, r \in R\backslash\{n\}, \tag{38}$$
$$T_0 = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(S_1^{0j} + P_{i1}^{j} + T_{i1}^{j} - \frac{\gamma}{\beta_0}\right) y_{i1}^{j}, \tag{39}$$
$$T_{ir}^{k}, T_0 \geq 0 \quad \forall i \in I, k \in F\backslash\{0\}, r \in R, \tag{40}$$

and constraints (25), (26), and (32).

The objective function (37) expresses the minimum expected makespan. Using constraints (38), the expected completion time of job i and configuration k at position r are computed. The expected makespan is calculated in (39).

By comparing Eqs. (17) and (36), we can notice that $-\frac{\gamma}{\beta_{ir}^{k}}$ is equivalent to the expected minimum time oscillation to be incurred if job i is chosen to be processed at position r under configuration k after switching from configuration j under which the job h has been addressed at stage $r-1$, i.e.,

$$\bar{\theta}_{ir}^{\bar{j}k} := \mathbb{E}_{\theta}\left[\min_{l \in L_{ir}^{\bar{j}k}} \theta_{ir}^{\bar{j}kl}\right] \equiv -\frac{\gamma}{\beta_{\bar{h},r-1}^{\bar{j}}}, \qquad \forall i \in I, r \in R, k \in F\backslash\{0\}. \tag{41}$$

The estimation of the expected minimum time oscillation in (41) enables us to approximate the total expected time needed to switch the machine from any configuration k to another configuration j at position r and process job i. The optimal sequence of jobs and configurations as well as the makespan can then be computed by finding a shortest path on a multi-stage network, as done in Roohnavazfar et al. (2019), through the following model:

$$\min_{x}\left[\sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{j=0}^{m}\sum_{k=1}^{m}(S_r^{jk} + P_{ir}^{k} + \bar{\theta}_{ir}^{jk})x_{ir}^{jk}\right] \tag{42}$$

subject to

$$\sum_{r=1}^{n}\sum_{j=0}^{m}\sum_{k=1}^{m} x_{ir}^{jk} = 1 \quad \forall i \in I, \tag{43}$$

$$\sum_{i=1}^{n}\sum_{j=0}^{m}\sum_{k=1}^{m} x_{ir}^{jk} = 1 \quad \forall r \in I, \tag{44}$$

$$\sum_{i=1}^{n}\sum_{j=0}^{m} x_{ir}^{jk} = \sum_{i=1}^{n}\sum_{t=1}^{m} x_{i,r+1}^{kt} \quad \forall k \in F, r \in R - \{n\}, \tag{45}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{m} x_{i1}^{0k} = 1, \tag{46}$$

$$x_{ir}^{jk} \in \{0, 1\} \quad \forall i \in I, r \in R, j, k \in F. \tag{47}$$

The objective function (42) expresses the minimum makespan. Constraints (43) ensure that each job must be selected to be processed at exactly one position under a switched configuration, while constraints (44) indicate that exactly one job and switched configurations are assigned to each position. Constraints (45) ensures the continuity of configurations switch over the positions. Constraints (46) indicate that the machine is switched from the initial configuration 0 to one of the configurations to process the job assigned to the first position. Finally, (47) are binary conditions on the variables.

5 Computational results

In this section, we present the results of the computational experiments carried out to evaluate the effectiveness of the EVTDA approach in comparison with the two-stage and multi-stage recourse models to address the proposed problem. In Sect. 5.1, we describe the instances set generated for our assessment. The calibration of the β parameters is presented in Sect. 5.3, while the value of stochastic solutions including both two-stage and multi-stage recourse models are discussed in Sects. 5.2.1 and 5.2.2, respectively. The results of our computational experiment are described and commented in Sect. 5.4.

Because of the computational complexity of the non-linear EVTDA-based model presented in Sect. 4.2, the derived equivalent shortest path model described in Sect. 4.3 is used to solve the instances. All the models are implemented in GAMS[1] on an Intel(R) Core(TM) i5-6200U (CPU2.30GHz) computer with 16 GB RAM.

[1] https://www.gams.com/. Last accessed: 2020-07-31.

5.1 Instance generation

To evaluate the performance of the proposed approximation approach, we randomly generated instances classified into small and large-scale groups. The small-scale group involves instances with 3 and 5 jobs processed on a machine with 2 configurations, while the larger ones include 10, 20, 30, and 40 jobs scheduled on a machine with 3, 4, and 5 configurations. The equivalent shortest path model is compared to the two-stage and the multi-stage stochastic models in dealing with the small and large-scale instances, respectively.

Our test generating procedure is somehow similar to the work proposed by Ertem et al. (2019). The nominal processing times are generated from the Uniform distribution in the range [1, 99]. In contrast, the sequence-dependent setup times between machine configurations are produced using the Uniform distribution in the range [51, 149]. The learning effect is assumed to be $\alpha = \log_2(0.8)$. The random oscillations are generated according to the Uniform, Normal, and Gumbel distributions in two ranges $[-0.5d, 0.5d]$ and $[-0.9d, 0.9d]$, where d is the sum of deterministic job processing time and the machine setup time associated with the random oscillation to be generated. Besides being common in many practical applications, the three above distributions have been chosen to represent quite extreme cases of possible unknown distributions of observations. Also, the two smaller and larger ranges allow us to assess the behavior of the proposed approaches against different magnitudes of random oscillations. For each problem, 10 random instances are generated, which result in 360 instances in total.

Note that, in generating realizations using the Gumbel and Normal distributions, the location parameter $\mu = 0$ is used for both small and large ranges. For the Gumbel distribution, the proportional scale parameter $\sigma = 0.125d$ and $\sigma = 0.220d$ are adopted for the smaller and larger ranges, respectively. As suggested by Manerba et al. (2018), the scale factor has been chosen experimentally so that the 96% of the probability lies in the considered truncated domains $[-0.5d, 0.5d]$ and $[-0.9d, 0.9d]$ for each possible instance. Similarly, for the Normal distribution, the standard deviation $\sigma = 0.166d$ and $\sigma = 0.3d$ is set to give a 99% confidence interval for the smaller and larger ranges, respectively.

5.2 Value of the stochastic solutions

Stochastic programs, in general, have the reputation of being computationally challenging to solve. Before solving a stochastic optimization problem, it is essential to make sure that the effort the computational effort spent on solving the whole deterministic equivalent model derived from the SP formulation is justified. In the next two subsections, the value of stochastic solutions for the generated instances is therefore determined both for the two-stage and multi-stage recourse models.

5.2.1 Value of two-stage stochastic solutions

The value of stochastic solution (VSS) is an indication of how much the decision-maker would gain if he manages to solve the whole deterministic equivalent model of the stochastic problem, instead of just ignoring the uncertainty by approximating each uncertain data with its average. This measure in two-stage recourse model is calculated as:

$$VSS = EEV - RP, \tag{48}$$

where EEV is obtained as follows: (1) solve the related expected value problem (EV); (2) fix the first stage solution for each scenario (in the recourse problem) at the optimal one obtained for the first stage of the EV problem; (3) solve the resulting problem for each scenario; (4) calculate the expectation over the set of scenarios of the value at the objective function of these modified recourse problems.

To compute the VSS, the number of observations considered in the model plays a key role. There is a trade-off between the optimality of the results and the computational complexity caused by increasing the realizations number. Here, to assess the required number of observations to solve the recourse problem in larger-sized instances, we implement a set of experiments and vary the number of realizations from 30 to 50 with the same probability. For all the instances in the larger-sized group, 50 realizations are considered. But, due to the computational complexity in the instances with 30 and 40 jobs (with 4 and 5 configurations), we fix the number of realizations to 30 for these instances. It should be noted that this number still shows well-enough results in a reasonable time. Table 1 reports the average value of the percentage VSS with respect to the RP value for the large-scale instances computed as:

$$VSS = \frac{EEV - RP}{RP} \times 100. \tag{49}$$

It can be observed, as expected, that the VSS of instances associated to the larger range of random variables is higher than those of obtained in the smaller one. Also, it can be seen that the percentage VSS is also affected by the type of probability distributions. It seems, the Uniform distribution results in larger average percentage VSS in comparison with the other two distributions.

5.2.2 Value of multi-stage stochastic solutions

In this work, the value of the multi-stage stochastic solution is calculated using the generalization of parameter VSS proposed by Escudero et al. (2007). They defined the value of stochastic solution at stage r, denoted by VSS_r, as

$$VSS_r = EEV_r - mRP \quad \forall r \in R, \tag{50}$$

where:

Table 1 Average value of the percentage VSS with respect to the RP value for large-scale instances

Instance			Uniform	Normal	Gumbel
$\|I\|$	$\|F\|$	Range	VSS_{avg}	VSS_{avg}	VSS_{avg}
10	3	Small	6.68	1.47	1.39
10	3	Large	42.64	8.08	7.86
20	4	Small	8.95	1.48	1.44
20	4	Large	52.43	10.87	10.09
30	4	Small	9.36	1.43	1.07
30	4	Large	57.71	10.95	10.20
40	5	Small	10.34	1.50	1.13
40	5	Large	60.22	11.05	10.61

– EEV_r for $r = 2, \ldots, n$, is the optimal value of the recourse problem in which the decision variables until stage $r-1$ are fixed at the optimal values obtained in the solution of the average value model, i.e.,

$$EEV_r = \begin{cases} (9)-(15) \\ s.t. \\ x_{i1}^{jk\omega} = \bar{x}_{i1}^{jk} \quad \forall \omega \in \Omega \\ \ldots \\ x_{ir-1}^{jk\omega} = \bar{x}_{ir-1}^{jk} \quad \forall \omega \in \Omega \end{cases} \tag{51}$$

where $\bar{x}_{ir}^{jk}$ denotes the optimal solution of the EV problem;
– mRP is the value of the multi-stage SP model (9)–(15).

It has been proved (Escudero et al. 2007) that the following relations for any multi-stage stochastic program hold:

$$0 \leq VSS_r \leq VSS_{r+1} \quad \forall r \in R. \tag{52}$$

This sequence of non-negative values represents the cost of ignoring uncertainty until stage r while making decision relying on a multi-stage models.

Because of the branching structure of the scenario tree, even with a small number of realizations and stages, the tree becomes huge and difficult to manage and solve. Therefore, only small-scale instances have been solved using the multi-stage recourse model. Here, we fix the number of realizations to 7 and 3 associated with the instances with 3 and 5 jobs, respectively. This leads to trees with $7^3 = 343$ and $3^5 = 243$ scenarios. These number of realizations are the values which can be handled in a reasonable computational time. Tables 2 and 3 report the average values of percentage VSS_r with respect to the mRP value for instances with 3 and 5 jobs, respectively, computed as:

$$VSS_r = \frac{EEV_r - mRP}{mRP} \times 100. \tag{53}$$

Table 2 Average values of percentage VSS_r with respect to the mRP value for instances with $|I| = 3$ jobs and $|F| = 2$ configurations

Distributions	Range	VSS_1	VSS_2	VSS_3
Uniform	Small	7.51	12.22	13.76
Uniform	Large	11.90	39.43	43.39
Normal	Small	1.34	2.24	2.70
Normal	Large	5.26	8.62	11.71
Gumbel	Small	0.00	1.49	4.19
Gumbel	Large	4.46	8.15	10.18

Table 3 Average values of percentage VSS_r with respect to the mRP value for instances with $|I| = 5$ jobs and $|F| = 2$ configurations

Distributions	Range	VSS_1	VSS_2	VSS_3	VSS_4	VSS_5
Uniform	Small	4.77	7.56	10.14	11.21	11.26
Uniform	Large	11.88	34.95	43.93	53.34	63.85
Normal	Small	1.37	2.32	2.39	3.34	3.34
Normal	Large	4.06	10.16	11.64	14.16	14.20
Gumbel	Small	0.95	2.54	3.67	3.67	3.69
Gumbel	Large	6.50	14.43	17.79	20.90	21.94

As it can be observed, the VSS average values increase over stages. It can be noticed that the average VSS per stage associated to the instances with small range of oscillations are less than those with largely affected by uncertainty. Moreover, comparing the results for the three distributions shows that the uniform distribution still yields higher average VSS.

5.3 Calibration of β parameters

The effectiveness of the proposed EVTDA-based models is highly dependent on the calibration of the β parameters.

Although EVTDA considers a single β parameter for modeling the uncertainty structure of the whole decision process, as yet mentioned, it is not straightforward to calibrate a single β parameter in such a way that it fully captures the uncertainty effect. Therefore, we propose to use many β parameters. In particular, we consider the possibility to use one β parameter to capture the uncertainty associated with the cost of each alternative in our stochastic choice process. To fully understand the process guiding the calibration of the β parameter for each alternative, it is essential to first analyze in more detail the actual role played by the single β parameter, initially considered by the EVTDA framework.

In Tadei et al. (2020) it has been shown that such a single β value is a parameter of a nested multinomial logit model (NMLM) expressing the choice probability of each alternative among those available at the next stage in a multi-stage stochastic choice process. In particular, according to such a NMLM, the choice probability of all the

Table 4 Value of parameter δ for problem sets

Distribution	Small range	Large range
Uniform	0.25	0.45
Normal	0.09	0.20
Gumbel	0.08	0.18

alternatives tends to be the same when β approaches to zero. In other words as β tends to zero, all the alternative are equivalent. On the other hand, when the magnitude of uncertainty increases, the effect of random oscillations on the deterministic cost of the different alternative becomes significant. In this case, all the alternative tend to be considered equivalent by the decision maker point of view, since it is difficult to decide which alternative is the best. The above discussion suggest that, if a single β parameter is to be considered as suggested by the EVTDA framework, it should be inversely proportional to the range of random oscillations, so that it tends to zero as the amplitudes of the oscillation increase.

In this work, as a β parameter is considered for each alternative to enhance the robustness of our solution, these parameters no longer model the diversity between the different alternatives available at the next stage and thus their choice probabilities. Instead each of them captures the variability of the future realizations of the random oscillations that determine the completion time if a certain alternative is chosen at a given stage, as suggested by the Eq. (41) that put in relation the β parameters of stage $r-1$ with the expected minimum of the random oscillations at stage r that in turn is strongly related to the expected completion time at stage $r-1$.

Based on the fact that the β parameters should tend to zero as the support of the oscillations increases, and that each β parameter should capture the dispersion of the realizations of the future random oscillations associated to its related alternative, we decided to calibrate the β parameters as follows:

$$\beta_{ir}^{k} = \frac{\gamma}{\delta \cdot STD\{\tilde{\theta}_{h,r+1}^{kjl}\}_{h\in I\setminus\{i\},\, j\in F\setminus\{0\}, l\in L_{h,r+1}^{kj}}} \qquad i \in I, r \in R, k \in F\setminus\{0\}, \quad (54)$$

where the nominator represents the Euler constant, while the denominator indicates a proportion $\delta < 1$ of the standard deviation of all possible realizations of the random oscillations in the stage $r+1$ if the machine processed job i under configuration k in stage r. We define $\delta = \frac{\sigma}{2}$ which indicates that this parameter is depending on the standard deviation of the estimated probability distribution derived from the realizations. The values of this parameter used in this paper are summarized in Table 4. As it can be seen, the δ value associated to the smaller range is less than that of the larger interval for the three distributions. Also, the δ value in the Uniform distribution is larger for both small and large ranges in comparison with the two other ones which implies the larger dispersion of this distribution. The experiments reported in Sect. 5.4 over random generated instances will show the accuracy of this calibration method.

Table 5 Average percentage gap between the two-stage and multi-stage models for the small-scale instances

Instance			Uniform	Normal	Gumbel
$\|I\|$	$\|F\|$	Range	gap_{avg}	gap_{avg}	gap_{avg}
3	2	Small	5.98	1.15	2.53
3	2	Large	21.45	6.39	5.44
5	2	Small	6.10	2.38	2.01
5	2	Large	22.99	9.18	16.72

5.4 Results and discussion

In this section, we summarize and discuss the results obtained from our experiments on the instance sets generated as discussed in Sect. 5.1. We first briefly compare the two-stage and the multi-stage SP formulations in Sect. 5.4.1, then we assess the performance of our EVTDA-based solution approach in Sect. 5.4.2.

5.4.1 Comparison of two-stage and multi-stage stochastic models

Here, we present the results of the computational experiments carried out with the aim of comparing the two-stage and multi-stage SP models. The comparison between the two-stage and multi-stage solutions is performed by computing the percentage gap calculated as

$$gap_{avg} = \frac{opt_{mRP} - opt_{RP}}{opt_{mRP}} \times 100, \tag{55}$$

where opt_{mRP} and opt_{RP} represent the optimal solutions of the multi-stage and two-stage stochastic models, respectively.

Table 5 reports the average percentage gap between the two-stage and multi-stage solutions for the instances with 3 and 5 jobs, and two configurations per job. Note that, given the computational complexity of the multi-stage model, the evaluation can be performed on just small-scale instances.

As it can be observed, the gaps are positive values for all the instances and distributions. It means, as expected, that the optimal solutions derived from the multi-stage model are better than those of obtained by the two-stage one. Also, the gaps associated to the larger support and jobs are bigger than those of smaller ones for the three distributions. This means that, as the scale of instances becomes larger, the gap between the two mentioned approaches increases. The only exception is corresponding to the average percentage gap for the instances with 5 jobs, the small support and the Gumbel distribution which is equal to 2.01. As it can be seen, this value is less than the gap associated to the instances with 3 jobs and same support and distribution. Moreover, comparing the three distributions shows the larger gaps of the Uniform distribution with respect to the Normal and Gumbel distributions.

In conclusion, the multi-stage formulation has more flexibility in dealing with the uncertainty, leading in general to possible lower costs. Note that our EVTDA-based model has been obtained by using the approximation results inside the multi-

stage structure. However, taking into account the computational time, the two-stage approach can solve the considered instances in a few seconds while the multi-stage one takes around 345 and 378 seconds to solve instances with 3 and 5 jobs, respectively (see, e.g., Table 10). Therefore, in the next section, we will compare our EVTDA-based solution approach to the multi-stage model for small-case instances, while we resort to the two-stage model for the larger ones.

5.4.2 EVTDA-based solution approach performance

The results contain two distinct parts, associated to the small and large-scale instances. The first part aims to assess the accuracy of the shortest path model derived from the EVTDA approach in comparison with the multi-stage stochastic model for small-scale problems, while in the second part, the two-stage SP formulation is used in dealing with the large-scale instances. The performance, in terms of percentage gap, is evaluated through the calculation of the Relative Percentage Error (RPE) as follows:

$$RPE = \frac{T_0 - opt_{SP}}{opt_{SP}} \times 100, \tag{56}$$

where T_0 and opt_{SP} represent the optimum makespan of the shortest path model [see Eq. (42)] and the optimum of either two-stage or multi-stage stochastic model, respectively.

We first evaluate the quality of the shortest path model with respect to the multi-stage model in dealing with the small-scale instances. Table 6 reports the results. Each entry of the table reports statistics of the RPE over 10 random generated instances, given a specific combination of size of instance and the range of random variables. In particular, the table shows the average, the best, the worst RPE, and its standard deviation (columns RPE_{avg}, RPE_{best}, RPE_{worst}, and RPE_{σ}, respectively). Those statistics are shown for the three considered probability distributions. Although we use small number of realizations in dealing with the multi-stage SP formulation, the results show promising performance of the shortest path model.

The average RPE associated to the small range of realizations is less than the one of the larger support for the considered distributions. As it can be seen, the average RPE values of the Uniform distribution are more than those of the Normal and Gumbel distributions. Taking into account the various sizes shows that the average RPE grows as the scale of instances increases for the three distributions.

Tables 7, 8, and 9 present the RPE associated to the large-scale instances generated using the Uniform, Normal, and Gumbel distributions, respectively.

The first thing that should be noticed is that, as expected, the average RPE associated with the broader range of oscillation is more significant than those of the smaller range across all instances sizes and the three distributions. It means the approximation model behaves better with smaller dispersion and magnitude of the oscillations. The best, worst, and standard deviations RPEs also follow the same described behavior with little discontinuities. Moreover, comparing the results of the three distributions highlights better performance, as expected, of the Normal and Gumbel distributions concerning the Uniform distribution in terms of the global average, best, worst, and

Table 6 RPE of the makespan between the deterministic approximation and multi-stage stochastic model in dealing with small-scale instances for the Uniform, Normal, and Gumbel distributions

Instance				$RPE(\%)$			
Distribution	$\|I\|$	$\|F\|$	Range	RPE_{avg}	RPE_{best}	RPE_{worst}	RPE_{σ}
Uniform	3	2	Small	2.12	0.49	4.76	1.25
	3	2	Large	3.61	0.06	6.67	1.79
			Avg	2.86	0.27	5.71	1.52
Uniform	5	2	Small	2.20	0.26	4.85	1.66
	5	2	Large	4.13	0.28	6.76	2.47
			Avg	3.16	0.27	5.80	2.06
Normal	3	2	Small	1.75	0.26	3.72	1.17
	3	2	Large	2.69	0.38	5.04	1.34
			Avg	2.22	0.32	4.38	1.25
Normal	5	2	Small	1.83	0.09	3.86	1.21
	5	2	Large	2.81	0.30	5.63	2.03
			Avg	2.32	0.19	4.74	1.62
Gumbel	3	2	Small	1.60	0.27	3.33	1.03
	3	2	Large	2.67	0.16	4.59	1.39
			Avg	2.13	0.21	3.96	1.21
Gumbel	5	2	Small	1.78	0.11	4.21	1.34
	5	2	Large	2.85	0.42	4.94	1.81
			Avg	2.31	0.26	4.57	1.57
Global avg				2.50	0.25	4.86	1.53

Table 7 RPE of the makespan between the deterministic approximation and two-stage stochastic model for the Uniform distribution

Instance			$RPE(\%)$			
$\|I\|$	$\|F\|$	Range	RPE_{avg}	RPE_{best}	RPE_{worst}	RPE_{σ}
10	3	Small	0.70	0.05	2.00	0.69
10	3	Large	1.19	0.27	3.90	1.09
		Avg	0.94	0.16	2.95	0.89
20	4	Small	1.48	0.50	2.88	0.67
20	4	Large	2.04	0.78	3.13	0.77
		Avg	1.76	0.64	3.00	0.72
30	4	Small	1.58	0.26	3.55	1.23
30	4	Large	2.11	0.23	5.35	1.78
		Avg	1.84	0.24	4.45	1.50
40	5	Small	1.68	0.30	3.57	0.99
40	5	Large	2.51	0.41	6.11	1.60
		Avg	2.09	0.35	4.84	1.29
Global avg			1.64	0.34	3.81	1.10

Table 8 RPE of the makespan between the deterministic approximation and two-stage stochastic model for the Normal distribution

Instance			$RPE(\%)$			
$\|I\|$	$\|F\|$	Range	RPE_{avg}	RPE_{best}	RPE_{worst}	RPE_{σ}
10	3	Small	0.42	0.04	0.88	0.27
10	3	Large	0.79	0.06	2.25	0.75
		Avg	0.60	0.05	1.56	0.51
20	4	Small	0.53	0.13	1.14	0.37
20	4	Large	1.64	0.82	2.11	0.52
		Avg	1.08	0.47	1.62	0.44
30	4	Small	0.79	0.40	1.04	0.23
30	4	Large	1.67	0.34	2.81	1.06
		Avg	1.23	0.37	1.92	0.64
40	5	Small	0.90	0.36	1.52	0.41
40	5	Large	1.93	0.18	3.97	1.09
		Avg	1.41	0.27	2.74	0.75
Global avg			1.08	0.29	1.96	0.58

Table 9 RPE of the makespan between the deterministic approximation and two-stage stochastic model for the Gumbel distribution

Instance			$RPE(\%)$			
$\|I\|$	$\|F\|$	Range	RPE_{avg}	RPE_{best}	RPE_{worst}	RPE_{σ}
10	3	Small	0.38	0.07	0.97	0.33
10	3	Large	0.64	0.04	1.51	0.43
		Avg	0.51	0.05	1.24	0.38
20	4	Small	0.42	0.12	1.18	0.42
20	4	Large	1.26	0.13	2.17	0.79
		Avg	0.84	0.12	1.67	0.60
30	4	Small	0.75	0.02	1.39	0.52
30	4	Large	1.42	0.06	2.84	0.88
		Avg	1.08	0.04	2.11	0.70
40	5	Small	0.89	0.20	1.49	0.41
40	5	Large	1.75	0.32	3.53	0.91
		Avg	1.32	0.26	2.51	0.66
Global avg			0.93	0.11	1.88	0.58

standard deviations RPEs. Finally, taking into account different sizes of instances, it can be seen that the average RPEs become more significant as the scale of instances increases for the three distributions. It means the quality of the approximation approach is also affected by the scale of the instance. The best, worst, and standard deviations RPEs also follow a similar behavior with little discontinuities.

Table 10 Computational time in seconds for calculating the minimum makespan using the deterministic approximation and stochastic models

$\|I\|$	$\|F\|$	t_{SPM}	t_{SP}
3	2	1	345
5	2	1	378
10	3	2	46
20	4	5	630
30	4	16	11,130
40	5	35	42,700

Finally, despite of the quality obtained by the deterministic approximation approach in terms of accuracy, we also want to point out its efficiency. The computational times for obtaining the minimum makespan of the shortest path model (t_{SPM}) and the stochastic problem (t_{SP}) are reported in Table 10. As we pointed out before, the small-scale instances are dealt with the multi-stage recourse model with a few number of realizations, while the larger ones are solved using the two-stage problem with the suitable number of observations. Since we have noticed that the computational times were not affected by the distribution and the range of observations, we just report average computational times for the various sizes of the instances. First, note that the CPU time increases as the size of the problem increases for both shortest path formulation and stochastic models. Comparing these two approaches, it is clear that the shortest path model derived from the deterministic approximation approach is capable of solving all instances in far less time than the stochastic models. Few seconds are needed for even the largest size (40 jobs and 5 configurations), while it takes 42700 seconds to deal with this scale using the stochastic model.

6 Conclusions

In this paper, we have addressed the stochastic single machine scheduling problem where learning effect on processing time, sequence-dependent setup times, and machine configuration selection are considered simultaneously. Also, random variables are assumed to represent uncertainty associated with job processing time and machine setup times. The problem aims at finding the sequence of jobs and choosing a configuration to process each job, which minimizes the makespan under uncertainty. First, we formulate the proposed problem as two-stage and multi-stage stochastic models that are computationally demanding. Also, an extreme value theory based deterministic approximation formulation is developed. Such a deterministic approximation can be derived even if the probability distribution of the random variables is not known. More precisely, the problem is first approximated through a mixed-integer non-linear programming model. Then, by defining a new measure of accessibility, the model above is converted into a shortest path problem on a multi-stage network, which is solvable in a few seconds, even for large-sized instances. Extensive computational experiments showed particularly good performance of our proposals with respect to the stochastic models both in terms of accuracy of solutions and computational time.

Future work will consider the use of EVTDA in different scheduling problems involving multi-stage random decision processes. From a methodological point of view, it could be interesting to develop and assess a moving-window EVTDA-based framework that iteratively considers a restricted horizon to provide optimal decisions over stages.

Acknowledgements This work has been partially supported by PIC4SeR (http://pic4ser.polito.it). The authors are also extremely grateful to Dr. Edoardo Fadda from Dept. of Control and Computer Engineering of Politecnico di Torino (Italy) for his support in the calibration method proposed for the parameters of our approximation approach.

Funding Open Access funding provided by Università degli Studi di Brescia.

References

Adamu MO, Adewumi AO (2014) A survey of single machine scheduling to minimize weighted number of tardy jobs. J Ind Manag Optim 10:219–241

Agrawala AK, Coffman JR, Garey MR, Tripathi SK (1984) A static optimization algorithm expected flow time on uniform processors. IEEE Trans Comput 33:351–357

Allahverdi A (2015) The third comprehensive survey on scheduling problems with setup times/costs. Eur J Oper Res 246:345–378

Allahverdi A, Gupta JND, Aldowaisan T (1999) A review of scheduling research involving setup considerations. Omega 27:219–239

Angel-Bello F, Alvarez A, Pacheco J, Martinez I (2011) A single machine scheduling problem with availability constraints and sequence-dependent setup costs. Appl Math Model 35:2041–2050

Azzouz A, Ennigrou M, Ben Said L (2018) Scheduling problems under learning effects: classification and cartography. Int J Prod Res 56:1642–1661

Bahalke U, Ulmeh AM, Shahanaghi K (2010) Meta-heuristics to solve single machine scheduling problem with sequence-dependent setup time and deteriorating jobs. Int J Adv Manuf Technol 50:749–759

Baker KR, Trietsch D (2009) Principles of Sequencing and Scheduling. Wiley, Hoboken

Birge JR, Louveaux F (2011) Introduction to stochastic programming, 2nd edn. Springer, New York

Biskup D (1999) Single-machine scheduling with learning considerations. Eur J Oper Res 115:173–178

Cai X, Wang L, Zhou X (2007) Single-machine scheduling to stochastically minimize maximum lateness. J Sched 10:293–301

Cheng TCE, Wu CC, Lee WC (2008) Some scheduling problems with sum-of-processing-times-based and job-position-based learning effects. Inf Sci 178:2476–2487

Cheng TCE, Wu WH, Cheng SR, Wu CC (2011) Two-agent scheduling with position-based deteriorating jobs and learning effects. Appl Math Comput 217:8804–8824

Cheng TCE, Kuo WH, Yang DL (2013) Scheduling with a position-weighted learning effect based on sum-of-logarithm-processing-times and job position. Inf Sci 221:490–500

Daniels RL, Kouvelis P (1995) Robust scheduling to hedge agains processing time uncertainity in single-stage production. Manage Sci 41:363–376

Dudek R, Smith M, Panwalkar S (1974) Use of a case study in sequencing/scheduling research. Omega 2:253–261

Ertem F, Ozcelik Tugba Sarac F (2019) Single machine scheduling problem with stochastic sequence-dependent setup times. Int J Prod Res 57:3273–3289
Escudero LF, Garin A, Merino M, Perez G (2007) The value of the stochastic solution in multistage problems. Soc Estad Invest Oper 15:48–64
Fadda E, Fotio Tiotsop L, Manerba D, Tadei R (2020) The stochastic multi-path traveling salesman problem with dependent random travel costs. Transp Sci 54(5):1372–1387. https://doi.org/10.1287/trsc.2020.0996
Galambos J (1994) Extreme value theory for applications. In: Galambos J, Lechner J, Simiu E (eds) Extreme value theory and applications. Springer, Boston. https://doi.org/10.1007/978-1-4613-3638-9_1
Gawiejnowicz SA (1996) A note on scheduling on a single processor with speed dependent on a number of executed jobs. Inf Process Lett 57:297–300
Hansen W (1959) How accessibility shapes land use. J Am Plan Assoc 25:73–76
Hu K, Zhang X, Gen M, Jo J (2015) A new model for single machine scheduling with uncertain processing time. J Intell Manuf 28:717–725
Huo JZ, Ning L, Sun L (2018) Group scheduling with general autonomous and induced learning effect. Math Prob Eng 2018:2172378. https://doi.org/10.1155/2018/2172378
Kaplanoglu V (2014) Multi-agent based approach for single machine scheduling with sequence-dependent setup times and machine maintenance. Appl Math Model 13:165–179
Kuo W-H, Yang D-L (2007) Single machine scheduling with past-sequence-dependent setup times and learning effects. Inf Process Lett 102:22–26
Lee WC, Wu CC, Sung HJ (2004) A bi-criterion single-machine scheduling problem with learning considerations. Acta Inf 40:303–315
Leksakul K, Techanitisawad A (2005) An application of the neural network energy function to machine sequencing. CMS 2:309–338
Li H (2016) Stochastic single-machine scheduling with learning effect. IEEE Trans Eng Manage 64:94–102
Lu CC, Lin SW, Ying KC (2012) Robust scheduling on a single machine to minimize total flow time. Comput Oper Res 39:1682–1691
Lu CC, Ying KC, Lin SW (2010) Robust single machine scheduling for minimizing total flow time in the presence of uncertain processing times. Comput Ind Eng 74:102–110
Maggioni F, Potra FA, Bertocchi M (2017) A scenario-based framework for supply planning under uncertainty: stochastic programming versus robust optimization approaches. Comput Manag Sci 14:5–44
Manerba D, Mansini R, Perboli G (2018) The capacitated supplier selection with total quantity discount policy and activation costs under uncertainty. Int J Prod Econ 198:119–132
Mosheiov G (2001) Scheduling problems with a learning effect. Eur J Oper Res 132:687–693
Mustu S, Eren T (2018) The single machine scheduling problem with sequence-dependent setup times and a learning effect on processing times. Appl Soft Comput 71:291–306
Perboli G, Tadei R, Baldi M (2012) The stochastic generalized bin packing problem. Discrete Appl Math 160:1291–1297
Perboli G, Tadei R, Gobbato L (2014) The multi-handler Knapsack problem under uncertainty. Eur J Oper Res 236:1000–1007
Pereira J (2016) The robust (minmax regret) single machine scheduling with interval processing times and total weighted completion time objective. Comput Oper Res 66:141–152
Pinedo ML (2012) Scheduling: theory, algorithms, and systems. Springer, New York
Ronconi DP, Powell WB (2010) Minimizing total tardiness in a stochastic single machine scheduling problem using approximate dynamic programming. J Sched 13:597–607
Roohnavazfar M, Manerba D, De Martin JC, Tadei R (2019) Optimal paths in multi-stage stochastic decision networks. Oper Res Perspect 6:100124
Seo DK, Klein CM, Jang W (2005) Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. Comput Ind Eng 48:153–161
Soroush HM (2014) Stochastic bicriteria single machine scheduling with sequence-dependent job attributes and job-dependent learning effects. Eur J Ind Eng 8:421–456
Soroush HM (2007) Minimizing the weighted number of early and tardy jobs in a stochastic single machine scheduling problem. Eur J Oper Res 181:266–287
Stecco G, Cordeau J, Moretti E (2008) A branch and cut algorithm for the production scheduling problem with sequence-dependent and time-dependent setup time. Comput Oper Res 35:2635–2655
Sun L (2009) Single-machine scheduling problems with deteriorating jobs and learning effects. Comput Ind Eng 57:843–846

Tadei R, Perboli G, Manerba D (2020) The multi-stage dynamic random decision process with unknown distribution of the random utilities. Optim Lett 14(5):1207–1218
Tadei R, Perboli G, Perfetti F (2017) The multi-path Traveling Salesman Problem with stochastic travel costs. Euro J Transp Log 6:3–23
Tadei R, Ricciardi N, Perboli G (2009) The stochastic p-median problem with unknown cost probability distribution. Oper Res Lett 37:135–141
Toksari MD, Guner E (2009) Parallel machine earliness/tardiness scheduling problem under the effects of position based learning and linear/nonlinear deterioration. Comput Oper Res 36:2394–2417
Trietsch M, Baker KR (2008) Minimizing the number of tardy jobs with stochastically-ordered processing times. J Sched 11:59–69
Yang SJ, Yang DL (2011) Single-machine scheduling simultaneous with position-based and sum-of-processing-times-based learning considerations under group technology assumption. Appl Math Model 35:2068–2074
Yang J, Yu G (2002) The robust single machine scheduling problem. J Comb Optim 6:17–33
Yelle LE (1979) The learning curve: historical review and comprehensive survey. Decis Sci 10:302–328
Yen BPC, Wan G (2003) Single machine bicriteria scheduling: a survey. Int J Ind Eng Theory Appl Pract 10:222–231
Yin Y, Xu D, Wang J (2010) Single-machine scheduling with a general sum-of-actual-processing-times-based and job-position-based learning effect. Appl Math Model 34:3623–3630
Ying KC, Bin Mokhtar M (2012) Heuristic model for dynamic single machine group scheduling in laser cutting job shop to minimize the makespan. Manuf Sci Technol 383:6236–6241
van den Akker M, Hoogeveen H (2008) Minimizing the number of late jobs in a stochastic setting using a chance constraint. J Sched 11:59–69
Zhang X, Sun L, Wang J (2013) Single machine scheduling with autonomous learning and induced learning. Comput Ind Eng 66:918–924
Zhang Y, Wu X, Zhou X (2013) Stochastic scheduling problems with general position-based learning effects and stochastic breakdowns. J Sched 16:331–336
Zhao CL, Zhang WL, Tang HY (2004) Machine scheduling problems with a learning effect. Dyn Contin, Discrete Impuls Syst, Ser A: Math Anal 11:741–750

Affiliations

Mina Roohnavazfar[1,2] · Daniele Manerba[3] · Lohic Fotio Tiotsop[1] · Seyed Hamid Reza Pasandideh[2] · Roberto Tadei[1]

Mina Roohnavazfar
mina.roohnavazfar@polito.it

Lohic Fotio Tiotsop
lohic.fotio@polito.it

Seyed Hamid Reza Pasandideh
shr_pasandideh@khu.ac.ir

Roberto Tadei
roberto.tadei@polito.it

1 Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy

2 Department of Industrial Engineering, Kharazmi University, Tehran, Iran

3 Department of Information Engineering, University of Brescia, Brescia, Italy