Annals of Operations Research
https://doi.org/10.1007/s10479-023-05592-z

ORIGINAL RESEARCH

Minimizing the expected maximum lateness for a job shop subject to stochastic machine breakdowns

Gabriel Mauricio Zambrano-Rey[1] · Eliana María González-Neira[1] · Gabriel Fernando Forero-Ortiz[1] · María José Ocampo-Monsalve[1] · Andrea Rivera-Torres[1]

Received: 28 October 2022 / Accepted: 28 August 2023
© The Author(s) 2023

Abstract
This paper addresses a stochastic job shop scheduling problem with sequence-dependent setup times, aiming to minimize the expected maximum lateness. The stochastic nature is modeled by considering uncertain times between failures (TBF) and uncertain times to repair (TTR). To tackle this problem, a simheuristic approach is proposed, which combines a tabu search (TS) algorithm with Monte Carlo simulation. A total of 320 instances were used to conduct multiple experiments. Instances were generated with two distributions to study the behavior of stochastic TTR and TBF under log-normal and exponential distributions. Firstly, the performance of the simheuristic was evaluated for small instances by comparing it with the simulation of optimal solutions obtained with a mixed-integer linear programming (MILP) model. The simheuristic approach demonstrated an average improvement of around 7% compared to the simulation of MILP model solutions. Secondly, the simheuristic performance was evaluated for medium and large-size instances by comparing it with the simulation of the solutions obtained by the earliest due date (EDD) and process time plus work in the next queue plus negative slack (PT + WINQ + SL) dispatching rules. The results showed an average improvement of around 11% compared to EDD and 14% compared to PT + WINQ + SL. Furthermore, the results highlight that even when the two distributions have the same expected value and coefficient of variation, they can yield different expected maximum lateness values. This emphasizes the importance of precise distribution fitting when solving real cases to achieve effective scheduling performance.

Keywords Stochastic job shop scheduling · Maximum lateness · Machine breakdowns · Simheuristics · Tabu search

✉ Eliana María González-Neira
eliana.gonzalez@javeriana.edu.co

[1] Departamento de Ingeniería Industrial, Pontificia Universidad Javeriana, Carrera 7 # 40-62, Bogotá, Colombia

1 Introduction

In today's dynamic and competitive world, the pursuit of excellence is a necessity for surviving in markets. Globalization and the technological revolution of the last century have compelled companies to improve all their internal processes. Under this premise, new methodologies and paradigms have emerged to address challenges such as reducing process times, waste, labor or production and logistics costs, among others (Hill, 2011). Additionally, new forms of customer demand and marketing are driving companies towards mass customization of products and services based on specific customer needs. As a result, companies are forced to continuously optimize their production planning and execution to remain competitive.

Production scheduling consists of allocating resources to operations to achieve one or more objectives (Pinedo, 2016). The study of scheduling in manufacturing environments began in the early twentieth century with the works of Gantt (1919) and other pioneers such as Smith (1956), Johnson (1954) and Jackson (1956). One way to classify production scheduling problems is based on the configuration of machines in the production environment. The generic categories include single machine, parallel machines, flow shop, job shop, and open shop. Specifically, the job shop scheduling problem (JSSP) refers to scheduling jobs on a set of m machines, where each job has a single predetermined path, which is independent of the paths of other jobs, and each machine can only process one job at a time (Çaliş & Bulkan, 2015; Jain & Meeran, 1999). This type of configuration occurs in industries such as aerospace (Chiang & Torng, 2016), machine tools production systems (Kadipasaoglu et al., 1999), construction equipment production systems (Jacobs & Lauer, 1994), among others.

The literature review presented by Abdolrazzagh and Adbullah (2017) summarizes the characteristics studied in the JSSP, as well as the objective functions considered. In particular, the authors mention that the classic constraints considered are related to precedence, capacity, and the inclusion of release times and due dates. However, there are other constraints that have been studied in fewer works due to the complexity they add to the problem. These constraints include multi-purpose machines, deteriorating jobs, maintenance activities, sequence-dependent setup times, availability constraints due to machine breakdowns, overlapping operations, controllable processing times, among others (Abdolrazzagh & Adbullah, 2017).

Additionally, Abdolrazzagh and Adbullah (2017) also reported that the most commonly studied objective functions in the JSSP are the makespan, weighted tardiness, and maximum tardiness. Based on the review of recent literature conducted for this project, it was found that more than 80% of the studies on the JSSP focus on minimizing the makespan. The remaining less than 20% of studies are divided among the minimization of weighted tardiness, process delays, and events associated with the machines.

On the other hand, most studies conducted in JSSP have considered all parameters as deterministic (Mohan et al., 2019). However, industrial environments are inherently complex with the occurrence of events with stochastic behavior such as machine breakdowns, maintenance time, processing times, among others (Meng et al., 2020). Most papers that address the JSSP, assume that machines are always available, although machine breakdowns are one of the most common and challenging issues in production environments (Zhang et al., 2022). If machine failures are not considered during scheduling development, such schedules can become inapplicable (Shady et al., 2021). Particularly, stochastic times between failures and repairs happen for various reasons. Whenever a failure occurs, the maintenance crew takes time to assess the problem, depending also on the crew availability and expertise on the technology. Then, based on the severity of the problem, repair times can vary, whether

immediate on-site repairs can be done (e.g. oil change, part replacement, parameter setting, among others), or the machine need be transferred to the maintenance workshop, or even replaced (Giacaman et al., 2002). Despite the benefits of including stochastic malfunctions within the scheduling process, the problem becomes much more complex due to the large search space (Shady et al., 2021).

One of the booming methods used in the literature to solve stochastic optimization problems are simheuristics, which involves the hybridization of metaheuristics and simulation techniques (Juan et al., 2015). The advantage of implementing simheuristics is that they allow for the verification of solutions through a simulation process that takes into account the previously defined stochastic variables. As a result, the obtained solutions are more suitable for the problem context since they consider uncertainty. Simheuristics have been successfully applied to various problem domains, including routing problems (González-Martín et al., 2016; Gruler et al., 2018; Quintero-Araujo et al., 2019), facility location problems (de Armas et al., 2017), production scheduling problems (González-Neira & Montoya-Torres, 2019; Juan et al., 2014), among others.

Considering that simheuristics are a hybridization between metaheuristics and simulation, it is important to mention that several metaheuristic approaches have been used to solve JSSP in the literature. These approaches include alternative graph (Meloni et al., 2004), simulated annealing (SA) (Baykasoglu, 2002; Naderi et al., 2010), tabu search (TS) (Nowicki & Smutnicki, 2005; C. Zhang et al., 2007), iterated greedy algorithm (Pranzo & Pacciarelli, 2016), hybridization of immune artificial and SA algorithms (Roshan et al., 2013), hybridization of evolutionary algorithm and TS (Cheng et al., 2016) and, hybridization genetic algorithm (GA) and SA (Shahsavari-Pour & Ghasemishabankareh, 2013).

In this paper, a TS algorithm is proposed to solve the stochastic job shop scheduling problem (SJSSP). One of the advantages of TS according to Abido (2002) is the robustness of the parameters and the initial solution. In addition, TS is characterized by its ability to avoid entrapment in local optimum solutions and to prevent cycles due to the use of flexible memory for historical search. In fact, recent works in the field of JSSP have demonstrated the effectiveness of TS, including studies by Aschauer et al. (2020a); Tamssaouet et al. (2018); and Xie et al. (2023).

Consequently, this paper reports a simheuristic approach to solve SJSSP, hybridizing a TS with Monte Carlo simulation (MCS), to minimize the expected maximum lateness, considering stochastic times between failures (TBF) and stochastic times to repair (TTR) with non-resumable jobs. The maximum lateness is related to the delivery date to customers, which implies minimizing the worst violation of due dates (Pinedo, 2016) and it is recommended for make-to-order environments (Botta-Genoulaz, 2000). Furthermore, the stochastic TBF and TTR are modeled with two probability distributions: log-normal and exponential. The log-normal distribution is important since it does not take negative values which it is appropriate for reproducing machine malfunction times. In turn, the exponential distribution is often used for breakdowns because there is a high degree of uncertainty about failures; thus, the only information available on them is their mean.

The remainder of the paper is organized as follows: Sect. 2 presents the literature review. In Sect. 3, a mixed integer linear programming formulation is proposed for the deterministic version of the problem. Section 4 explains the simheuristic approach, while Sect. 5 describes the computational experiments carried out to evaluate the performance of the proposed approach. Finally, Sect. 6 provides conclusions and future work.

2 Literature review

Firstly, this section reports the most recent studies on the deterministic version of the JSSP. The second subsection focuses on articles that address SJSSP. Lastly, the third subsection discuses simheuristic approaches to tackle both the JSSP and the SJSSP.

2.1 Deterministic JSSP

The interest in theoretical models for the resolution of the deterministic JSSP began in the middle of the last century (Abdolrazzagh & Adbullah, 2017). Two of the first studies in JSSP were performed by Akers and Friedman (1955) who used Boolean Algebra, and Jackson (1956) that generalizes Johnson's Flow Shop algorithm to JSSPs to minimize makespan. From those researches until today, literature in JSSP has grown considerably using innovative techniques. Methods to solve the JSSP vary from simple dispatching rules to sophisticated algorithms that hybridize metaheuristics to obtain more accurate results. Figure 1 shows the different techniques that have been implemented in recent years.

In the field of optimization, exact methods have been employed to solve classical JSSPs (Zhang et al., 2019). These methods rely on input data and a series of predefined rules to determine the flow of the process and obtain an optimal global solution. However, the runtime of these methods is unpredictable, and for larger problems, they would require significant computational time (Gmys et al., 2020). One of the most commonly used exact methods in JSSP is the Branch & Bound algorithm (Gmys et al., 2020). Additionally, there are several mathematical models that are widely used, including mixed-integer linear programming (Manne, 1960), Langrangian relaxation (Fisher, 1973), and integer linear programming (Li et al., 2018).

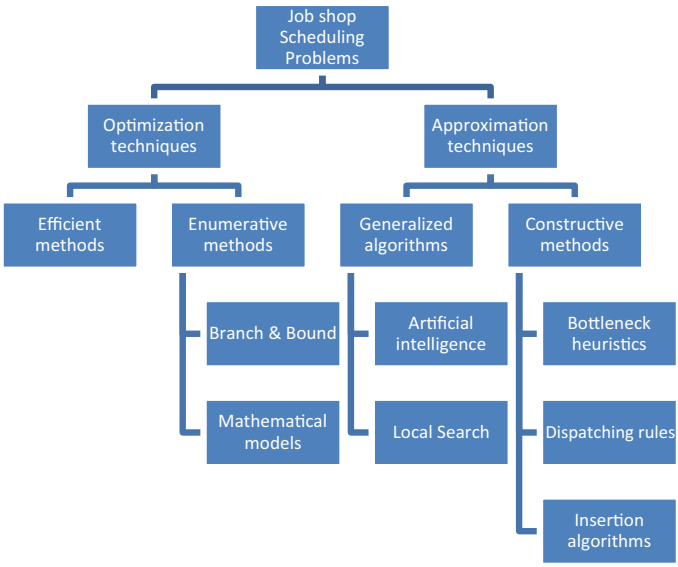


Fig. 1 Techniques for job-shop scheduling

Since exact methods require high computational times for medium to large instances, approximation methods have been developed to find solutions proximate to global optimum in less computational time (Fernandez-Viagas et al., 2017). In addition, optimal solutions for small instances can be used as a standard reference to develop approximate methods (Li et al., 2018). Among the approximation methods one can find the constructive methods such as dispatching rules, bottleneck heuristics and insertion algorithms (Jain & Meeran, 1999). There are also iterative methods, which use different phases to approach an optimal solution, selecting it either from a random list or with a probability function. For instance, greedy algorithms such as GRASP (Bierwirth & Kuhpfahl, 2017; Feo & Resende, 1995) or biased-randomization algorithms.

Generally speaking, JSSP approaches mostly focus on minimizing makespan and tardiness. The makespan is defined as the time taken to complete all jobs. On the other hand, tardiness is related to due dates and measures the delay of jobs in relation to their due date (Pinedo, 2016). Regarding the constraints considered, in the most recent works reviewed for this paper, characteristics such as maintenance, breakdowns, and slack time represent only around 10% of the studies.

2.2 Stochastic JSSPs

Most decision-making problems are combinatorial optimization problems that can be modeled with uncertain conditions (Juan et al., 2015). Therefore, stochastic approximation methods are used to optimize or control systems where the relationship between variables and controllable factors of a process and its analytical model are unknown. These problems are known as stochastic combinatorial optimization problems. Banerjee (1965) and Makino (1965) were the first authors to conduct studies in production scheduling with stochastic processing times. Banerjee solved a single-machine problem in which processing times follow a known probability distribution, with the aim of minimizing the maximum probability of delay. On the other hand, Makino carried out the study with 2 and 3 machines in a flow shop environment considering exponential and k-Erlang distributions for processing times, minimizing the makespan (Juan et al., 2015). Some works assume that the starting times and duration of disturbances (for instance, machine failures) are known in advance and fixed (Yahyaoui et al., 2009), but in other works, probability distributions are used to generate such variables.

A search on Scopus using the following formula TITLE-ABS-KEY(((“Stochastic Job Shop Scheduling”) OR (“job shop scheduling”)) AND (“Machine Breakdowns”)) gave 134 results, but less than 90 papers were related to the JSSP with stochastic machines breakdowns. The first paper was reported back in 1982 by Muhlemann et al. (1982). The authors included dynamic job arrivals, uncertainty in job processing times, and the likelihood of machine breakdowns. Various heuristics in the form of ***dispatching rules*** were applied to determine the impact of variability in the mean ratio of flow time to process time, mean queueing time, mean lateness and percentage of tardy jobs. Later on, Tunali and Orhun (1990) develop a knowledge-based system for production scheduling in a job-shop using around 300 dispatching rules with simulation-based performance evaluation. The authors studied the impact of unexpected machine breakdowns on the predefined schedule, modeling the time to failure with a Poisson distribution. Output analysis was focused on machine utilization (under or over) and job lateness. Likewise, Paredis and Van Rij (1992) used a simulator and an evaluator to understand the effects of the decisions taken during the scheduling process. The simulator tests the robustness of the schedules, provided by a Genetic Algorithm, with respect to

machine failures. Similarly, Holthaus (1999) also reported a simulation-based analysis of dispatching rules for scheduling in a dynamic job shop, with respect to flow times and due dates. The study showed that, in terms of mean flow time, the PT + WINQ (processing time plus work in the queue of its next operation) outperforms other rules; but when the interest is focused on minimizing the maximum flow time, the AT-RPT (arrival time minus remaining processing time) becomes superior. With respect to tardiness-based objectives, scheduling rules showed high sensitivity to the percentage of machine downtimes and mean time to repair. Experiments were carried out using an exponential distribution to represent TTR and TBF for all machines.

As reported by Fang and Xi (1997), a periodic and event-driven rolling-horizon strategy can help accommodate critical situations in job-shop environments caused by continuous changes in job arrivals, machine breakdowns, machine repair times and job due dates. In this work, the authors hybridized a GA with dispatching rules, while also considering sequence-dependent setup times. All events are simulated at specific points in time, and the impact on makespan is analyzed. In the presence of uncertainty, a robust or flexible solution is more valuable than an optimal solution. For instance, Tjornfelt-Jensen and Hansen (1999) proposed a robust performance measure to identify robust solutions provided by a GA. The robustness measure is defined as the weighted average of makespan of schedules within a neighborhood. To simulate machine failures, an operation is randomly chosen using a uniform distribution. The starting time of this operation is called the breakdown time. The initial setup time for the machine in failure is set to the time the broken machine becomes operational again, which is the breakdown time plus the breakdown duration. Five different rescheduling methods were investigated. The computational experiments revealed that, in most cases, the rescheduling performance of schedules with a low robustness measure is significantly better than the performance of ordinary schedules for all tested strategies. The proposed measure yielded good results for moderate breakdowns when compared to the slack-based measure. Likewise, based on a rolling-horizon strategy, Upasani and Uzsoy (2008) proposed a problem-reduction scheme to obtain global schedules with an integrated procedure that provides priority indices for a dispatching approach in a multi-product environment with stochastic machine breakdowns and job arrivals. The method was used to schedule operations of a wafer fabrication facility where the presence of two unreliable stations can cause starvation in the system. TBF and TTR were generated using gamma distributions.

Zou and Li (2006) used a GA and a hybrid allocation-based method for event-oriented scheduling in JSSP under rush orders and machine breakdowns. If a rush order arrives, the schedule is put on hold until the rush order is processed, but in the case of a machine breakdown, only the affected operations are rescheduled. As pointed out by Suwa and Sandoh (2007), the timing and frequency of rescheduling are important aspects to study. Therefore, the authors used a tabu search (TS) to define a predictive schedule. Based on a control limit policy, the time to conduct a schedule revision is determined based on the cumulative size of delays. Sixteen machine breakdown scenarios were analyzed using simulation, and the right-shift scheduling (RSS) policy was employed when a breakdown occurred. In terms of minimizing maximum tardiness and scheduling frequency, this approach outperformed other event-driven rescheduling approaches. Also using GA, Al-Hinai and Elmekkawy (2011) presented a robust two-stage hybrid genetic algorithm (HGA). In the first stage, the algorithm optimized the makespan using deterministic data. Then, in the second stage, it focused on a bi-objective function by integrating machine assignments and operation sequencing with the expected machine breakdown in the decoding space. While in most works, breakdown variables are assumed to be random, the authors proposed that a machine with a higher load is more likely to fail, and its breakdown time and duration can be assumed to be proportional

to its elapsed busy time. The timing of the breakdown can also be set early or late in the schedule.

Another example of a hybrid GA is reported by Hasan et al. (2011), in which a GA is hybridized with two simple priority rules, the partial re-ordering (PR) and the gap reduction rule (GR), as well as the gap reduction and restricted swapping rule (GR-RS). The approach was first tested using 40 benchmarks for the deterministic version. For the stochastic, a uniform distribution was used to identify the machine with a failure, a Poisson distribution generated the start time of the breakdown, and an Exponential distribution was used for recovery duration. First, the solution for the ideal JSSP is generated, and then the GR algorithm is applied to re-optimize the affected sub-solution when a breakdown occurs. Similar to other studies, the authors concluded that the impact of machine unavailability is quite low if the information is known in advance. However, reactive scheduling becomes quite complex when the breakdown is sudden. In the study conducted by Lei (2011), the SJSSP is addressed using a simplified multi-objective GA, with exponentially distributed processing times, aiming to minimize both makespan and total tardiness. The author recommended incorporating of stochastic machine breakdowns into the model. In a subsequent work by Lei (2012), the same JSSP was examined, but with stochastic normal processing times, resumable jobs, and stochastic machine breakdowns, with the objective of minimizing makespan. To assess the effectiveness of the proposed approach, it was compared to particle swarm optimization (PSO), and the results demonstrated that GA outperforms PSO.

Hybrid metahaheuristic approaches have also been proposed to improve the pedictive-reactive scheduling strategy. For instance, X. Li and Gao (2020) combined the advantage of a GA and TS to deal with the SJSSP, coupled with event-driven rescheduling policies. The approach was set up to minimize mean flow time, mean tardiness, maximum tardiness, and the number of tardy jobs. The TBF and TTR are assumed to follow an exponential distribution. To generate unexpected events, a simulator was designed not only to reproduce machine breakdowns but also uncertain job arrivals.

Chen et al., (1997) proposed a solution methodology that combines ***Lagrangian relaxation, stochastic dynamic programming, and ordinal optimization*** to address the JSSP considering uncertain machine failures and job processing times. The machines are categorized into reliable and unreliable sets, where the latter have known conditional failure and recovery rates based on maintenance records. As a result, the number of available machines becomes a random variable, and the algorithm's performance is analyzed using Monte Carlo simulation (MCS). Similarly employing Lagrangian relaxation, Zhu and Soh (1999) introduced a two-stage procedure for off-line scheduling and an affected operations rescheduling strategy (AOR) to generate new solutions in the presence of machine breakdowns, changes in job priority, rush orders, order cancellations, and delays in raw material arrivals. The objective is to minimize the mean weighted flow time for solving FMS job shop problems. Events are simulated at specific moments, and the authors concluded that among the five types of interruptions, machine breakdowns are the most challenging because their impact depends on the timing, duration, and number of available machines. However, this conclusion assumes that the breakdown time is known when it occurs. Subramaniam and Raheja (2003) proposed a modified AOR strategy that depends on the type of disruption. The authors discussed strategies for machine breakdowns, process time variations, arrival of unexpected jobs, and urgent jobs. For machine failures, an idle time equivalent to the breakdown duration is inserted into the schedule, and the operation on the machine is put on hold until the machine is repaired. Then, the schedule is repaired using the AOR heuristic. A simulation study was conducted to examine the impact of disruptions in terms of size, occurrence time, and schedule size. For machine breakdowns, the disruption size was analyzed for both small

(1–3% of the makespan) and large magnitudes (6–8% of the makespan). The experimental results demonstrated that the AOR outperforms the RSS heuristic significantly in terms of efficiency and stability.

Within the robust strategy to cope with uncertain machine failures, the buffering approach can be applied to predictive schedules to reduce performance deterioration and improve production system stability. Wu et al. (2020) proposed a multi-objective predictive scheduling algorithm that considers three objectives: makespan, makespan risk and stability risk simultaneously. Two types of operation-block-based buffering strategies were implemented, and after extensive experiments, the proposed approach was found to provide a better Pareto solution set in terms of both diversity and convergence.

Fuzzy systems have been applied widely, mostly on control problems. Bilkay et al. (2004) proposed a two-stage fuzzy logic-based algorithm. In the first stage, the algorithm assigns priorities to part types, and in the second stage, an operation allocation is performed to minimize makespan, relative machine loads and cost. The proposed algorithm regenerates the schedule after a machine breakdown, making it suitable for use as an on-line controller. In the same vein, proactive scheduling aims to constructs schedules that take into account statistical knowledge of uncertainty upfront. Bollapragada and Sadeh (2004) reported proactive technique that determines the release dates of different jobs based on measures of shop load, statistical data of machine failures, and repairs, with a tradeoff between inventory and tardiness costs. Using a backward scheduling procedure, initial release dates are determined. Then, a forward scheduling procedure is used to make sequencing decisions. Experiments were carried out assuming normally distributed TTR and TBF. The approach was compared with 39 other rules, and SA proved to yield better schedules even with an increasing frequency of breakdowns. Petrovic et al. (2008) proposed a fuzzy approach to determine the size of lots, considering fuzzy processing times and fuzzy due dates, to minimize the multiple objectives such as weighted tardiness of jobs, total setup times, total tardy jobs, total flow time, and total machine idle times. To evaluate the efficiency of the solution method, data from a printing company were used.

Neural networks have also been used, where neurons can represent processing restrictions and resolve constraint conflicts. For example, Y. Li and Chen (2009) combined the backpropagation neural network (BPNN) to model machine breakdowns and new job arrivals by detecting whether constraints are satisfied. Then, based on the BPNN's feasible solution for the JSSP, a GA is applied for sequence optimization and makespan minimization. Likewise, the works of Zandieh and Adibi (2009) and Adibi et al. (2010) also configured an artificial neural network with a backpropagation error learning algorithm, but in this work, the neural network was employed to update parameters of a variable neighborhood search (VNS) algorithm in response to random job arrivals and machine breakdowns. Similarly, Azadeh et al. (2012) proposed a hybrid algorithm that evaluates its performance using a discrete-event simulation model and artificial neural networks (ANN) to solve the JSSP. The objective of the proposed algorithm is to select the best dispatching rule to assign to each machine with the aim of minimizing the makespan. The authors considered random processing times measured in three probability distributions: normal, exponential, and uniform. The standard deviation and mean variability were kept consistent for each machine.

Multi-Agent Systems (MAS) are considered decentralized models in which agents represent entities capable of making decisions in real-time. Liu et al. (2004) explored the use of multiple agents employing an auction-based mechanism to schedule job shops in a dynamic environment, utilizing a rolling horizon time procedure. Only available machines can host auctions; therefore, if a machine breaks down, any job can be allocated to it. Similarly, based

on decentralization, M. Cheng et al. (2005) proposed a method based on the reaction–diffusion equation on a graph, which treats the operations of JSSP as autonomous decentralized system elements. When an event-rescheduling is triggered, the machine disturbance is considered as a constraint and is modeled as local and global potentials. However, message-based protocols can impose communication burdens, so Renna (2009) included pheromone-based information exchange strategies for a multi-agent system operating in a dynamic JSSP with inter-arrival time, machine breakdowns and processing times variability.

Holonic Manufacturing Systems (HMS), as well as MAS, are distributed paradigms for production planning and control. Similar to agents, holons in HMS are autonomous, cooperative, and intelligent entities that collaborate with others to complete the tasks. For instance, Zhao et al. (2010) utilized a fuzzy inference system (FIS) for machine selection and allocation. The machine reliability is determined by the FIS based on its mean time between failures (MTBF) values. Higher priority is given to the most reliable machines, and to prevent unbalanced machine workloads, a hybrid PSO with differential evolution (DE) is applied. The MTBF values are represented by a triangular membership function, ranging from 0 and 50.

Recently, ***Deep Learning (DL)*** techniques have been applied to solve the dynamic JSSP due to their capability to extract features to solve the problem, while the context changes. S. Chen et al. (2022) proposed a spatial pyramid pooling-based transformer (SPP-Transformer) that extracts environment features in order to select the best priority rule. The authors claim that this approach enables fast rescheduling by leveraging the simplicity of priority rules and the ability of DL methods to extract context features. They reported an improvement of 11% in the makespan.

One way to test new solutions in the presence of perturbations is through simulation. For instance, Feng et al. (2020) designed a capacity adjustment system for a multi-product job shop based on a heuristic decision-making module. The simulation module is utilized to assess the performance of the solutions when machine alterations occur. The objective is to monitor the system and propose corrective actions to minimize the delay rate, maximize the machine utilization rate, minimize the number of used machines, and minimize the mean total waiting time. The machine parameters follow an exponential distribution, and the results showed that the system can enhance the agility of the production system to cope with the complex and ever-changing industrial setting. In the following subsection, approaches coupling simulation with the concept of simheuristics are described.

2.3 Simheuristic approaches to solve SJSSP

Few works using simheuristics have been proposed to deal with uncertain parameters in JSSP. For instance, the following approaches use MCS to account for stochastic problem parameters. Particularly, Beck and Wilson (2007) proposed four constructive algorithms and two TS algorithms hybridized with MCS to solve a SJSSP with uncertain processing time, aiming to obtain the solution with a high probability of a good makespan. The algorithms were tested for instances up to 20 jobs with 20 machines. The results showed that for smaller instances, the constructive algorithms performed better, while for larger instances, the TS algorithms obtained the best results. Similarly, Horng et al. (2012) proposed an evolutionary algorithm by embedding evolutionary strategy in ordinal optimization to minimize expected sum of inventory costs and tardiness costs in a SJSSP. The approach was evaluated in comparison to five dispatching rules demonstrating an excellent performance. Araki and Yoshitomi (2016) minimized expected makespan in a SJSSP with stochastic processing times. A pseudo PSO

was combined with MCS to solve the problem. The performance of the algorithm was evaluated by comparing it to a GA that was also hybridized with MCS. The results demonstrated the effectiveness of the proposed simheuristic in achieving good performance. Using the same strategy, Hao et al. (2017) tackled a multi-objective SJSSP with stochastic processing times. The objectives were expected makespan and expected tardiness. A multi-objective estimation of distribution algorithm was hybridized with MCS to solve the problem. The simheuristic was compared to other simheuristics, namely NSGA-II, strength Pareto evolutionary algorithm and adaptive-weight GA, demonstrating a good trade-off between quality and computational efficiency. Aschauer et al. (2020b) solved a no-wait SJSSP with controllable processing times to minimize makespan. Firstly, the authors proposed a decomposition approach that divided the problem into timetabling and sequencing problems. Then, using MCS, the approach was evaluated for the stochastic version of the problem in benchmark instances, resulting in a 40% reduction in the standard deviation of the makespan.

Due to the complexity of population metaheuristics and the requirement for multiple replications to estimate stochastic parameters, R. Zhang and Wu (2011) employed the K-armed bandit model as a time-saving strategy. In this study, an artificial bee colony algorithm was integrated with the budget allocation method. MCS was applied to approximate the minimum expected maximum lateness, considering stochastic processing times. The algorithm was compared to a PSO hybridized with SA, demonstrating better results when integrating optimal computing budget allocation (OCBA). The approach was tested on several instances, validating the effectiveness of the algorithm. Similarly, by employing the OCBA, Yang et al. (2014) addressed the minimization of the expected sum of earliness and tardiness in a SJSSP with stochastic processing times. The authors utilized an Evolutionary Strategy in Ordinal Optimization (ESOO) hybridized with OCBA to solve the problem. The ESOO technique is based on MCS. The method was tested by modeling the processing times using truncated normal, uniform, and exponential distributions. The results were compared with those obtained by ESOO alone, demonstrating that the usage of OCBA improved the results.

Lastly, other works combined optimization methods with discrete-event simulation. For instance, Sel and Hamzadayı (2018) considered a dynamic JSSP, in which jobs arrive continuously, with the objective of minimizing the mean flowtime and mean tardiness. They proposed a simulation–optimization method that utilizes a SA algorithm and includes the analysis of machine failures. The metaheuristic calls a simulation model implemented in Arena to estimate the objective function for a set of solutions. The method was compared with simulations using three dispatching rules: earliest due date (EDD), shortest process time (SPT), and first in first out (FIFO). The results showed that SPT achieved the best performance in terms of objectives, while the SA algorithm provided a trade-off between objectives and computational times.

Based on this literature review, the simheuristic-based approach described in this paper contributes in the following ways. Firstly, no reports have been found on simheuristic approaches for addressing the SJSSP with sequence-dependent setup times, to the best of the authors' knowledge. Secondly, just-in-time non-linear objective functions, such as minimizing the maximum lateness, have also been scarcely addressed in literature. The closest referenced found was the work of Zhang and Wu (2011), which addressed the minimization of expected maximum lateness was addressed but assumed stochastic processing times, without considering stochastic machine breakdowns or sequence-dependent setup times as in the proposed approach. Thirdly, since the problem has not been previously tackled, this paper proposed a set of instances, modified from the related literature, that can serve as a reference for future research and method comparisons. Lastly, realistic applications necessitate comprehensive data collection and pre-processing, highlighting the significance of selecting

appropriate probabilistic distributions for representing the data. To address this, the paper proposes a sensitivity analysis of problem representation (e.g., machine failures) and method parameterization.

3 Mathematical model of the deterministic JSSP

This section presents the mathematical model for the deterministic JSSP that minimizes maximum lateness, taking into account deterministic breakdowns. The model incorporates subsets of breakdowns, denoted as MB_i for each machine i. The quantity of breakdowns in subset MB_i is calculated based on three factors:

(i) The planning horizon that represents the maximum length that can take to process all the jobs. It is calculated as the sum of processing times of all jobs in all machines plus the sum of maximum values of setup for each job at each machine.
(ii) The mean time to repair (MTTR) that represent the expected duration of a breakdown repair.
(iii) The mean time between failures (MTBF) that represents the expected duration between two consecutive breakdowns.

These three factors are used to calculate the starting and finishing times of a breakdown ($SB_{b,i}$ and $FB_{b,i}$), which are treated as parameters in the model. The MTTR and MTBF are further explained in Sect 4.3. First, the notations used in the model are presented, including sets, parameters, and decision variables. Then, the objective functions and constraint sets are presented and explained.

Sets:
I: Machines $\{1, \ldots, |I|\}$
J: Jobs $\{1, \ldots, |J|\}$
O_j: operations of job $j \in J$ $\left\{1, \ldots, \left|O_j\right|\right\}$
MB_i: machine breakdowns of machine $i \in I$ $\{1, \ldots, |MB_i|\}$

Parameters:

- $P_{o,j}$: processing time of operation $o \in O_j$ of job $j \in J$
- D_j: due date of job $j \in J$
- $R_{o,j}$: machine index in which operation $o \in O_j$ of job $j \in J$ has to be performed
- $S_{j,k,i}$: sequence dependent setup time if job $j \in J \cup \{0\}$ is processed before job $k \in J$ at machine $i \in I$. Before means without the insertion of other jobs.
- $SB_{b,i}$: starting time of breakdown $b \in MB_i$ at machine $i \in I$
- $FB_{b,i}$: finishing time of breakdown $b \in MB_i$ at machine $i \in I$
- M: a very large positive number

Variables:

- $X_{j,k,i}$: binary variable that takes the value of 1 if job $j \in J \cup \{0\}$ is processed before job $k \in J$ at machine $i \in I$, $j \neq k$. 0 otherwise. Note: 0 is a dummy job to detect the job to be processed in the first position of sequence at machine $i \in I$. Before means without the insertion of other jobs.
- $C_{i,j}$: completion time of job $j \in J$ at machine $i \in I$
- $B_{i,j}$: starting time of setup at machine $i \in I$ before processing of job $j \in J$ at that machine. Before means without the insertion of other jobs.

- L_j: Lateness of job $j \in J$
- L_{max}: Maximum lateness
- $W_{j,i,b}$: binary variable that takes the value of 1 if job $j \in J$ is processed after the breakdown $b \in MB_i$ takes place, 0 if it is processed before the breakdown $b \in MB_i$ takes place.

Objective function:

$$\text{MinZ} = L_{max} \tag{1}$$

Subject to:

$$L_{max} \geq L_j \quad \forall j \in J \tag{2}$$

$$L_j = C_{i,j} - D_j \quad \forall j \in J, \forall i \in I \tag{3}$$

$$\sum_{j \in J \cup \{0\}, k \neq j} X_{j,k,i} = 1 \quad \forall i \in I, \quad \forall k \in J \tag{4}$$

$$\sum_{k \in J, k \neq j} X_{j,k,i} \leq 1 \quad \forall i \in I, \quad \forall j \in J \tag{5}$$

$$\sum_{j \in J} X_{0,j,i} = 1 \quad \forall i \in I \tag{6}$$

$$B_{(R_{o,j}),j} \geq C_{(R_{o-1,j}),j} \quad \forall j \in J, \quad \forall o \in O_j, o > 1 \tag{7}$$

$$C_{(R_{o,j}),j} = B_{(R_{o,j}),j} + P_{o,j} + \sum\nolimits_{k \in J \cup \{0\}} S_{k,j,(R_{o,j})} \cdot X_{k,j,(R_{o,j})} \quad \forall j \in J, \quad \forall o \in O_j \tag{8}$$

$$B_{i,k} \geq C_{i,j} - M \cdot \left(1 - X_{j,k,i}\right) \quad \forall i \in I, \forall j \in J, \quad \forall k \in J, j \neq k \tag{9}$$

$$SB_{b,i} \leq C_{i,j} + M \cdot \left(1 - W_{j,i,b}\right) \quad \forall j \in J, \quad \forall i \in I, \quad \forall b \in MB_i \tag{10}$$

$$SB_{b,i} \leq B_{i,j} + M \cdot \left(1 - W_{j,i,b}\right) \quad \forall j \in J, \quad \forall i \in I, \quad \forall b \in MB_i \tag{11}$$

$$FB_{b,i} \geq C_{i,j} - M \cdot W_{j,i,b} \quad \forall j \in J, \quad \forall i \in I, \quad \forall b \in MB_i \tag{12}$$

$$FB_{b,i} \geq B_{i,j} - M \cdot W_{j,i,b} \quad \forall j \in J, \quad \forall i \in I, \quad \forall b \in MB_i \tag{13}$$

$$C_{ij} \geq 0 \quad \forall i \in I, \quad \forall j \in J \tag{14}$$

$$B_{ij} \geq 0 \quad \forall i \in I, \quad \forall j \in J \tag{15}$$

$$SB_{b,i} \geq 0 \quad \forall b \in M_i, \quad \forall i \in I \tag{16}$$

$$FB_{b,i} \geq 0 \quad \forall b \in M_i, \quad \forall i \in I \tag{17}$$

$$X_{j,k,i} \in \{0,1\} \quad \forall i \in I, \quad \forall k, j \in J, j \neq k \tag{18}$$

$$W_{j,i,b} \in \{0,1\} \quad \forall j \in J, \quad \forall i \in I, \quad \forall b \in MB_i \tag{19}$$

The objective function (1) is to minimize the maximum lateness. Constraint set (2) allows for the calculation of the maximum lateness. Equation (3) describes the lateness of job j. Constraint set (4) ensures that each job is scheduled only once on each machine. Constraint set (5) guarantees that each job has at most one successor job on each machine. Equation (6)

specifies that each machine must have a job that initiates the sequence. Equation (7) ensures that the starting time of an operation of a job must be greater than or equal to the completion time of the previous operation of the same job. Constraint set (8) calculates the completion time of an operation of a job, considering the setup time of the previous job processed just before the operation. Constraint set (9) guarantees that the setup times and operations of two jobs do not overlap on the same machine. Equations (10)–(13) ensure that an operation of any job is performed either before or after a repair time for a breakdown, i.e., the operation is not processed while the repair is being done. Constraint sets (14)–(19) indicate the domain of the decision variables.

4 A simheuristic approach based on tabu search

In this section the proposed tabu search (TS) metaheuristic is explained. The main components include the coding scheme, the initial solution, the exploration of the neighborhood, and the utilization of Monte Carlo simulation (MCS) to achieve a stochastic simheuristic approach.

4.1 Coding scheme and the initial solution

Different coding schemes have been used in JSSP literature, and one of the most popular is the permutation-based approach. In this paper, the job-order coding is used, which denotes the order in which the jobs' first operation is arranged. Therefore, the size of the sequence is equal to the number of jobs. For example, a 6×6 JSSP (6 jobs and 6 machines, 36 operations), randomly generates a vector such as [6,2,1,4,5,3]. For those jobs that share the same machine for the first operation, the priority in the machine's queue is determined by the position of the job in the vector. For instance, in the given example, job #2 and job #4 share machine #1. Since job #2 is placed before job #4 in the sequence, machine #1 processes job #2 first and then job #4. As mentioned earlier, one way to initialize the search procedure is by generating a random permutation sequence. However, priority rules can also be utilized, and in this case, the earliest due date (EDD) rule was explored to schedule jobs' first operation (Su et al., 1998; Vinod & Sridharan, 2011).

The job-order coding scheme was chosen for three reasons. Firstly, the vector size remains as small as possible, unlike the working procedure where the vector´s size is equal to the number of operations (the coding determines the sequence of each job on each machine). Secondly, considering that a resource is not kept idle if it can start an activity (i.e., non-delay scheduling), once a job finishes its first operation it immediately moves to the next machine in its sequence, and so on, until all its operations are processed. By using the working procedure, sequences with idle times are analyzed in the exploration process, thus enlarging the search space and requiring more computational cost. Lastly, since the job-ordered permutation approach determines the allocation of the first operation, it allows for the integration of priority rules for sequencing the remaining operations. Priority rules such as SPT, EDD, slack (SL), waiting time in the next queue (WINQ), among others, can be chosen depending on the objective function. Those priority rules are used once a machine finishes an operation, and there are two or more jobs in the machine's queue, allowing the machine to select the next job according to the rule. In this paper, given that the objective function is due-date dependent, the EDD rule was chose as the tiebreaker.

4.2 The exploration of neighbor solutions

When the initial solution is complete, the TS begins constructing neighbor solutions by exchanging the positions of two jobs in the sequence. Figure 2 illustrates the process of neighborhood construction. Finding neighbors can be done by the exhaustive procedure, which involves considering all possible combinations. In this case, $\sum_{i=2}^{n-1} i$ neighbors must be analyzed (where n is the number of jobs). As an alternative to the exhaustive neighborhood exploration, probabilistic TS assignes a probability to each move (i.e., position exchange). If this probability is lower than a threshold probability (TP in Fig. 2), the move is executed and the neighbor is analyzed. Otherwise, the neighbor solution is not considered. The main advantage of probabilistic TS is the computational efficiency; however, determining the optimal value for the threshold probability requires experimentation. If the threshold probability is set to 1.0, then the neighborhood exploration becomes exhaustive. The best neighbor solution

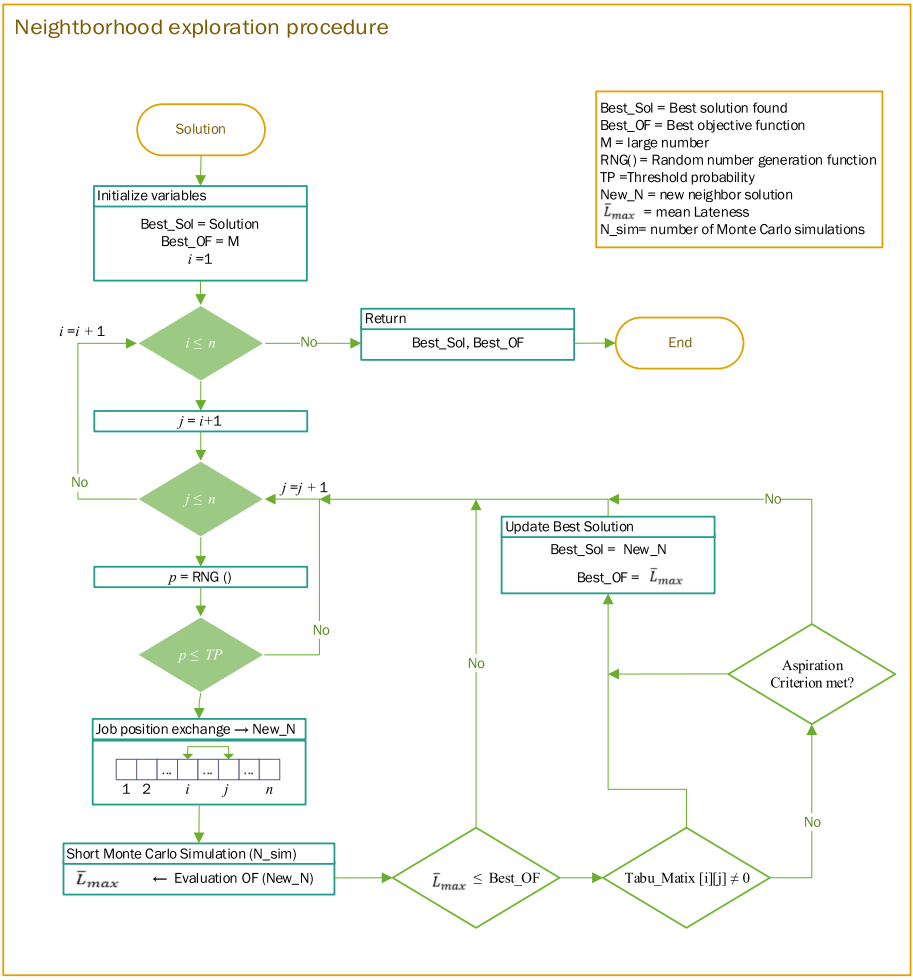


Fig. 2 Neighborhood exploration

becomes the current solution, and the neighborhood exploration restarts with this solution until the search process concludes. In the following, the integration of MCS into the process of neighborhood construction is explained.

4.3 The simheuristic approach using Monte Carlo simulation (MCS)

As seen Fig. 2, during the neighborhood construction, each neighbor solution undergoes a short MCS to explore different scenarios. These scenarios involve randomly generated times between failures (TBF) and times to repair (TTR). The MCS parameters include the number of simulations, the mean times between failures (MTBF), the mean times to repair (MTTR) and the distribution that represents breakdown behavior.

To perform the MCS, failure and repair times were tested with exponential and log-normal distributions across different instances. In order to logically assign those distribution parameters, namely the expected values (MTTR and MTBF) and their standard deviations (σ_{TTR} and σ_{TBF}), the Ag parameter, proposed by Gholami et al. (2009), was utilized. The Ag parameter, described in Eq. (20), represents the proportion of mean time that a machine spends "in repair" relative to its average total time between breakdowns.

$$\mathrm{Ag} = \frac{\mathrm{MTTR}}{\mathrm{MTBF} + \mathrm{MTTR}} \tag{20}$$

Based on the studies conducted by Adibi et al. (2010), Gholami et al. (2009) and Shahrabi et al. (2017), the MTTR value was taken as $5*\overline{P}$, where $\overline{P}$ represents the average processing time of each machine. Therefore, the repair time takes five times the average processing time. From Eq. (20) the value of MTBF can be calculated as presented in Eq. (21).

$$\mathrm{MTBF} = \frac{\mathrm{MTTR}(1 - \mathrm{Ag})}{\mathrm{Ag}} = \frac{5 * \overline{\mathrm{P}}(1 - \mathrm{Ag})}{\mathrm{Ag}} \tag{21}$$

Particularly for the log-normal distribution, the standard deviations of the times to repair and times between failures (σ_{TTR} and σ_{TBF}) are calculated as follows:

$$\sigma_{TBF} = \mathrm{CV} * \mathrm{MTBF} \tag{22}$$

$$\sigma_{TTR} = \mathrm{CV} * \mathrm{MTTR} \tag{23}$$

For this particular case, it is assumed that a machine breakdown triggers a reactive maintenance that starts right after the malfunction arises. Figure 3 depicts three possible scenarios. In the first situation (*Scenario A*), the machine fails while empty, therefore such failure does not impact job scheduling. In *Scenario B*, the machine fails during setup, and consequently, the setup procedure must be executed when the machine becomes available. Lastly, *Scenario C* describes the situation in which a machine breaks down during job processing, causing a right-time shift for the ongoing operation and for all the remaining operations scheduled after the machine failure.

During the TS procedure, the best solution with its objective function value are stored in memory. As seen in Fig. 2, the aspiration criterion is applied if a neighbor solution is listed as a forbidden move in the Tabu Matrix but has a better objective function value compared to the best value found so far.

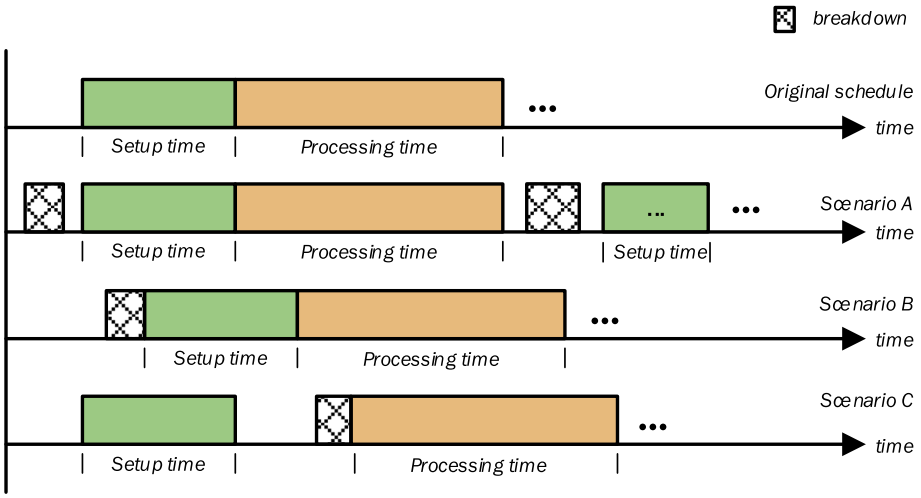


Fig. 3 Impact of machine failures on job scheduling

4.4 Simheuristic with intensive simulation

In addition to incorporating MCS during the neighborhood construction, a number of simulations are applied to the best solution found by the metaheuristic. The aim of conducting intensive simulations is to obtain a more accurate prediction of the mean maximum lateness across various machine failure scenarios. Pseudocode #1 explains the TS procedure, which terminates after a certain number of iterations without any improvement.

Pseudocode #1. Tabu Search Procedure

```
1   Initialize tabu_list, max_iterations, tabu_list_size
2   Construct initial solution using EDD rule
3   Best_sol := initial solution
4   Best_of := Objective Function (initial solution )
5   iter_w_imp = 0   /*counter for iterations without any improvement
6       Repeat
7           Explore neighborhood   /* as described in Figure 2
8           Select best neighbor → new_solution, new_of
9           Update tabu_list with tabu_list_size
10          If new_of < Best_of
11              Best_sol := new_solution   /*replace the current solution
12              Best_of := new_of          /*replace the current objective function value
13              iter_w_imp = 0
14          Else
15              iter_w_imp += 1
16          End if
17      Until iter_w_imp >= max_iterations
18      Execute L_max ← Intensive Monte Carlo Simulation with Best_sol
19  Return Best_sol, L_max
```

Table 1 Benchmark instances

Instance	Authors	#Jobs	#Machines
ft06	Fisher & Thompson (1963)	6	6
la01	Lawrence (1984)	10	5
abz5	Adams et al. (1988)	10	10
la06	Lawrence (1984)	15	5
la21	Lawrence (1984)	15	10
la36	Lawrence (1984)	15	15
ft20	Fisher & Thompson (1963)	20	5
la26	Lawrence (1984)	20	10
abz7	Adams et al. (1988)	20	15
yn1	Yamada & Nakano (1992)	20	20

5 Computational experiments

This section presents a series of experiments conducted to evaluate the proposed simheuristic approach. The section is divided into six parts. Sect. 5.1 provides a detailed description of the instances used. Section 5.2 explains the implementation of two dispatching rules to compare the results of the tabu search (TS) in small and large instances. Sect. 5.3 discusses the parameter tuning of the simheuristic. Sect. 5.4 evaluates the performance of the TS simheuristic against the simulated solutions of the dispatching rules and the proposed Mixed Integer Linear Programming (MILP) approach for the instance ft06. This specific instance was chosen because it is the only one for which the MILP has found an optimal solution. Finally, Sect. 5.5 compares the results of the TS simheuristic with the simulated solutions of the dispatching rules.

5.1 Instances

The benchmark instances listed in Table 1 were used.[1] Each instance only reports processing times and machine sequences for each job. On the contrary, sequence-dependent setup times (S_{kji}) on each machine were uniformly generated in the interval [1, max_times], where max_times represents the total processing time on each machine. Additionally, job's due dates (D_j) were generated with a discrete uniform distribution based on Eq. (25). The parameters used in Eq. (25) include the average value of due dates $\boldsymbol{\mu}$, which depends on the tightness factor $\boldsymbol{\tau}$ that indicated the percentage of jobs expected to arrive late, and the best reported makespan (C_{max}) for each benchmark instance (see Eq. 24). $\boldsymbol{\tau}$ values of 0.3 and 0.6 were used for all instances, with the former allowing for looser due dates compared to the latter. The parameter due date range ($\boldsymbol{R}$) took on values of 0.25 and 0.5 to generate values with different levels of due-date variation. As a result, a total of four combinations of $\boldsymbol{\tau}$ and $\boldsymbol{R}$ were used, resulting in four different combinations of due dates for each set of processing times, jobs and machines.

$$\mu = (1 - \tau)(C_{max}) \tag{24}$$

[1] https://www.eii.uva.es/elena/Elena%20Perez%20Vazquez_archivos/files_optimaJSSP/jobshop1.txt

$$D_j = \left[\mu\left(1 - \frac{R}{2}\right), \mu\left(1 + \frac{R}{2}\right)\right] \tag{25}$$

For the experimental data regarding the mean times between failures (MTBFs) and mean times to repair (MTTRs), two values of Ag (as defined in Eq. 20) were used: 0.05 and 0.15. Additionally, three different values of coefficients of variation (0.0, 0.5 and 1.0) are proposed for the log-normal distribution, as the exponential distribution always has a coefficient of variation (CV) of 1.0. The CV 0.0 in log-normal distribution was used for the deterministic case, as it is equivalent, while 0.5 and 1.0 for stochastic events.

5.2 Dispatching rules

In order to compare the results reported by the proposed simheuristic approach, the earliest due date first (EDD) and the process time plus work in next queue plus negative slack (PT + WINQ + SL) dispatching rules were implemented using discrete-event simulation software, i.e. FlexSim version 2022. Figure 4 shows the simulation model composed of: (1) four sources, each one is in charge of generating a number of jobs according to the instance being evaluated. If other instances are analyzed, more sources can be easily added. Depending on the instance, just one source generates jobs. (2) An initial queue that stores all jobs, connected to all machine queues. (3) One queue per machine to store pending jobs while the machine is busy; (4) 20 machines (because of the largest instance studied in this paper) which execute either the EDD or the PT + WINQ + SL rule to select the next job to process; and finally, (5) a final queue that stores jobs that have finished their sequence. Jobs are called items in the model, and each item is represented by a colored box in Fig. 4. Each item has a label that is dynamically updated as the job progresses in its sequence. This label stores the priority index value that each machine uses to select the next job. The index value is calculated depending on the dispatching rule equation, as described by (Mizrak & Bayhan, 2006). Failure behavior is set for each machine, and depending on the probability distribution and MTBF/MTTR values, the simulation model generates random failures during the simulation time. For example, in Fig. 4, Processor 5 is in failure, therefore, the machine becomes unavailable. Two dashboards

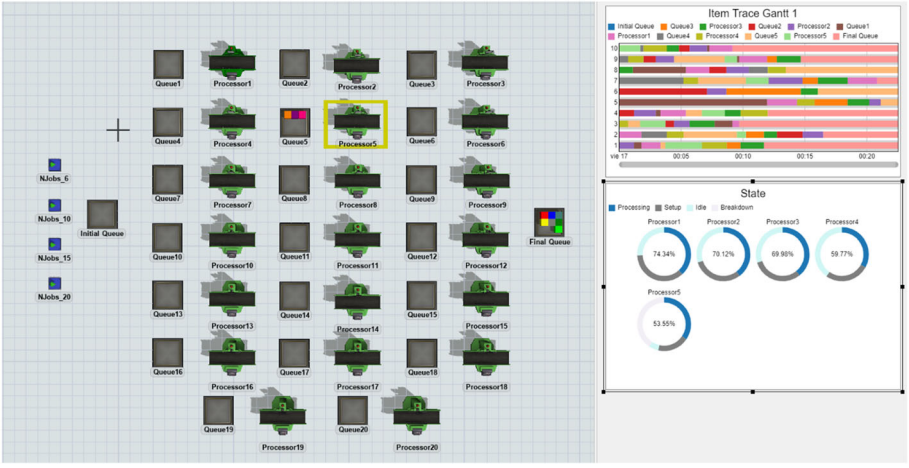


Fig. 4 Flexsim model implementing dispatching rules

are used to visualize production progress: (1) a Gantt chart that helps visualize setup times and processing times for each job, and (2) one pie chart diagram per machine to visualize the machines' states (i.e., idle, setup, processing, or breakdown). Global tables and performance metrics are used to store all data related to each instance, including finish times, job lateness, and the maximum lateness. The Experimenter add-in was used to perform 50 replications for each instance to obtain the mean maximum lateness, the standard deviation, and the 95% confidence interval.

5.3 Tabu search parameter tuning

To perform the parameterization of the simheuristic, the number of iterations without improvement (NIT) and the size of the tabu memory (TM) were considered as the main factors in the computational experiment. Three independent experiments were conducted based on the instance sizes. The first experiment was executed with **ft06** instances, which are considered small. The second experiment included medium-sized instances, namely **la01** to **la36**, which range from 10 to 15 jobs. Finally, the third experiment focused on instances with 20 jobs, specifically **ft20** to **yn1**. For the second experiment, the 32 instances corresponding to **la36** were used, and for the third experiment, the 32 instances corresponding to **yn1** were utilized. In total, 96 instances were used for parameter tuning.

For the NIT factor, values of 5, 10, 15, 30 and 100 iterations were chosen after conducting some preliminary experiments. Regarding the TM factor, values around the number of jobs were utilized, specifically 5, 10, 15, and 20. In each experiment, 12 treatments were analyzed by pairing the values of both factors. The response variable was defined as the improvement percentage of mean expected maximum lateness. To ensure proper parameterization, considering that the assumptions of normality and homoscedasticity were not met, we conducted a separate ANOVA-Type statistic for small, medium, and large instances. The aim was to identify the optimal combination of NIT and TL parameters for each instance size.

The ANOVA-Type statistics revealed that neither the NIT factor nor the TM factor showed statistical significance. Based on these results, a descriptive analysis was conducted to select the combination of NIT and TM that provided the lowest average of expected maximum lateness while considering computational effort. Figure 5 presents the interaction plots of TS parameters for each instance size.

For small instances, the combinations that yield lower average values of expected maximum lateness are as follows: NIT 30 TM 5, NIT 100 TM 5, and NIT 5 TM 10. Among them, the combination NIT 30 TM 5 requires the least computational effort. Additionally, this combination has a TM value that is closest to the size of the vector codification of the tabu for these instances, which is 6 jobs. As a detailed example of the convergence curves provided by the simheuristic, Fig. 6 shows the convergence curves for a specific small instance run with a log-normal distribution with a CV = 1 and an exponential distribution. The blue lines represent the expected maximum lateness of the best solution at each value of NIT, and the red line represents the expected maximum lateness of the best global solution at each value of NIT. It can be observed that the convergence is achieved after NIT = 25 for the log-normal distribution, whereas for the exponential distribution, it occurs after NIT = 5. Despite both graphs shown in Fig. 6 having the same coefficient of variation (CV = 1), and the MTTRs and MTBFs values being the same for each machine in both executions, the exponential distribution exhibits higher expected maximum lateness values compared to the log-normal distribution.

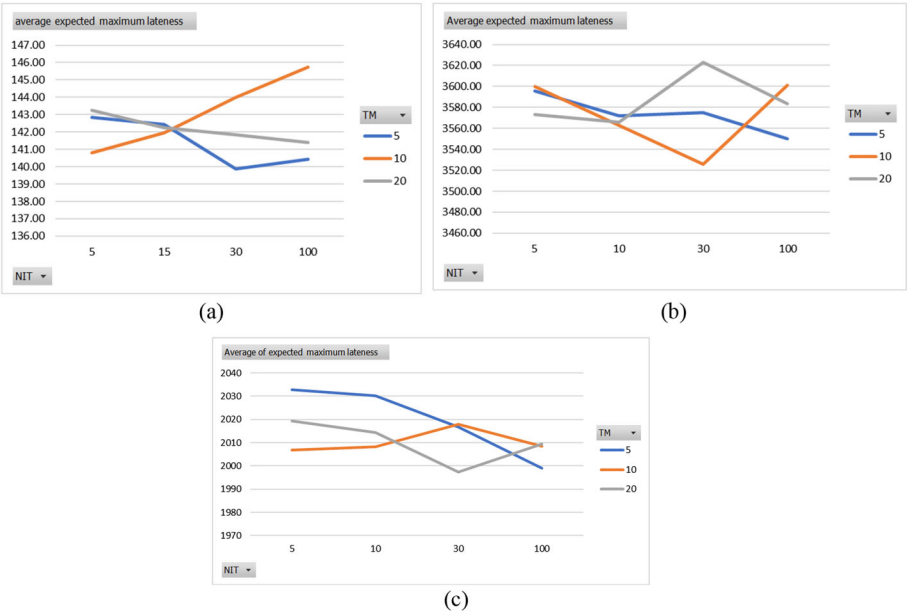


Fig. 5 Results of parameter tunning: **a** small instances, **b** medium instances, **c** large instances

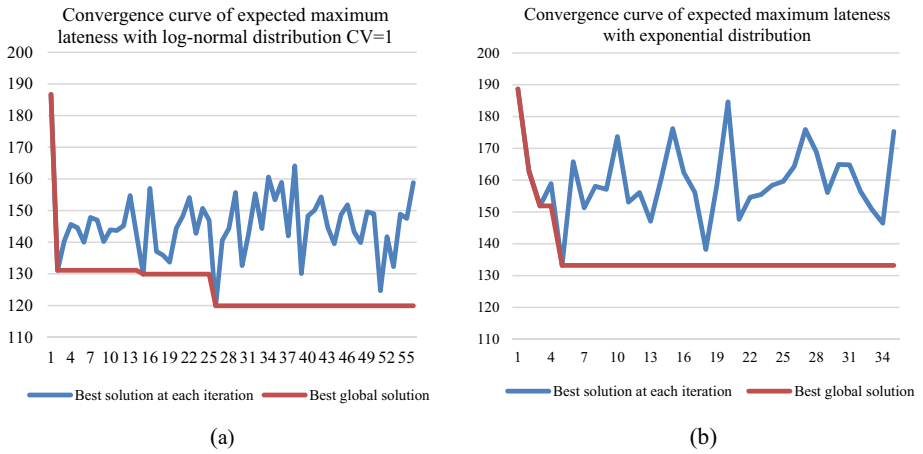


Fig. 6 Convergence curves of simheuristic for one small instance: **a** executed with log-normal distribution and CV = 1, **b** executed with exponential distribution

For medium instances, the lowest average value is 3525.7 (NIT 30 TM 10), followed by 3550.31 (NIT 100 TM 5). Considering that the combination NIT 100 TM 5 requires approximately three times more computational effort, the combination NIT 30 TM 10 was chosen for medium-sized instances. Similarly to small instances, a TM value of 10 is closer to the size of the codification vectors (job instance sizes), which are 10 and 15.

Table 2 Parametrization selected for the simheuristic TS

Type of instances	NIT	TM
Small (6 × 6)	30	5
Medium (10 × 5, 10 × 10, 15 × 5, 15 × 10, 15 × 15)	30	10
Large (20 × 5, 20 × 10, 20 × 15)	30	20

For large instances, the lowest average value is 1997.34 (NIT 30 TM 20), followed by 1999.13 (NIT 100 TM 5). Considering that the combination NIT 100 TM 5 requires around three times more computational effort, the combination NIT 30 TM 20 was chosen for large-sized instances. Similar to medium and small instances, a TM value of 20 is closer to the size of the codification vectors (job instance sizes), which is 20. Therefore, the parametrization used for the computational experiments in the following subsections is presented in Table 2.

5.4 Simheuristic vs. simulated solutions given by EDD, PT + WINQ + SL and MILP model

The objective of this section is to compare the results obtained with the simheuristic against the simulated solutions provided by the EDD dispatching rule, PT + WINQ + SL dispatching rule and the proposed MILP model for the small instances (ft06: 6 × 6). The MILP model was implemented in AMPL language and run in NEOS server. The optimal solutions obtained were then simulated with 50 replications of Monte Carlos simulation (MCS) (see Pseudocode 1). Larger instances were also tested in MILP model, but no optimal solution was obtained after 8 h of running. The performance measure for comparisons is the relative percentage deviation (RPD), as presented in Eq. (26).

$$\mathrm{RPD}_{\mathrm{TSvsMethod}} = \frac{\left(\overline{L}_{max\,simheuristic} - \overline{L}_{max\,method}\right)}{\overline{L}_{max\,method}} * 100 \tag{26}$$

Table 3 shows that the expected maximum lateness values are consistently lower when using the proposed TS simheuristic, except for the case of MILP with Log-normal_0 (i.e., the deterministic case). It improves the EDD dispatching rule by an average of 11.34%, the PT-WINQ + SL dispatching rule by 14.92%, and the simulated solutions of the proposed MILP model by 7.12. In Log-normal_0, the MILP provides a better solution than the deterministic TS, with a GAP of approximately 7.54%. Nevertheless, the TS simheuristic improves the simulated solution of the MILP model by an average of 10.53% for the stochastic cases (Log-normal_0.5, Log-normal_1, and Exponential). This is because the proposed approach is specifically designed to solve the SJSSP, where the TS simheuristic consistently provides better expected maximum lateness than the simulation of the MILP model, as well as the EDD or PT + WINQ + SL dispatching rules, for all the cases. The fact that the TS simheuristic provides better solutions than the simulation of deterministic procedures highlights the importance of implementing methods that account for uncertain events, rather than relying solely on deterministic solutions. Even though deterministic solutions are simulated to account for uncertainty, the optimization method does not anticipate variability, resulting in inefficient utilization of the system.

Due to fact that the results of expected maximum lateness did not fulfilled the assumptions of normality and homoscedasticity, a non-parametric rank-based ANOVA-Type Statistic

Table 3 Expected maximum lateness of methods and RPD

Distribution_CV	Ag	Method				RPD		
		TS	EDD	PT + WINQ + SL	MILP	$RPD_{TSvsEDD}$	$RPD_{TSvsPT+WINQ+SL}$	$RPD_{TSvsMILP}$
Log-normal_0	0.05	98.75	112.00	123.73	94.50	− 11.83%	− 20.19%	4.50%
	0.15	104.50	121.50	123.68	94.50	− 13.99%	− 15.51%	10.58%
Total Log-normal_0		**101.63**	**116.75**	**123.70**	**94.50**	**− 12.96%**	**− 17.85%**	**7.54%**
Log-normal_0.5	0.05	98.99	112.66	123.79	114.79	− 12.13%	− 20.03%	− 13.76%
	0.15	121.80	143.48	150.05	140.82	− 15.11%	− 18.83%	− 13.51%
Total Log-normal_0.5		**110.39**	**128.07**	**136.92**	**127.81**	**− 13.80%**	**− 19.37%**	**− 13.62%**
Log-normal_1	0.05	105.24	120.68	132.73	125.53	− 12.80%	− 20.71%	− 16.17%
	0.15	158.17	172.95	163.95	165.51	− 8.55%	− 3.53%	− 4.43%
Total Log-normal_1		**131.70**	**146.82**	**148.34**	**145.52**	**− 10.29%**	**− 11.22%**	**− 9.50%**
Exponential	0.05	121.54	136.84	146.51	135.77	− 11.18%	− 17.05%	− 10.48%
	0.15	163.20	176.37	178.15	175.33	− 7.47%	− 8.40%	− 6.92%
Total Exponential		**142.37**	**156.60**	**162.33**	**155.55**	**− 9.09%**	**− 12.30%**	**− 8.47%**
Total general		**121.52**	**137.06**	**142.82**	**130.84**	**− 11.34%**	**− 14.92%**	**− 7.12%**

The rows marked in bold represent partial totals for each combination of probability distribution and coefficient of variation, regardless of the Ag values

(Brunner et al., 1997) was performed to evaluate the significance of three factors (method, combination Distribution_CV, and Ag) in relation to the expected maximum lateness. Using a significance level of 5%, all three factors were found to have a significant effect on the expected maximum lateness, as well as the double interaction between Ag and the combination Distribution_CV (see Table 4).

To analyze these results in more detail, Table 5 presents the factor information that provides the confidence intervals of ranks under a confidence level of 95% for each of the three mentioned factors. The lower the values of rankings, the lower the expected maximum lateness, and the higher the values of rankings, the higher the expected maximum lateness.

For the method factor, it can be observed that the proposed simheuristic TS has a confidence interval is lower than the other three solution methods, and it does not overlap with the others. In constrast, the confidence intervals of EDD, PT + WNQ + SL, and MILP overlap with

Table 4 ANOVA-type statistic for small instances

Source of variation	Statistic	df1	df2	*p*-Value
Method	7.4866	2.3128	55.156	**0.0008**
Ag	75.3054	1	55.156	**0.0000**
Method:Ag	0.781	2.3128	55.156	0.4798
Distribution_CV	40.9829	2.9349	55.156	**0.0000**
Method:Distribution_CV	1.2094	6.6844	55.156	0.3140
Ag:Distribution_CV	6.4682	2.9349	55.156	**0.0009**
Method:Ag:Distribution_CV	0.203	6.6844	55.156	0.9811

The values marked in bold are those *p*-values less than the significance level of 0.05, meaning those factors or interactions that are statistically significant

Table 5 Factor information of ANOVA-Type Statistic for small instances

	Rel.Effect	Std.error	Lower	Upper
Method				
EDD	0.5370	0.0302	0.4776	0.5953
MILP	0.4814	0.0179	0.4466	0.5165
PT + WINQ + SL	0.5894	0.0332	0.5230	0.6526
TS	0.3922	0.0219	0.3501	0.4359
Ag				
0.05	0.3663	0.0142	0.3390	0.3945
0.15	0.6337	0.0142	0.6055	0.6610
Distribution_CV				
Log-normal_0	0.2534	0.0237	0.2099	0.3026
Log-normal_0.5	0.4393	0.0287	0.3842	0.4960
Log-normal_1	0.6036	0.0258	0.5521	0.6530
Exponential	0.5931	0.0255	0.6513	0.7510

each other, indicating that these methods do not have a statistically significant difference. Indeed, the proposed TS is significantly different and superior than the other three methods.

Regarding the Ag, as expected, the objective function is significantly lower for an Ag of 0.05 compared to an Ag of 0.15, indicating that as the Ag increases, the expected maximum lateness also increases. Finally, regarding the combination of Distribution_CV, two analysis can be performed. For the log-normal distribution, as the coefficient of variation increases, the expected maximum lateness also increases. Furthermore, when comparing Log-normal_1 with Exponential, the confidence intervals overlap, but only to a very small extent. This generally indicated that despite having the same coefficient of variation and using the same values of MTTR and MTBF, exponential distributions tend to yield greater expected maximum lateness than Log-normal distributions with CV $= 1$, because exponential behavior increases variability.

5.5 Results and analysis for medium and large instances

The aim of this section is to compare the results obtained with the simheuristic against the simulated solutions provided by the EDD and PT + WINQ + SL dispatching rules for medium and large-sized instances (la01, abz5, la06, la21, la36, ft20, la26, abz7, and yn1), which range in size from 10×5 to 20×20. The experiment in this case consisted of four factors: solution method (TS, EDD, PT + WINQ + SL), instance size (10×5, 10×10, 15×5, 15×10, 15×15, 20×5, 20×10, 20×15, and 20×20), Distribution_CV, and Ag. Table 6 presents the averages of expected maximum lateness by instance size, the combination of Distribution_CV, and the solution method. For all instance sizes and probability distributions, the TS simheuristic provides better results than the simulated solutions of both dispatching rules.

Table 7 shows the average results summarized for each combination of Distribution_CV, Ag, and solution method, and presents the RPD defined in Eq. (26). The results demonstrate that the expected maximum lateness values are consistently lower when utilizing the proposed TS simheuristic, resulting in an average improvement of 11.72% over the EDD dispatching rule and 14.08% over the PT-WINQ + SL dispatching rule.

The non-parametric rank-based ANOVA-Type Statistic was performed to evaluate the effect of the mentioned four factors and their interactions on the expected maximum lateness. Using a significance level of 5%, all four factors were found to have a significant effect on the expected maximum lateness (see Table 8). All significant p-values are indicated in bold.

To analyze these results in more detail, Table 9 presents the factor information that provides the confidence intervals of ranks under a confidence level of 95% for each of the three mentioned factors. The lower the values of rankings, the lower the expected maximum lateness, and the higher the values of rankings, the higher the expected maximum lateness.

For the method factor, it can be observed that the proposed simheuristic TS has a confidence interval is lower than the other three solution methods, and it does not overlap with the others. In contrast, the confidence intervals of EDD and PT + WNQ + SL overlap with each other, indicating that these methods do not have a statistically significant difference. Indeed, the proposed TS is significantly different and superior than the other three methods.

As in the results of small instances, as the Ag increases the expected maximum lateness also increases. In relation to Distribution_CV, the same two analysis than in small instances were performed. For the log-normal distribution, as the coefficient of variation increases, the expected maximum lateness also increases and the intervals do not overlap. In the second analysis, that compares the Log-normal_1 with Exponential, the confidence intervals overlap,

Table 6 Average expected maximum lateness by instance size, combination Distribution_CV and solution method

Instance size	Log-normal_0			Log-normal_0.5			Log-normal_1			Exponential		
	EDD	PT + WINQ + SL	TS	EDD	PT + WINQ + SL	TS	EDD	PT + WINQ + SL	TS	EDD	PT + WINQ + SL	TS
10_5	1259.00	1249.10	898.50	1321.56	1344.15	1055.01	1519.04	1545.95	1236.66	1620.63	1672.33	1433.82
10_10	2215.13	2222.25	1985.00	2424.03	2422.41	2236.27	2833.87	2828.74	2724.49	3149.73	3162.84	3020.02
15_5	1830.38	1989.29	1372.75	1939.05	2019.95	1495.12	2129.40	2180.74	1739.87	2223.13	2254.81	1998.73
15_10	2454.25	2487.74	1812.00	2650.85	2694.34	2099.72	2987.69	2998.23	2675.34	3095.18	3130.08	2839.21
15_15	2785.38	2790.32	2336.38	3022.90	3080.40	2799.50	3531.17	3507.79	3437.98	3679.44	3873.91	3669.03
20_5	2623.75	3181.48	2020.00	2794.96	3026.21	2305.52	2973.63	3268.97	2650.76	3094.57	3219.68	2774.20
20_10	2502.50	2585.07	1853.00	2714.09	2764.01	2295.67	3108.39	3194.41	2884.13	3183.42	3281.37	3216.30
20_15	1363.50	1348.64	1076.25	1541.60	1556.52	1355.35	1808.04	1857.74	1656.57	1886.50	1945.19	1846.36
20_20	1894.75	1858.63	1378.63	1988.23	1990.06	1835.55	2323.81	2322.41	2286.37	2452.61	2454.61	2438.71

Table 7 Expected maximum lateness of methods and RPD for medium and large instance sizes

Distribution_CV	Ag	Method			RPDs	
		TS	EDD	PT + WINQ + SL	RPD TS versus EDD (%)	RPD TS vs PT + WINQ + SL (%)
Log-normal_0	0.05	1498.00	1884.61	1896.70	− 20.51	− 21.02
	0.15	1775.89	2321.75	2483.86	− 23.51	− 28.50
Total Log-normal_0		1636.94	2103.18	2190.28	− 22.17	− 25.26
Log-normal_0.5	0.05	1586.22	1975.45	1984.89	− 19.70	− 20.09
	0.15	2297.71	2557.28	2659.12	− 10.15	− 13.59
Total Log-normal_0.5		1941.97	2266.36	2322.01	− 14.31	− 16.37
Log-normal_1	0.05	1896.95	2188.86	2216.61	− 13.34	− 14.42
	0.15	2834.64	2970.04	3051.16	− 4.56	− 7.10
Total Log-normal_1		2365.80	2579.45	2633.89	− 8.28	− 10.18
Exponential_1	0.05	2139.33	2305.34	2393.31	− 7.20	− 10.61
	0.15	3024.31	3113.59	3161.09	− 2.87	− 4.33
Total Exponential_1		2581.82	2709.47	2777.20	− 4.71	− 7.04
Total general		2131.63	2414.61	2480.84	− 11.72	− 14.08

Table 8 ANOVA-Type Statistic for medium and large instances

Source of variation	Test statistic	df1	df2	*p*-value
Method	39.3677	1.8608	144.5530	**0.0000**
Instance	149.2012	5.2312	144.5530	**0.0000**
Method:Instance	1.2038	8.4215	144.5530	0.2992
Ag	409.0242	1.0000	144.5530	**0.0000**
Method:Ag	0.3258	1.8608	144.5530	0.7068
Instance:Ag	4.0306	5.2312	144.5530	**0.0016**
Method:Instance:Ag	0.3849	8.4215	144.5530	0.9334
Distribution_CV	90.2349	2.9770	144.5530	**0.0000**
Method:Distribution_CV	2.1899	5.4790	144.5530	0.0527
Instance:Distribution_CV	1.2937	14.5685	144.5530	0.2152
Method:Instance:Distribution_CV	0.1379	22.5369	144.5530	0.9999
Ag:Distribution_CV	7.8402	2.9770	144.5530	**0.0000**
Method:Ag:Distribution_CV	0.8033	5.4790	144.5530	0.5590
Instance:Ag:Distribution_CV	0.3135	14.5685	144.5530	0.9927
Method:Instance:Ag:Distribution_CV	0.1328	22.5369	144.5530	0.9999

The values marked in bold are those *p*-values less than the significance level of 0.05, meaning those factors or interactions that are statistically significant

Table 9 Factor information of ANOVA-Type Statistic for small instances

	Rel.Effect	Std.Error	Lower	Upper
Method				
EDD	0.5313	0.0120	0.5078	0.5546
PT + WINQ + SL	0.5400	0.0121	0.5162	0.5637
TS	0.4287	0.0116	0.4061	0.4516
Ag				
0.05	0.3937	0.0080	0.3781	0.4095
0.15	0.6063	0.0080	0.5905	0.6219
Distribution_CV				
Log-normal_0	0.3796	0.0142	0.3522	0.4077
Log-normal_0.5	0.4509	0.0147	0.4223	0.4798
Log-normal_1	0.5588	0.0148	0.5295	0.5877
Exponential	0.6108	0.0143	0.5825	0.6383

but only to a very small extent. This generally indicated that despite having the same coefficient of variation and using the same values of MTTR and MTBF, exponential distributions tend to yield greater expected maximum lateness than Log-normal distributions with CV = 1, because exponential behavior increases variability.

Execution times of the solution methods are presented in Table 10. The dispatching rules always provide very fast execution times, taking less than 8 s for the 50 replicates. However, this comes at the cost of increasing the average maximum lateness by more than 14%, which is quite high. The TS presents a maximum of 18 min for solving the largest instances, which is very fast considering that it solves the stochastic problem. In the case of the MILP model, only the small instances (6×6) provided a solution. However, once this solution is simulated, the average maximum lateness is 12% larger than the one provided by the proposed

Table 10 Execution times of solution methods by instance sizes

Instance size	Average execution times (s)			
	EDD	PT + WINQ + SL	TS	MILP
6_6	6.43	4.78	56.68	38.11 (33.60 + 4.51)
10_5	6.56	5.14	63.60	> 28,800
10_10	7.10	5.18	106.58	> 28,800
15_5	6.84	5.24	271.97	> 28,800
15_10	6.87	5.37	398.62	> 28,800
15_15	7.35	5.46	511.29	> 28,800
20_5	7.14	5.19	767.23	> 28,800
20_10	7.16	5.55	641.34	> 28,800
20_15	7.93	5.86	932.07	> 28,800
20_20	6.86	6.37	1094.65	> 28,800

TS. Therefore, the proposed simheuristic yields good results for the stochastic problem in reasonable computational times.

6 Conclusions and future research

This paper focuses on the study of a stochastic job shop scheduling problem (SJSSP) with the objective of minimizing the expected maximum lateness. The main source of uncertainty considered in this research is machine breakdowns. To address this problem, we formulate a mixed-integer linear programming model that incorporates the expected times for repairs and between failures. However, due to the problem's complexity, the model is only capable of providing solutions for very small instances. To overcome this limitation, we propose a tabu search (TS) simheuristic that combines Monte Carlo simulation. The exponential and log-normal distributions are employed to model the times between failures and repair durations, respectively, using different coefficients of variation. During each replica of the Monte Carlo simulation, we generate realizations of these times while maintaining the same expected value for each machine and for a specific combination of probability distribution and coefficient of variation. The proposed simheuristic is parameterized by defining the size of the tabu list and the number of iterations without improvement for three instance sizes: small, medium, and large.

To evaluate the performance of the simheuristic, two computational experiments were conducted. Firstly, for small instances, we compared the results of the simheuristic with the simulation of solutions obtained from the MILP model, as well as the EDD and PT + WINQ + SL dispatching rules. Secondly, for medium and large instances, we compared the simheuristic results with the simulation of solutions provided by the EDD and PT + WINQ + SL dispatching rules. The average percentage improvement in expected maximum lateness was as follows: for small instances, 7.12% compared to the simulation of the best solution found by the mathematical model, 11.34% compared to the simulation of EDD, and 14.92% compared to the simulation of the PT + WINQ + SL dispatching rule. For medium and large instances, an average improvement of 11.72% and 14.08% was achieved, respectively, when compared to the simulation of solutions provided by the EDD and PT + WINQ + SL dispatching rules. These results demonstrate the excellent performance of the proposed approach. It is important to note that the average execution times of the simheuristic were 56.68 s for small instances, 4.51 min for medium instances, and 14.82 min for large instances. In contrast, the MILP model took 38.11 s for small instances, but it did not yield an optimal solution within an 8-h execution time for medium and large instances.

The results also provide further insights into the behavior of the expected maximum lateness under different probability distributions and coefficients of variation. It was observed that higher expected maximum lateness values were obtained with the exponential distribution compared to the Log-normal distribution with a coefficient of variation of 1. Additionally, in the case of the log-normal distribution alone, as the coefficient of variation increases, the expected maximum lateness also increases. These findings highlight the importance of appropriately adjusting the probability distributions for the times between damages and repair times to obtain accurate solutions. Furthermore, it was discovered that solving the deterministic problem and implementing such a solution is not suitable as it leads to significant deviations.

Future work will focus on evaluating the hybridization of TS with other simulation paradigms, such as agent-based simulation, in order to compare their performances. Another

important recommendation to consider is the implementation of another metaheuristic combined with Monte Carlo simulation, such as a genetic algorithm, GRASP, or iterated local search. Additionally, it is suggested to introduce other parameters with random behavior, such as processing times and ready times. Lastly, exploring the use of alternative probability distributions to model the random behavior of times to repair and between failures is also recommended.

Funding Open Access funding provided by Colombia Consortium.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Abdolrazzagh, M., & Adbullah, S. (2017). Job Shop scheduling: classification, constraints and objective funtions. *World Academy of Science-Engineering and Technology, 11*, 4–20.

Abido, M. A. (2002). Optimal power flow using tabu search algorithm. *Electric Power Components and Systems, 30*(5), 469–483. https://doi.org/10.1080/15325000252888425

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science*, *34*(3), 391–401.

Adibi, M. A., Zandieh, M., & Amiri, M. (2010). Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications, 37*(1), 282–287. https://doi.org/10.1016/J.ESWA.2009.05.001

Akers, S. B., & Friedman, J. (1955). A Non-numerical approach to production scheduling problems. *Journal of the Operations Research Society of America, 3*(4), 429–442. https://doi.org/10.1287/opre.3.4.429

Al-Hinai, N., & Elmekkawy, T. Y. (2011). Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics, 132*(2), 279–291. https://doi.org/10.1016/J.IJPE.2011.04.020

Araki, K., & Yoshitomi, Y. (2016). Stochastic job-shop scheduling: A hybrid approach combining pseudo particle swarm optimization and the Monte Carlo method. *Journal of Advanced Mechanical Design, Systems and Manufacturing, 10*(3), 1–10. https://doi.org/10.1299/jamdsm.2016jamdsm0053

Aschauer, A., Roetzer, F., Steinboeck, A., & Kugi, A. (2020a). Efficient scheduling of a stochastic no-wait job shop with controllable processing times. *Expert Systems with Applications, 162*, 113879. https://doi.org/10.1016/j.eswa.2020.113879

Azadeh, A., Negahban, A., & Moghaddam, M. (2012). A hybrid computer simulation-artificial neural network algorithm for optimisation of dispatching rule selection in stochastic job shop scheduling problems. *International Journal of Production Research, 50*(2), 551–566. https://doi.org/10.1080/00207543.2010.539281

Banerjee, B. P. (1965). Single facility sequencing with random execution times. *Operations Research, 13*(3), 358–364. https://doi.org/10.1287/opre.13.3.358

Baykasoglu, A. (2002). Linguistic-based meta-heuristic optimization model for flexible job shop scheduling. *International Journal of Production Research, 40*(17), 4523–4543. https://doi.org/10.1080/00207540210147043

Beck, J. C., & Wilson, N. (2007). Proactive algorithms for job shop scheduling with probabilistic durations. *Journal of Artificial Intelligence Research, 28*, 183–232. https://doi.org/10.1613/jair.2080

Bierwirth, C., & Kuhpfahl, J. (2017). Extended GRASP for the job shop scheduling problem with total weighted tardiness objective. *European Journal of Operational Research, 261*(3), 835–848. https://doi.org/10.1016/j.ejor.2017.03.030

Bilkay, O., Anlagan, O., & Kilic, S. E. (2004). Job shop scheduling using fuzzy logic. *International Journal of Advanced Manufacturing Technology, 23*(7–8), 606–619. https://doi.org/10.1007/S00170-003-1771-2/FIGURES/11

Bollapragada, R., & Sadeh, N. M. (2004). Proactive release procedures for just-in-time job shop environments, subject to machine failures. *Naval Research Logistics (NRL), 51*(7), 1018–1044. https://doi.org/10.1002/NAV.20036

Botta-Genoulaz, V. (2000). Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics, 64*(1–3), 101–111. https://doi.org/10.1016/S0925-5273(99)00048-1

Brunner, E., Dette, H., & Munk, A. (1997). Box-type approximations in nonparametric factorial designs. *Journal of the American Statistical Association, 92*(440), 1494–1502. https://doi.org/10.1080/01621459.1997.10473671

Çaliş, B., & Bulkan, S. (2015). A research survey: Review of AI solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing, 26*(5), 961–973. https://doi.org/10.1007/s10845-013-0837-8

Chen, D., Liu, F., & Luh, P. B. (1997). Scheduling job shops with uncertainties. *Proceedings of the IEEE Conference on Decision and Control, 4*, 3556–3561. https://doi.org/10.1109/CDC.1997.652402

Chen, S., Huang, Z., & Guo, H. (2022). An end-to-end deep learning method for dynamic job shop scheduling problem. *Machines, 10*(7), 573. https://doi.org/10.3390/machines10070573

Cheng, M., Sugi, M., Ota, J., Yamamoto, M., Ito, H., & Inoue, K. (2005). Online job shop rescheduling with reaction-diffusion equation on a graph. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 3219–3224. https://doi.org/10.1109/IROS.2005.1545459

Cheng, T. C. E., Peng, B., & Lü, Z. (2016). A hybrid evolutionary algorithm to solve the job shop scheduling problem. *Annals of Operations Research, 242*(2), 223–237. https://doi.org/10.1007/s10479-013-1332-5

Chiang, P. H., & Torng, C. C. (2016). A production planning and optimisation of multi-mode job shop scheduling problem for an avionics manufacturing plant. *International Journal of Manufacturing Technology and Management, 30*(3/4), 179. https://doi.org/10.1504/IJMTM.2016.077813

de Armas, J., Juan, A. A., Marquès, J. M., & Pedroso, J. P. (2017). Solving the deterministic and stochastic uncapacitated facility location problem: From a heuristic to a simheuristic. *Journal of the Operational Research Society, 68*(10), 1161–1176. https://doi.org/10.1057/s41274-016-0155-6

Fang, J., & Xi, Y. (1997). A rolling horizon job shop rescheduling strategy in the dynamic environment. *The International Journal of Advanced Manufacturing Technology, 13*(3), 227–232. https://doi.org/10.1007/BF01305874

Feng, X., Zhao, Z., & Zhang, C. (2020). Simulation optimization framework for online deployment and adjustment of reconfigurable machines in job shops. In *IEEE International Conference on Industrial Engineering and Engineering Management*, *2020-Decem*, 731–735. https://doi.org/10.1109/IEEM45057.2020.9309782

Feo, T. A., & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization, 6*(2), 109–133. https://doi.org/10.1007/BF01096763

Fernandez-Viagas, V., Ruiz, R., & Framinan, J. M. (2017). A new vision of approximate methods for the permutation flowshop to minimise makespan: State-of-the-art and computational evaluation. *European Journal of Operational Research, 257*(3), 707–721. https://doi.org/10.1016/j.ejor.2016.09.055

Fisher, M. L. (1973). optimal solution of scheduling problems using lagrange multipliers: Part II. In *Symposium on the Theory of Scheduling and its Applications*. https://doi.org/10.1007/978-3-642-80784-8_20

Fisher, H., & Thompson, G. L. (1963). Probabilistic learning combinations of local job-shop scheduling rules. In J. F. Muth & G. L. Thompson (Ed.), *Industrial Scheduling* (pp. 225–251). Prentice-Hall.

Gantt, H. L. (1919). *Organizing for work*. Harcourt.

Gholami, M., Zandieh, M., & Alem-Tabriz, A. (2009). Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *The International Journal of Advanced Manufacturing Technology, 42*, 189–201.

Giacaman, G. J., Medel, R. P., & Tabilo, J. A. (2002). Simulation of the material transporting and loading process in Pedro de Valdivia mine. In *Proceedings of the Winter Simulation Conference*, pp. 1349–1355. https://doi.org/10.1109/WSC.2002.1166401

Gmys, J., Mezmaz, M., Melab, N., & Tuyttens, D. (2020). A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2020.01.039

González-Martín, S., Juan, A., Riera, D., Elizondo, M., & Ramos-González, J. (2016). A Simheuristic algorithm for solving the arc routing problem with stochastic demands. *Journal of Simulation*. https://doi.org/10.1057/jos.2016.11

González-Neira, E. M., & Montoya-Torres, J. R. (2019). A simheuristic for bi-objective stochastic permutation flow shop scheduling problem. *Journal of Project Management, 4*, 57–80. https://doi.org/10.5267/j.jpm.2019.1.003

Gruler, A., Panadero, J., de Armas, J., Moreno Pérez, J. A., & Juan, A. A. (2018). Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs. *Computers & Industrial Engineering, 123*, 278–288. https://doi.org/10.1016/j.cie.2018.06.036

Hao, X., Gen, M., Lin, L., & Suer, G. A. (2017). Effective multiobjective EDA for bi-criteria stochastic job-shop scheduling problem. *Journal of Intelligent Manufacturing, 28*(3), 833–845. https://doi.org/10.1007/s10845-014-1026-0

Hasan, S. M. K., Sarker, R., Essam, D., & Kacem, I. (2011). A DSS for job scheduling under process interruptions. *Flexible Services and Manufacturing Journal, 23*(2), 137–155. https://doi.org/10.1007/S10696-011-9094-3/FIGURES/5

Hill, C. (2011). Negocios Internacionales. In *Corporate environmental orientation: Conceptualization and the case of Andean exporters* (Vol. 30). https://doi.org/10.1039/C4NJ00351A

Holthaus, O. (1999). Scheduling in job shops with machine breakdowns: An experimental study. *Computers & Industrial Engineering, 36*(1), 137–162. https://doi.org/10.1016/S0360-8352(99)00006-6

Horng, S.-C., Lin, S.-S., & Yang, F.-Y. (2012). Evolutionary algorithm for stochastic job shop scheduling with random processing time. *Expert Systems with Applications, 39*(3), 3603–3610. https://doi.org/10.1016/j.eswa.2011.09.050

Jackson, J. R. (1956). An extension of Johnson's results on job IDT scheduling. *Naval Research Logistics Quarterly, 3*(3), 201–203. https://doi.org/10.1002/nav.3800030307

Jacobs, L. W., & Lauer, J. (1994). DSS for job shop machine scheduling. *Industrial Management & Data Systems, 94*(4), 15–23. https://doi.org/10.1108/02635579410059455

Jain, A. S., & Meeran, S. (1999). Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research, 113*(2), 390–434. https://doi.org/10.1016/S0377-2217(98)00113-1

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly, 1*(1), 61–68. https://doi.org/10.1002/nav.3800010110

Juan, A. A., Barrios, B. B., Vallada, E., Riera, D., & Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory, 46*, 101–117. https://doi.org/10.1016/j.simpat.2014.02.005

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives, 2*, 62–72. https://doi.org/10.1016/j.orp.2015.03.001

Kadipasaoglu, S. N., Peixoto, J. L., & Khumawala, B. M. (1999). Global manufacturing practices: An empirical evaluation. *Industrial Management & Data Systems, 99*(3), 101–108. https://doi.org/10.1108/02635579910370652

Lawrence, S. (1984). *Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques (Supplement)*. Carnegie Mellon University, Pittsburgh, PA: Graduate School of Industrial Administration.

Lei, D. (2011). Simplified multi-objective genetic algorithms for stochastic job shop scheduling. *Applied Soft Computing, 11*(8), 4991–4996. https://doi.org/10.1016/J.ASOC.2011.06.001

Lei, D. M. (2012). Minimizing makespan for scheduling stochastic job shop with random breakdown. *Applied Mathematics and Computation, 218*(24), 11851–11858. https://doi.org/10.1016/J.AMC.2012.04.091

Li, Y., & Chen, Y. (2009). Neural network and genetic algorithm-based hybrid approach to dynamic job shop scheduling problem. In *Conference Proceedings-IEEE International Conference on Systems, Man and Cybernetics*, 4836–4841. https://doi.org/10.1109/ICSMC.2009.5346060

Li, J. Q., Sang, H. Y., Han, Y. Y., Wang, C. G., & Gao, K. Z. (2018). Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *Journal of Cleaner Production, 181*, 584–598. https://doi.org/10.1016/j.jclepro.2018.02.004

Li, X., & Gao, L. (2020). A hybrid intelligent algorithm and rescheduling technique for dynamic JSP. *Engineering Applications of Computational Methods, 2*, 345–375. https://doi.org/10.1007/978-3-662-55305-3_17/TABLES/12

Liu, N., Abdelrahman, M. A., & Ramaswamy, S. (2004). A multi-agent model for reactive job shop scheduling. *Proceedings of the Annual Southeastern Symposium on System Theory, 36*, 241–245. https://doi.org/10.1109/SSST.2004.1295656

Makino, T. (1965). On a scheduling problem. *Journal of the Operations Research Society Japan, 8*, 32–44.

Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research*. https://doi.org/10.1287/opre.8.2.219

Meloni, C., Pacciarelli, D., & Pranzo, M. (2004). A rollout metaheuristic for job shop scheduling problems. *Annals of Operations Research, 131*(1), 215–235. https://doi.org/10.1023/B:ANOR.0000039520.24932.4b

Meng, L., Zhang, C., Ren, Y., Zhang, B., & Lv, C. (2020). Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem. *Computers & Industrial Engineering, 142*, 106347. https://doi.org/10.1016/j.cie.2020.106347

Mizrak, P., & Bayhan, G. M. (2006). Comparative study of dispatching rules in a real-life job shop environment. *Applied Artificial Intelligence, 20*(7), 585–607. https://doi.org/10.1080/08839510600779738

Mohan, J., Lanka, K., & Rao, A. N. (2019). A review of dynamic job shop scheduling techniques. *Procedia Manufacturing, 30*, 34–39. https://doi.org/10.1016/j.promfg.2019.02.006

Muhlemann, A. P., Lockett, A. G., & Farn, C. K. (1982). Job shop scheduling heuristics and frequency of scheduling. *International Journal of Production Research, 20*(2), 227–241. https://doi.org/10.1080/00207548208947763

Naderi, B., Tavakkoli-Moghaddam, R., & Khalili, M. (2010). Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan. *Knowledge-Based Systems, 23*(2), 77–85. https://doi.org/10.1016/j.knosys.2009.06.002

Nowicki, E., & Smutnicki, C. (2005). An advanced tabu search algorithm for the job shop problem. *Journal of Scheduling, 8*(2), 145–159. https://doi.org/10.1007/s10951-005-6364-5

Paredis, J., & Van Rij, T. (1992). Simulation and constraint programming as support methodologies for job shop scheduling. *Journal of Decision Systems, 1*(1), 59–77. https://doi.org/10.1080/12460125.1992.10511509

Petrovic, S., Fayad, C., Petrovic, D., Burke, E., & Kendall, G. (2008). Fuzzy job shop scheduling with lot-sizing. *Annals of Operations Research, 159*(1), 275–292. https://doi.org/10.1007/s10479-007-0287-9

Pinedo, M. L. (2016). *Scheduling* (5th ed.). Springer International Publishing.

Pranzo, M., & Pacciarelli, D. (2016). An iterated greedy metaheuristic for the blocking job shop scheduling problem. *Journal of Heuristics, 22*(4), 587–611. https://doi.org/10.1007/s10732-014-9279-5

Quintero-Araujo, C. L., Guimarans, D., & Juan, A. A. (2019). A simheuristic algorithm for the capacitated location routing problem with stochastic demands. *Journal of Simulation*. https://doi.org/10.1080/17477778.2019.1680262

Renna, P. (2009). A performance comparison between efficiency and pheromone approaches in dynamic manufacturing scheduling. *Intelligent Systems for Automated Learning and Adaptation: Emerging Trends and Applications*. https://doi.org/10.4018/978-1-60566-798-0.CH012

Roshan, S., Jooibari, M., Teimouri, R., Asgharzadeh-Ahmadi, G., Falahati-Naghibi, M., & Sohrabpoor, H. (2013). Optimization of friction stir welding process of AA7075 aluminum alloy to achieve desirable mechanical properties using ANFIS models and simulated annealing algorithm. *The International Journal of Advanced Manufacturing Technology*. https://doi.org/10.1007/s00170-013-5131-6

Sel, Ç., & Hamzadayı, A. (2018). A simulated annealing approach based simulation-optimisation to the dynamic job-shop scheduling problem. *Pamukkale University Journal of Engineering Sciences, 24*(4), 665–674. https://doi.org/10.5505/pajes.2017.47108

Shady, S., Kaihara, T., Fujii, N., & Kokuryo, D. (2021). Evolving dispatching rules using genetic programming for multi-objective dynamic job shop scheduling with machine breakdowns. *Procedia CIRP, 104*, 411–416. https://doi.org/10.1016/j.procir.2021.11.069

Shahrabi, J., Adibi, M. A., & Mahootchi, M. (2017). A reinforcement learning approach to parameter estimation in dynamic job shop scheduling. *Computers & Industrial Engineering, 110*, 75–82.

Shahsavari-Pour, N., & Ghasemishabankareh, B. (2013). A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling. *Journal of Manufacturing Systems, 32*(4), 771–780. https://doi.org/10.1016/J.JMSY.2013.04.015

Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly, 3*(1–2), 59–66. https://doi.org/10.1002/nav.3800030106

Subramaniam, V., & Raheja, A. S. (2003). mAOR: A heuristic-based reactive repair mechanism for job shop schedules. *International Journal of Advanced Manufacturing Technology, 22*(9–10), 669–680. https://doi.org/10.1007/S00170-003-1601-6/FIGURES/8

Suwa, H., & Sandoh, H. (2007). Capability of cumulative delay based reactive scheduling for job shops with machine breakdowns. *Computers & Industrial Engineering, 53*(1), 63–78. https://doi.org/10.1016/J.CIE.2007.04.002

Tamssaouet, K., Dauzère-Pérès, S., & Yugma, C. (2018). Metaheuristics for the job-shop scheduling problem with machine availability constraints. *Computers & Industrial Engineering, 125*, 1–8. https://doi.org/10.1016/j.cie.2018.08.008

Tjornfelt-Jensen, M., & Hansen, T. K. (1999). Robust solutions to job shop problems. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, pp. 1138–1144. https://doi.org/10.1109/CEC.1999.782551

Tunali, S., & Orhun, E. (1990). A knowledge-based production scheduling system. *Proceedings of the IEEE International Workshop on Intelligent Motion Control, IMC, 1990*(2), 577–580. https://doi.org/10.1109/IMC.1990.687384

Upasani, A., & Uzsoy, R. (2008). Integrating a decomposition procedure with problem reduction for factory scheduling with disruptions: a simulation study. *International Journal of Production Research, 46*(21), 5883–5905. https://doi.org/10.1080/00207540601156215

Vinod, V., & Sridharan, R. (2011). Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. *International Journal of Production Economics, 129*(1), 127–146.

Wu, Z., Sun, S., & Yu, S. (2020). Optimizing makespan and stability risks in job shop scheduling. *Computers & Operations Research, 122*, 104963. https://doi.org/10.1016/J.COR.2020.104963

Xie, J., Li, X., Gao, L., & Gui, L. (2023). A new neighbourhood structure for job shop scheduling problems. *International Journal of Production Research, 61*(7), 2147–2161. https://doi.org/10.1080/00207543.2022.2060772

Yahyaoui, A., Fnaiech, N., & Fnaiech, F. (2009). New shifting method for job shop scheduling subject to invariant constraints of resources availability. In *IECON Proceedings (Industrial Electronics Conference)*, pp. 3387–3392. https://doi.org/10.1109/IECON.2009.5415368

Yamada, T., & Nakano, R. (1992). A genetic algorithm applicable to large-scale job-shop problems. In *PPSN* (Vol. 2, pp. 281–290).

Yang, H. A., Lv, Y., Xia, C., Sun, S., & Wang, H. (2014). Optimal computing budget allocation for ordinal optimization in solving stochastic job shop scheduling problems. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2014/619254

Zandieh, M., & Adibi, M. A. (2010). Dynamic job shop scheduling using variable neighbourhood search. *International Journal of Production Research, 48*(8), 2449–2458. https://doi.org/10.1080/00207540802662896

Zhang, C., Li, P., Guan, Z., & Rao, Y. (2007). A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Computers & Operations Research, 34*(11), 3229–3242. https://doi.org/10.1016/j.cor.2005.12.002

Zhang, G., Lu, X., Liu, X., Zhang, L., Wei, S., & Zhang, W. (2022). An effective two-stage algorithm based on convolutional neural network for the bi-objective flexible job shop scheduling problem with machine breakdown. *Expert Systems with Applications, 203*, 117460. https://doi.org/10.1016/j.eswa.2022.117460

Zhang, J., Ding, G., Zou, Y., Qin, S., & Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing, 30*(4), 1809–1830. https://doi.org/10.1007/s10845-017-1350-2

Zhang, R., & Wu, C. (2011). An artificial bee colony algorithm for the job shop scheduling problem with random processing times. *Entropy, 13*(9), 1708–1729. https://doi.org/10.3390/e13091708

Zhao, F., Hong, Y., Yu, D., Yang, Y., & Zhang, Q. (2010). A hybrid particle swarm optimisation algorithm and fuzzy logic for process planning and production scheduling integration in holonic manufacturing systems. *International Journal of Computer Integrated Manufacturing, 23*(1), 20–39. https://doi.org/10.1080/09511920903207472

Zhu, L., & Soh, Y. C. (1999). FMS job-shop scheduling under disruptions with consideration of time and sequence deviation. In *IEEE International Symposium on Intelligent Control-Proceedings*, pp. 138–143. https://doi.org/10.1109/ISIC.1999.796644

Zou, Z., & Li, C. (2006). Integrated and events-oriented job shop scheduling. *International Journal of Advanced Manufacturing Technology, 29*(5), 551–556. https://doi.org/10.1007/s00170-005-2537-9

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