Multi-objective Optimization of Distributed Job Shop Scheduling Problems with Customer Time Requirements

Francisco dos Santos[1,2(✉)], Lino Costa[1,3], and Leonilde Varela[1,3]

[1] ALGORITMI Research Centre/LASI, University of Minho, Braga, Portugal
{lac,leonilde}@dps.uminho.pt
[2] Polytechnic Institute, Kimpa Vita University, Uíge, Angola
francisco_dos_santos@outlook.pt
[3] Department of Production and Systems, University of Minho, Braga, Portugal

Abstract. In the contemporary industrial context, characterized by intense competitive pressures, consumers have become increasingly discerning, imposing stringent criteria on suppliers regarding delivery times, quality standards, and costs. In recent years, in response to this growing demand and the search for greater operational efficiency, several organizations have chosen to fragment their production units, migrating to decentralized manufacturing models. In this context, production systems are characterized by the need to coordinate multiple operations that are processed on machines distributed across different manufacturing locations. The present paper proposes a multi-objective model for the job shop scheduling problem in a distributed context, incorporating uncorrelated parallel machines and setup times influenced by jobs ordering. The model incorporates a temporal dimension that accounts for the time required to transport raw materials to the manufacturing unit selected for each operation, as well as the deadlines set by customers. The integration of customer priorities, materialized in the simultaneous optimization of three indicators: total processing time (makespan), delivery delays (tardiness) and production advances (earliness), represents a novel innovation of this study. In order to validate the proposed methodology, NSGA-III was used to solve small instances of the problem. The preliminary findings substantiate the efficacy of the proposed strategy.

Keywords: Distributed job shop scheduling problem · multi-objective optimization · evolutionary algorithms

This work has been supported by FCT Fundação para a Ciência e Tecnologia within the R&D Unit Project Scope UID/00319/2025 - Centro ALGORITMI (ALGORITMI/UM) https://doi.org/10.54499/UID/00319/2025..

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16760, pp. 182–198, 2027.
https://doi.org/10.1007/978-3-032-30521-3_12

1 Introduction

In recent decades, the industrial sector has undergone significant transformations driven by market globalization and technological advances. This new economic paradigm is characterized by fierce competition, where organizations are constantly challenged to offer high-quality products, short delivery times, and competitive costs thanks to the use of job shop scheduling in production processes [13]. In this dynamic environment, adaptability and operational efficiency have become critical factors for survival and success in these industries [18].

At the same time, a structural change in the organization of production has been observed. Traditionally concentrated in single manufacturing units, which are generally traditional job shop scheduling problems (JSS) [15], manufacturing has evolved toward decentralized models, where companies distribute their operations across multiple geographic locations. This strategy, known as distributed production, not only increases production capacity, but also brings production closer to consumer markets, reduces logistics costs, and mitigates risks associated with localized contingencies. However, this decentralization introduces additional complexities in production planning and control, particularly with regard to the coordination of jobs between different manufacturing units.

In this context, the problem of distributed job shop scheduling problem (DJSS) arises [4,7,8], involving different optimization criteria such as costs, delivery time, delay, among others [6,17]. Unlike traditional job shop scheduling problems [10], where all machines are located in a single location, DJSS problems involve the allocation and sequencing of jobs on machines distributed across different factories [16], adding an additional layer of complexity related to the choice of production unit for each operation and the transport times involved [12]. The existing literature has devoted considerable attention to the optimization of classic performance indicators such as makespan (total completion time). However, in a customer-oriented market, other criteria are becoming increasingly relevant. Meeting delivery deadlines has become a key competitive advantage, involving not only minimizing delays (tardiness) but also reducing production advances (earliness). Early deliveries can generate storage and capital immobilization costs, while late deliveries compromise customer satisfaction and the company's reputation. The simultaneous consideration of these factors, aligned with customer preferences, represents an important gap in current research.

Incorporating customer preferences into this environment requires the simultaneous optimization of multiple objectives. In addition to minimizing makespan, it is essential to minimize tardiness, ensuring that deliveries occur within promised deadlines and earliness, avoiding the storage and capital immobilization costs associated with early production. The joint consideration of these three criteria, in a context of unrelated parallel machines, sequence-dependent setup times, and transport times, presents a highly complex problem, the solution of which requires sophisticated multi-objective approaches, such as Non-dominated Sorting Genetic Algorithm III (NSGA-III) [5,15].

This paper proposes a multi-objective model for the distributed job shop scheduling problem that incorporates unrelated parallel machines, sequence-

dependent setup times [1], raw material transport times, and customer order delivery times. As a main, the model integrates customer preferences such as meeting delivery deadlines and reducing warehousing costs through the simultaneous optimization of three objectives: minimization of makespan, minimization of tardiness, and minimization of earliness. To solve the problem, NSGA-III is used, a metaheuristic recognized for its effectiveness in addressing complex multi-objective problems.

This paper is organized as follows: in addition to the introduction, Sect. 2 presents a multi-objective optimization model, including an overview of the relevant literature on distributed job shop scheduling problems, as well as a mathematical formulation of the problem, and the optimization approach. Section 3 reports and analyzes the experimental results for three instances of the problem. Finally, Sect. 4 addresses the conclusions and suggestions for future work.

2 Multi-objective Optimization Model

2.1 Distributed Job Shop Scheduling

The traditional job shop scheduling problem is one of the most studied classes in the field of production optimization, recognized for its computational complexity and practical relevance. In its classic formulation, JSS problems involve a set of jobs that must be processed on a given number of machines, following a predefined sequence for each job [3,7,10]. Each operation requires a specific machine for a given period of time, and the central objective is to find the processing sequence that optimizes certain performance criteria. Traditionally, the most commonly used criterion has been the minimization of makespan, i.e., the total time required to complete all tasks. However, the evolution of production systems and market demands has broadened this spectrum, incorporating metrics related to meeting deadlines, energy efficiency, or load balancing between resources [14].

The transition to decentralized production models gave rise to the distributed job shop scheduling problem, an extension of JSS problems that reflects contemporary industrial reality. In this scenario, machines are not all concentrated in a single location, but rather distributed across multiple factories, geographically dispersed [16,17]. This configuration introduces two interdependent decisions: the assignment of each task to a manufacturing unit and the sequencing of operations within each factory. The complexity of the problem is further amplified by the need to consider transport times between different locations, as well as the possibility that each factory may have multiple parallel machines capable of performing the same operation with potentially different processing times (unrelated parallel machines). Additionally, the setup times of the machines can vary depending on the sequence of tasks processed, reflecting the need for adjustments, tool changes, or cleaning between consecutive operations [1,10].

In addition to the structural complexities of DJSS problems, customer preferences emerge as a determining factor in the definition of optimization objectives. In a service-oriented market, strict compliance with delivery deadlines has

become strategic importance. This reality translates into the need to minimize not only tardiness, which represents deliveries made after the date promised to the customer, but also earliness, which corresponds to early completion of production [17]. Although early delivery may seem benign at first glance, it entails significant costs related to storage, additional handling, and capital immobilization. The simultaneous optimization of makespan, tardiness, and earliness poses a complex multi-objective problem, where there are often trade-offs between the different criteria: for example, minimizing makespan can lead to solutions that compromise the fulfillment of individual deadlines. The integrated consideration of these objectives, aligned with customer priorities, represents a significant advance over traditional approaches focused exclusively on internal production efficiency.

2.2 Mathematical Formulation

In traditional job shop scheduling problems, there is a set of jobs that must be executed on a set of machines, following predefined processing sequences. The classic objective generally focuses on minimizing the total completion time of all jobs, known as the makespan. However, this traditional approach does not adequately address the requirements associated with meeting delivery deadlines, ignoring both the costs resulting from delays (tardiness) and the inefficiencies caused by early completion (earliness).

In distributed job shop scheduling problems, complexity increases significantly. In addition to the set of jobs that must be executed on a set of machines, these machines are dispersed across multiple factories located in different geographic regions. This dispersion introduces additional decisions related to the assignment of each job to a specific manufacturing unit. Consequently, in addition to the processing and setup times of the jobs on the machines, it is imperative to consider other factors, such as the transport times of raw materials between factories or to production units, as well as the implications of this geographic distribution on meeting deadlines agreed upon with customers. Thus, a DJSS problem involves the following sets:

- $\mathbb{N} = \{n_1, n_2, n_3, ...n_N\}$ with N jobs;
- $\mathbb{F} = \{f_1, f_2, f_3, ...f_F\}$ with F factories;
- $\mathbb{M}_1 = \{m_{1,1}, \ldots, m_{1,M_1}\}$, $\mathbb{M}_2 = \{m_{2,1}, \ldots, m_{2,M_2}\}$, ..., $\mathbb{M}_F = \{m_{F,1}, \ldots, m_{F,M_F}\}$ with $M_1, M_2, \ldots, M_F$ machines for each factory, respectively.

It is assumed that each machine in each factory is continuously available and can execute only one job at a time. Moreover, all jobs are independent, and none of them have precedence constraints. Preemption is not allowed; that is, the processing of any job on a machine cannot be interrupted. It also assumed that the following problem data is known:

- $T_{F \times N}$ matrix, where $t_{u,j}$ is the transportation time to transport job j to the factory u;

- $P_{F\times M\times N}$ matrix, where $p_{u,i,j}$ is the processing time of job j on machine i of factory u;
- $S_{F\times M\times N\times N}$ matrix, where $s_{u,i,j,k}$ is the setup time to process job j after job k on machine i of factory u ($s_{u,i,j,0}$ is the setup time to process job j at the initial instant on machine i of factory u);
- D_N vector, where d_j is the delivery date of the job j.

The goal is to find a feasible sequence of jobs on each machine in different factories that minimizes the makespan [12,17], the tardiness time and the earliness [17]. The feasible sequence π assigns all jobs to machines in each factory, i.e., it is the vector of all feasible subsequences $\pi_{u,i}$ in each machine i and factory u. For a given sequence π, the makespan ($c_{\max}(\pi)$) can be expressed as follows:

$$c_{\max}(\pi) = \max_{u=1,\ldots,F} \Big(\max_{i=1,\ldots,M_u} (c_{u,i}(\pi)) \Big) \tag{1}$$

where $c_{u,i}(\pi)$ is the total completion time of the jobs on machine i of the factory u for the feasible sequence π. This value can be computed by

$$c_{u,i}(\pi) = \sum_{j\in\pi_{u,i}} t_{u,j} + \sum_{j\in\pi_{u,i}} p_{u,i,j} + \sum_{j\in\pi_{u,i}} s_{u,i,j,k}$$

with k being the jobs that precede jobs j in the subsequence $\pi_{u,i}$ of the feasible sequence π. It should be noted that when the job j is the first job in the subsequence $\pi_{u,i}$, k is zero, and therefore $s_{u,i,j,0}$ is the initial setup time.

The tardiness time ($t_{\max}(\pi)$) for a given sequence π is given by

$$t_{\max}(\pi) = \max_{u=1,\ldots,F} \Big(\max_{i=1,\ldots,M_u} (t_{u,i}(\pi)) \Big) \tag{2}$$

where $t_{u,i}(\pi) = \max_{j\in\pi_{u,i}} (0, c^j_{u,i}(\pi) - d_j)$, and

$$c^j_{u,i}(\pi) = \sum_{\forall z\preceq j\wedge z,j\in\pi_{u,i}} t_{u,j} + \sum_{\forall z\preceq j\wedge z,j\in\pi_{u,i}} p_{u,i,z} + \sum_{\forall z\preceq j\wedge z,j\in\pi_{u,i}} s_{u,i,z,k}$$

is the completion time of the job j, with z being all jobs that precede or are equal to the job j in the subsequence $\pi_{u,i}$ of the feasible sequence π.

The earliness time ($l_{\max}(\pi)$) for a given sequence π is defined by

$$l_{\max}(\pi) = \max_{u=1,\ldots,F} \Big(\max_{i=1,\ldots,M_u} (l_{u,i}(\pi)) \Big) \tag{3}$$

where $l_{u,i}(\pi) = \max_{j\in\pi_{u,i}} (0, d_j - c^j_{u,i}(\pi))$.

In this paper, two bi-objective DJSS problems are considered:

$$\min \mathbf{F_1}(\pi) = \big(c_{\max}(\pi), t_{\max}(\pi)\big)^T, \tag{4}$$

and

$$\min \mathbf{F_2}(\pi) = \big(c_{\max}(\pi), l_{\max}(\pi)\big)^T. \tag{5}$$

Additionally, the following multi-objective DJSS problem with three objectives is also addressed:

$$\min \mathbf{F_3}(\pi) = \big(c_{\max}(\pi), t_{\max}(\pi), l_{\max}(\pi)\big)^T. \tag{6}$$

In these formulations, π is a feasible sequence that belongs to the feasible set Ω (the set of all possible permutations) and $\mathbf{F_1}(\pi)$, $\mathbf{F_2}(\pi)$, and $\mathbf{F_3}(\pi)$ are the vectors of objective functions defined in the objective space.

2.3 Optimization Approach

DJSS problems with customer time requirements involve several conflicting objectives such as makespan, tardiness and earliness expressed by equations (1), (2) and (3). Multi-objective optimization algorithms aim to find optimal trade-offs between objectives, the so-called Pareto-optimal set. The images of the Pareto-optimal solutions define a Pareto optimal frontier in the objective space. This frontier allows the decision-maker to select a solution taking into account the several optimal alternatives that best fulfill his preferences [9].

Evolutionary algorithms draw inspiration from the genetic mechanisms of living organisms, which evolve over time through natural selection. In particular, genetic algorithms are widely applied to multi-objective problems because their working principles are based on the evolution of a population of chromosomes representing candidate solutions and the application of genetic operators to generate new ones. Therefore, it is possible to approximate the Pareto-optimal set effectively in a single run.

Furthermore, DJSS problems are also highly complex combinatorial problems. Certain features of evolutionary algorithms make them particularly suitable for addressing such problems, since constraints can be managed through permutation-based chromosome representations, and genetic operators can be specifically tailored to these search spaces. This enables the algorithm to efficiently identify feasible solutions.

NSGA-III [5], an evolutionary algorithm that can tackle multi-objective optimization problems, was adopted to solve DJSS problems with customer time requirements since it has good performance even for problems with more than two objectives [11,12,15].

Each chromosome in the population is represented as a permutation of length $N + \sum_{u=1}^{F} M_u - 1$ genes, encoding a feasible sequence of jobs in machines in different factories (π). Within the chromosome, genes with values greater than N act as delimiters that partition the permutation into subsequences ($\pi_{u,i}$). Each subsequence specifies the set of jobs, together with their processing order, assigned to machine $i \in \mathbb{M}_u$ in the factory $u \in \mathbb{F}$.

To select chromosomes for reproduction, a binary tournament selection operator was used, allowing offspring to be generated from parent solutions through the application of genetic operators. To ensure solution feasibility in the search process, order-based crossover and inversion mutation were adopted. In order-based crossover, genes from the two parent chromosomes are combined by first

selecting two random positions and swapping the genes located between them. The remaining empty positions are then filled with the genes from the other parent while avoiding duplicates. In the inversion mutation, two positions are randomly chosen, and the sequence of genes between them is reversed.

As result of the optimization process, an approximation to the Pareto-optimal set is obtained. Each solution in this set is a different trade-off in terms of the criteria. Various tools can be used to present and visualize a solution, such as Gantt charts to display job scheduling, graph-based representations of the Pareto-optimal frontier, or parallel coordinate plots to highlight trade-offs among objectives. These visualization techniques are intended to support a clearer analysis of solutions to assist in the decision-making process.

3 Experimental Results

3.1 General Setup

In order to test the approach to solve distributed job shop scheduling problems with customer time requirements, three instances (P_1, P_2, and P_3) are borrowed from a previous work [16]. The P_1 instance has 6 jobs ($N = 6$) to be assigned to machines distributed by 2 factories ($F = 2$). Each factory has 2 machines ($M_1 = 2$ and $M_2 = 2$). For this instance, the delivery deadlines for the jobs, reflecting customer time requirements, are $d_1 = \big(200, 230, 200, 260, 285, 235\big)^T$.

Instances P_2 and P_3 both have 11 jobs ($N = 11$). The P_2 instance has 2 factories ($F = 2$) but with different numbers of machines, one with 2 machines and the other with 4 machines ($M_1 = 2$ and $M_2 = 4$). Finally, instance P_3 has 3 factories ($F = 3$), two of them with 2 machines, and the other with 4 machines ($M_1 = 2$, $M_2 = 2$ and $M_3 = 4$). Additionally, the delivery deadlines for the jobs for instances P_2 and P_3 are $d_2 = \big(200, 230, 200, 260, 285, 235, 205, 290, 246, 278, 201\big)^T$ and $d_3 = \big(199, 200, 230, 296, 200, 201, 260, 285, 235, 188, 268\big)^T$, respectively. Processing times p, setup times s, and transportation times t are given in a previous study [16].

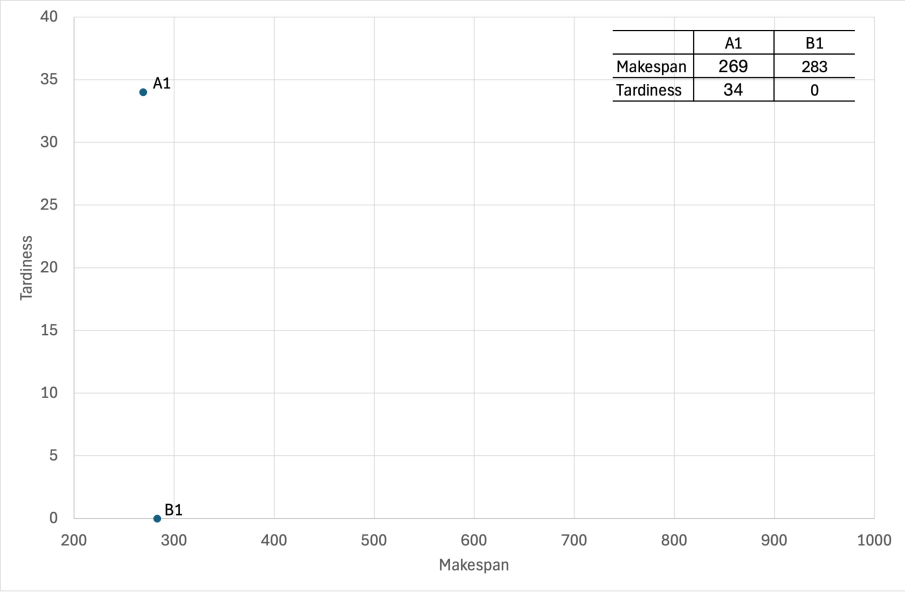
In this study, NSGA-III was used to assess the proposed multi-objective DJSS with the customer time requirements approach. All instances were solved considering the DJSS problems $\mathbf{F_1}(\pi)$, $\mathbf{F_2}(\pi)$, and $\mathbf{F_3}(\pi)$ expressed by the equations (4), (5), and (6), respectively. The implementation was carried out in Python using the PyMOO framework [2]. The population size was set to 100 chromosomes for bi-objective problems $\mathbf{F_1}(\pi)$ and $\mathbf{F_2}(\pi)$, and 200 chromosomes for problem $\mathbf{F_3}(\pi)$ with 3 objectives. The stopping criteria combined a maximum generation limit with a stagnation threshold: the algorithm terminated after reaching 10,000 generations or after 100 consecutive generations without improvement. In addition, a limit of 500,000 objective function evaluations was imposed.

Due to the stochastic nature of evolutionary algorithms, 20 independent runs were conducted for each instance and problem. In the following, with respect to

each instance, the results for the run with greater hypervolume (among the 20 independent runs) are reported. Note that the non-dominated solutions obtained in the run with the highest hypervolume value correspond to the best approximation to the Pareto optimal set. The reference point for hypervolume computation was $(2000, 2000)^T$ for bi-objective problems ($\mathbf{F_1}(\pi)$ and $\mathbf{F_2}(\pi)$). For problem $\mathbf{F_3}(\pi)$ with 3 objectives, the reference point $(2000, 2000, 2000)^T$ was considered.

3.2 P_1 Instance Analysis

In Fig. 1, for instance P_1, the approximation to the Pareto optimal front obtained when solving problem $\mathbf{F_1}(\pi)$, the minimization of makespan and tardiness, is represented. This Pareto front intends to facilitate the decision-maker process, i.e., to select the most convenient solution according to his/her preferences.



	A1	B1
Makespan	269	283
Tardiness	34	0

Fig. 1. Representation of Pareto optimal front for the minimization of makespan and tardiness ($\mathbf{F_1}(\pi)$ problem) for the instance P_1.

For this instance, in all 20 independent runs of NSGA-III, the same two non-dominated solutions (solutions A1 and B1) were obtained. This figure also includes the pay-off table with the values of the objectives for each solution. Solution A1 is better than solution B1 in terms of makespan, but worst in terms of tardiness. Actually, solution A1 has a tardiness value superior to zero, which means that customer delivery dates for the jobs are not fulfilled completely. On

the other hand, solution B1 satisfies all delivery dates with a slightly higher value of makespan. So, it is worth taking a deeper analysis of these solutions using Gantt diagrams to represent the corresponding scheduling of jobs.

For illustration purposes, Fig. 2 and Fig. 3 are Gantt diagrams, representing the scheduling of jobs corresponding to solutions A1 and B1, for instance P_1 with 2 factories, both with 2 machines ($m_{1,1}$, $m_{1,2}$, $m_{2,1}$ and $m_{2,2}$) and 6 jobs (n_1, n_2, n_3, n_4, n_5 and n_6). In these figures, the transportation times, setup times, and processing times of jobs are represented by gray, blue, and orange boxes. The customer delivery dates for jobs (d_1, d_2, d_3, d_4, d_5 and d_6) are also indicated in the diagram.

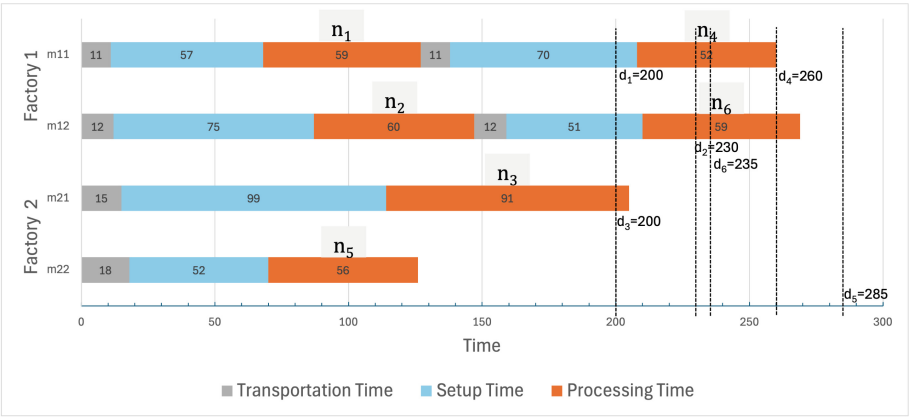


Fig. 2. Gantt diagram representing solution A1 for problem $\mathbf{F_1}(\pi)$ for instance P_1.

The Gantt diagram (Fig. 2) represents the solution A1 and corresponds to the sequence $\pi =< \pi_{1,1}, \pi_{1,2}, \pi_{2,1}, \pi_{2,2} >$ where $\pi_{1,1} = [1, 4]$, $\pi_{1,2} = [2, 6]$, $\pi_{2,1} = [3]$ and $\pi_{2,2} = [5]$. The total completion times are $c_{1,1}(\pi) = 260$, $c_{1,2}(\pi) = 269$, $c_{2,1}(\pi) = 205$ and $c_{2,2}(\pi) = 126$, and therefore $c_{\max}(\pi) = 269$. Since $c^1_{1,1}(\pi) = 127$, $c^4_{1,1}(\pi) = 260$, $c^2_{1,2}(\pi) = 147$, $c^6_{1,2}(\pi) = 269$, $c^3_{2,1}(\pi) = 205$ and $c^5_{2,2}(\pi) = 126$, then $t_{1,1}(\pi) = 0$, $t_{1,2}(\pi) = 34$, $t_{2,1}(\pi) = 5$, and $t_{2,2}(\pi) = 0$ and therefore $t_{\max}(\pi) = 34$. Note that, despite this solution having the same makespan, it is different from solution A previously published in [16] since the order of jobs in machine 1 of factory 1 (job n_1 before job n_4) is the opposite due to the existence of customer delivery dates.

Solution B1 is represented in the Gantt diagram (Fig. 3). This solution corresponds to the sequence $\pi =< \pi_{1,1}, \pi_{1,2}, \pi_{2,1}, \pi_{2,2} >$ where $\pi_{1,1} = [1, 4]$, $\pi_{1,2} = [3]$, $\pi_{2,1} = [2]$ and $\pi_{2,2} = [6, 5]$. The total completion times are $c_{1,1}(\pi) = 260$, $c_{1,2}(\pi) = 164$, $c_{2,1}(\pi) = 168$ and $c_{2,2}(\pi) = 283$, and therefore $c_{\max}(\pi) = 283$. Since $c^1_{1,1}(\pi) = 127$, $c^4_{1,1}(\pi) = 260$, $c^3_{1,2}(\pi) = 164$, $c^2_{2,1}(\pi) = 168$, $c^6_{2,2}(\pi) = 158$ and $c^5_{2,2}(\pi) = 283$, then $t_{1,1}(\pi) = 0$, $t_{1,2}(\pi) = 0$, $t_{2,1}(\pi) = 0$, and $t_{2,2}(\pi) = 0$ and

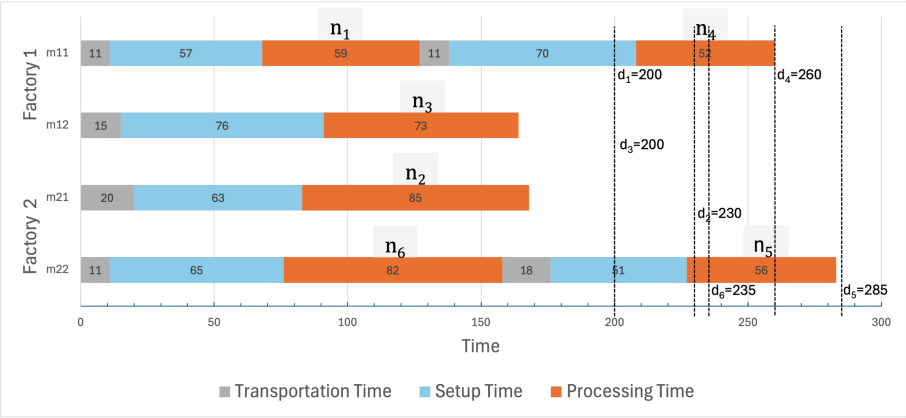


Fig. 3. Gantt diagram representing solution B1 for problem $\mathbf{F_1}(\pi)$ for instance P_1.

therefore $t_{\max}(\pi) = 0$. This solution satisfies all delivery dates for the customer. Thus, it is a good solution from the point of view of the customer's requirements.

In Fig. 4, for instance P_1, the approximation to the Pareto optimal front obtained when solving problem $\mathbf{F_2}(\pi)$, the minimization of makespan and earliness, is represented. Four non-dominated solutions (A2, B2, C2 and D2) were found. Solutions A2 and D2 are extreme solutions located at the upper-left and lower-right extremes of the Pareto front, respectively.

Although solution A2 has the same makespan value as solution A1 of problem $\mathbf{F_1}(\pi)$, it corresponds to a different scheduling. Figure 5 shows the scheduling of jobs corresponding to the sequence $\pi =< \pi_{1,1}, \pi_{1,2}, \pi_{2,1}, \pi_{2,2} >$ where $\pi_{1,1} = [1, 4]$, $\pi_{1,2} = [2, 6]$, $\pi_{2,1} = [5]$ and $\pi_{2,2} = [3]$. This solution is very similar to solution A1, but with jobs n_3 and n_5 assigned to different machines in factory 2 (Fig. 5).

The total completion times for solution A2 are $c_{1,1}(\pi) = 260$, $c_{1,2}(\pi) = 269$, $c_{2,1}(\pi) = 185$, and $c_{2,2}(\pi) = 174$, and therefore $c_{\max}(\pi) = 269$. Since $c^1_{1,1}(\pi) = 127$, $c^4_{1,1}(\pi) = 260$, $c^2_{1,2}(\pi) = 147$, $c^6_{1,2}(\pi) = 269$, $c^5_{2,1}(\pi) = 185$ and $c^3_{2,2}(\pi) = 174$, then $l_{1,1}(\pi) = 73$, $l_{1,2}(\pi) = 83$, $l_{2,1}(\pi) = 100$, and $l_{2,2}(\pi) = 26$ and therefore $l_{\max}(\pi) = 100$. Note that this solution A2 has the same tardiness value ($t_{\max}(\pi) = 34$) as that of solution A1. However, the earliness of solution A1 is higher ($l_{\max}(\pi) = 159$). Therefore, solution A1 is dominated by solution A2 when the 3 objectives are considered simultaneously.

Solutions B2 and C2 are located at the elbow of the Pareto front and can be good trade-offs between both objectives. Its earliness values do not differ significantly. At the other extreme of the Pareto front, solution D2 has zero earliness, which means that all jobs terminate exactly at the customer delivery date, but with a high makespan. The sequence $\pi =< \pi_{1,1}, \pi_{1,2}, \pi_{2,1}, \pi_{2,2} >$ for solution D2 is $\pi_{1,1} = [\,]$, $\pi_{1,2} = [\,]$, $\pi_{2,1} = [3, 5, 2, 1, 6, 4]$ and $\pi_{2,2} = [\,]$ with a

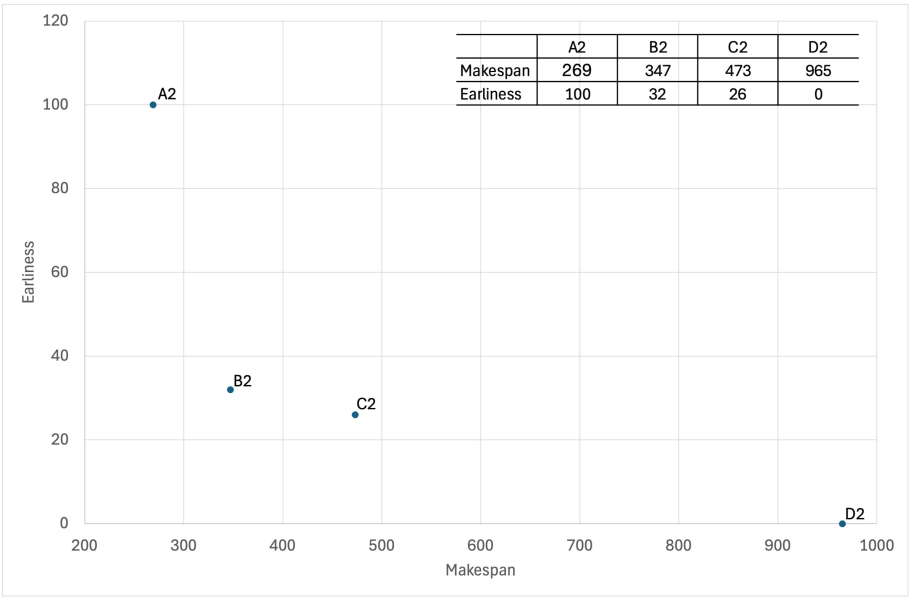


Fig. 4. Representation of Pareto optimal front for the minimization of makespan and earliness ($\mathbf{F_2}(\pi)$ problem) for the instance P_1.

makespan of $c_{\max}(\pi) = 965$. In this solution D2, all jobs are assigned to machine 1 of factory 2. All other machines are not occupied.

The results of the problems $\mathbf{F_1}(\pi)$ and $\mathbf{F_2}(\pi)$ highlight the importance of simultaneously considering makespan, tardiness, and earliness. So, in Fig. 6, the approximation to the Pareto optimal front obtained for the problem $\mathbf{F_3}(\pi)$ is shown. In this figure, makespan, tardiness, and earliness are represented on the X-axis, the Y-axis, and by the size of the points. It can be observed that a larger number of non-dominated solutions was found compared to the bi-objective formulations ($\mathbf{F_1}(\pi)$ and $\mathbf{F_2}(\pi)$). As stated above, the solution A3 is equal to A2, but different from A1. On the other hand, solutions B3, F3, G3 and H3 are equal to solutions B1, B2, C2 and D2, respectively.

From Fig. 6, it seems that, in general, makespan and earliness are negatively associated, i.e., when makespan increases, earliness tends to decrease. This behavior does not occur with tardiness. These results confirm the conflicting nature of these three objectives.

3.3 P_2 and P_3 Instances Analysis

The problems $\mathbf{F_1}(\pi)$, $\mathbf{F_2}(\pi)$, and $\mathbf{F_3}(\pi)$ were solved for instances P_2 and P_3. Since all nom-dominated solutions obtained for $\mathbf{F_1}(\pi)$ and $\mathbf{F_2}(\pi)$ were also obtained when solving $\mathbf{F_3}(\pi)$, in this paper, only the results for the latter problem are presented.

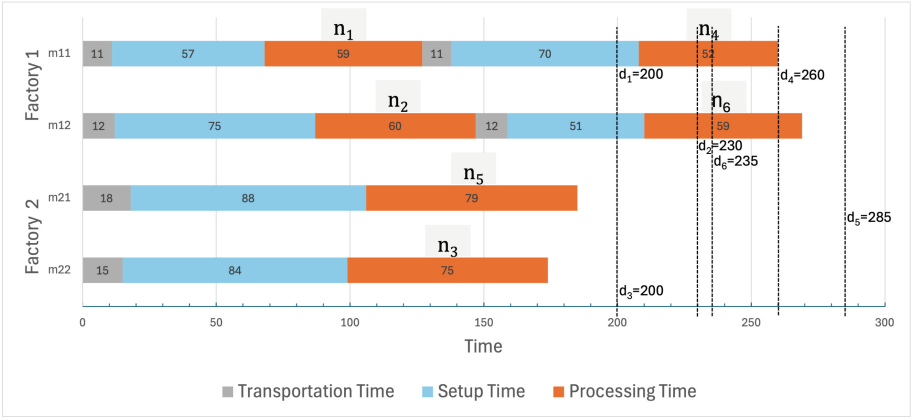


Fig. 5. Gantt diagram representing solution A2 for problem $\mathbf{F_2}(\pi)$ for instance P_1.

In Fig. 7, the approximation to the Pareto optimal front obtained for the minimization of the makespan, tardiness and earliness for the instance P_2 is shown. For this instance, 11 non-dominated solutions were found (solutions A to K). Solutions A, B and K are the solutions with extreme values of the objectives. It should be noted that there is no solution with a tardiness equal to zero, i.e., the customer delivery dates cannot be all satisfied simultaneously. The solution with the smallest tardiness has a delay of 24. No solution was found with earliness equal to zero. However, solution K has an earliness value very small (equal to one). These solutions A, B and K correspond to the scheduling plans defined by the following sequences $\pi =< \pi_{1,1}, \pi_{1,2}, \pi_{2,1}, \pi_{2,2}, \pi_{2,3}, \pi_{2,4} >$:

- Solution A: $\pi_{1,1} = [1, 10]$, $\pi_{1,2} = [7]$, $\pi_{2,1} = [4, 8]$, $\pi_{2,2} = [3, 2]$, $\pi_{2,3} = [11, 9]$, and $\pi_{2,4} = [5, 6]$;
- Solution B: $\pi_{1,1} = [2, 10, 5]$, $\pi_{1,2} = [11, 6, 8]$, $\pi_{2,1} = [3, 4, 1]$, $\pi_{2,2} = [\,]$, $\pi_{2,3} = [7, 9]$, and $\pi_{2,4} = [\,]$;
- Solution K: $\pi_{1,1} = [\,]$, $\pi_{1,2} = [\,]$, $\pi_{2,1} = [\,]$, $\pi_{2,2} = [\,]$, $\pi_{2,3} = [3, 1, 9, 2, 7, 8, 11, 10, 5, 6, 4]$, and $\pi_{2,4} = [\,]$.

Solution A has the same makespan as solution A previously reported in [16] for the same instance P_2. Despite sharing some similarities, they have different jobs assigned to machines in factory 1. Other alternative solutions representing different trade-offs between objectives were also found.

The number of non-dominated solutions is larger due to the combinatorial nature of the problem and the conflicting objectives. We are reporting the approximation to the Pareto optimal front obtained from the run with the best hypervolume. However, it should be noted that taking into account all approximations to the Pareto optimal front obtained on the 20 runs, 36 non-dominated solutions were identified for the instance P_2. This suggests the existence of a higher number of solutions in the Pareto optimal front.

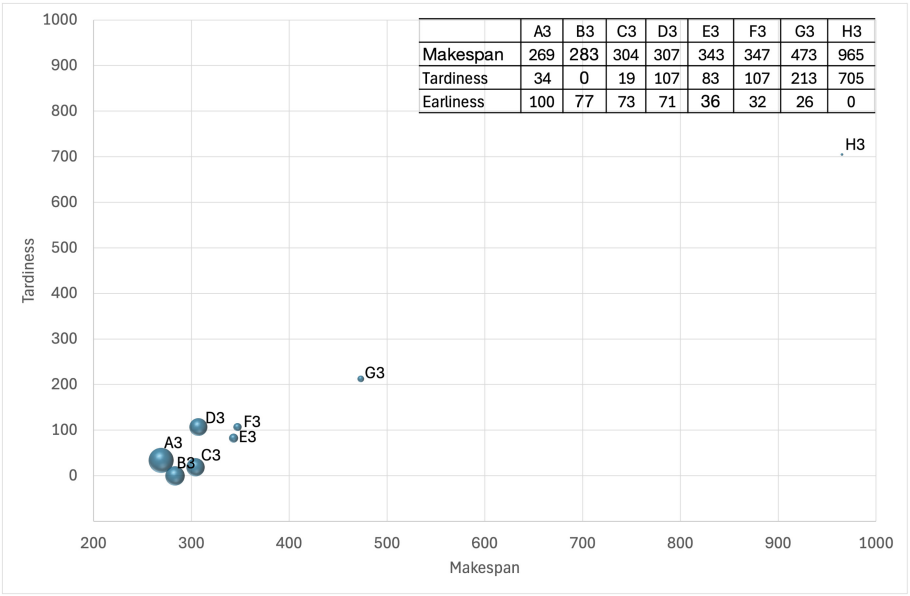


Fig. 6. Representation of Pareto optimal front for minimization of makespan, tardiness and earliness ($\mathbf{F_3}(\pi)$ problem) for the instance P_1.

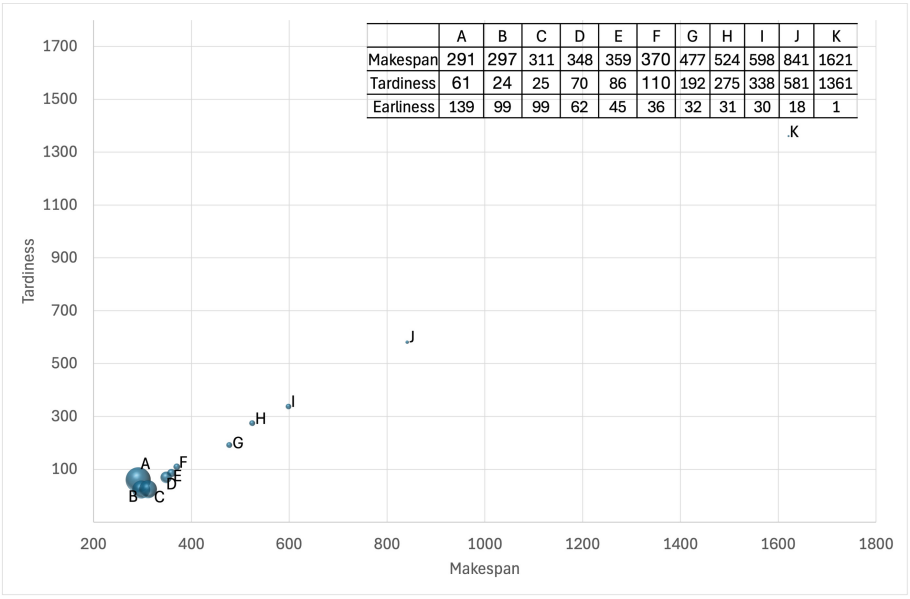


Fig. 7. Representation of Pareto optimal front for minimization of makespan, tardiness and earliness ($\mathbf{F_3}(\pi)$ problem) for the instance P_2.

Figure 8 depicts the approximation to the Pareto optimal front obtained for the minimization of the makespan, tardiness and earliness for the instance P_3. For this instance, the approximation to the Pareto optimal front is constituted by 16 non-dominated solutions (solutions A to P). Solutions A, D and P are the solutions with extreme values of the objectives. For this instance, there exists one solution with tardiness equal to zero (solution D) and another one with earliness equal to zero (solution P). The scheduling plans for these solutions are defined by the sequences $\pi =< \pi_{1,1}, \pi_{1,2}, \pi_{2,1}, \pi_{2,2}, \pi_{3,1}, \pi_{3,2}, \pi_{3,3}, \pi_{3,4} >$:

- Solution A: $\pi_{1,1} = [7,3]$, $\pi_{1,2} = [8]$, $\pi_{2,1} = [11]$, $\pi_{2,2} = [2]$, $\pi_{3,1} = [1]$, $\pi_{3,2} = [5,4]$, $\pi_{3,3} = [6]$, and $\pi_{3,4} = [10]$;
- Solution D: $\pi_{1,1} = [6]$, $\pi_{1,2} = [1,8]$, $\pi_{2,1} = [9,11]$, $\pi_{2,2} = [2]$, $\pi_{3,1} = [3]$, $\pi_{3,2} = [5]$, $\pi_{3,3} = [10,4]$, and $\pi_{3,4} = [7]$;
- Solution P: $\pi_{1,1} = [\,]$, $\pi_{1,2} = [10,4,3,11]$, $\pi_{2,1} = [\,]$, $\pi_{2,2} = [\,]$, $\pi_{3,1} = [2,5,9]$, $\pi_{3,2} = [\,]$, $\pi_{3,3} = [\,]$, and $\pi_{3,4} = [6,1,7,8]$.

In Fig. 8, the approximation to the Pareto optimal front obtained from the run with the best hypervolume is depicted. However, taking into account all approximations to the Pareto optimal front obtained in the 20 runs, 34 non-dominated solutions were identified for this instance P_3. Therefore, as for instance P_2, it seems that, due to the combinatorial nature of the search space, the Pareto optimal set has a higher cardinality.

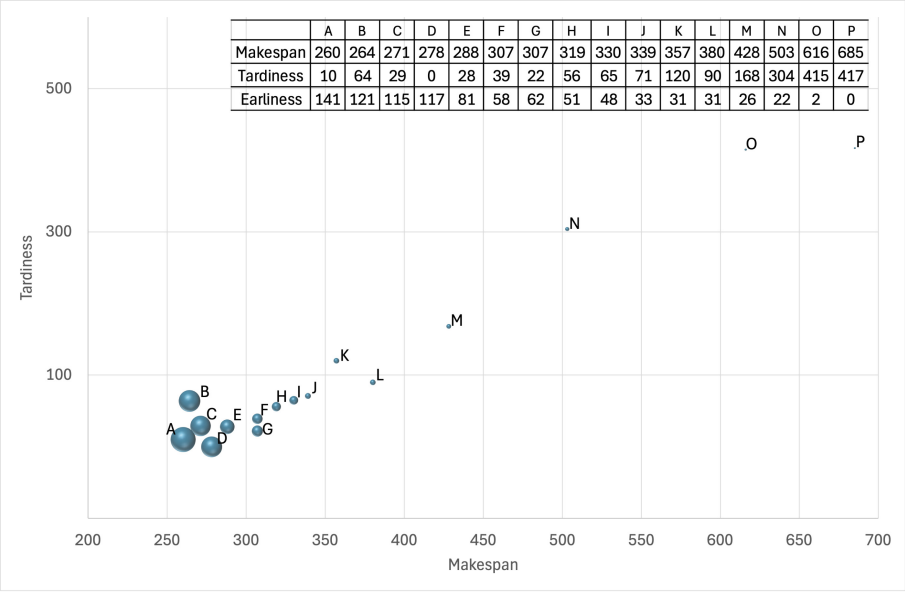


Fig. 8. Representation of Pareto optimal front for minimization of makespan, tardiness and earliness ($\mathbf{F_3}(\pi)$ problem) for the instance P_3.

From Fig. 7 and Fig. 8, it can observed again the existence of a tendency similar to that identified for the instance P_1, i.e., in general, makespan and earliness are negatively associated. However, this tendency does not exist with tardiness.

These results suggest that this approach to solving DJSS problems with customer delivery date requirements provides the decision-maker with relevant information about scheduling plans from both production and customer satisfaction perspectives.

4 Conclusions and Future Work

This paper proposes a multi-objective optimization approach for solving distributed job shop scheduling (DJSS) problems in an environment with unrelated parallel machines and sequence-dependent setup times. The main contribution of this study lies in the incorporation of customer preferences into the model, achieved through the simultaneous optimization of three fundamental objectives: minimization of makespan ($c_{\max}$), minimization of tardiness ($t_{\max}$) and minimization of earliness ($l_{\max}$). To this effect, the NSGA-III was employed, recognized for its effectiveness in addressing multi-objective problems. To this end, three test instances (P_1, P_2, and P_3) were considered, with different dimensions and configurations adapted from the previous work incorporating customer delivery dates for jobs.

In instance P_1, it was found that a diverse set of non-dominated solutions could be identified for the two-objective problems F_1 and F_2 , as well as solutions for the three-objective problem F_3. In instances P_2 and P_3, approximations to the Pareto optimal solutions were obtained. However, the results suggest that these Pareto optimal sets include a large number of solutions due to the combinatorial nature of the problems.

The graphical representation of the obtained approximations to the Pareto optimal fronts and the analysis of Gantt charts for specific solutions revealed the trade-offs inherent in the different criteria. It was observed that minimizing the makespan may lead to missed deadlines (tardiness), while strict compliance with delivery dates may require an increase in total production time. On the other hand, minimizing earliness can help reduce costs associated with customer inventory. These results demonstrate the usefulness of the proposed approach in supporting the decision-making process, providing the decision-maker with a set of optimal alternatives that reflect different trade-offs between production efficiency and the level of customer service.

Future work intends to extend the study to larger and more complex systems, as well as to explore the incorporation of additional objectives, such as reducing the number of idle machines or considering environmental sustainability criteria. Finally, comparing the performance of NSGA-III with other multi-objective metaheuristics is also a relevant line of research for future developments.

References

1. Arnaout, J.P., Musa, R., Rabadi, G.: A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines—Part II: enhancements and experimentations. J. Intell. Manuf. **25**, 43–53 (2014)
2. Blank, J., Deb, K.: PYMOO: multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020)
3. Chan, F.T., Chung, S.H., Chan, L., Finke, G., Tiwari, M.: Solving distributed FMS scheduling problems subject to maintenance: genetic algorithms approach. Robot. Comput.-Integr. Manuf. **22**(5–6), 493–504 (2006)
4. De Giovanni, L., Pezzella, F.: An improved genetic algorithm for the distributed and flexible job-shop scheduling problem. Eur. J. Oper. Res. **200**(2), 395–408 (2010)
5. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
6. Khan, B., Hanoun, S., Johnstone, M., Lim, C.P., Creighton, D., Nahavandi, S.: Multi-objective job shop scheduling using I-NSGA-III. In: 2018 Annual IEEE International Systems Conference (SysCon), pp. 1–5. IEEE (2018)
7. Lohmer, J., Lasch, R.: Production planning and scheduling in multi-factory production networks: a systematic literature review. Int. J. Prod. Res. **59**(7), 2028–2054 (2021)
8. Marzouki, B., Driss, O.B., Ghédira, K.: Solving distributed and flexible job shop scheduling problem using a chemical reaction optimization metaheuristic. Procedia Comput. Sci. **126**, 1424–1433 (2018)
9. Santos, F., Costa, L.: Multivariate analysis to assist decision-making in many-objective engineering optimization problems. In: Gervasi, O., et al. (eds.) ICCSA 2020. LNCS, vol. 12251, pp. 274–288. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58808-3_21
10. dos Santos, F., Costa, L., Varela, L.: Multi-objective optimization of the job shop scheduling problem on unrelated parallel machines with sequence-dependent setup times. In: Gervasi, O., et al. (eds.) ICCSA 2023. LNCS, vol. 14105, pp. 495–507. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37108-0_32
11. dos Santos, F., Costa, L., Varela, L.: Multi-objective scheduling optimization in job shop with unrelated parallel machines using NSGA-III. In: Gervasi, O., Murgante, B., Garau, C., Taniar, D., C. Rocha, A.M.A., Faginas Lago, M.N. (eds.) ICCSA 2024. LNCS, vol. 14816, pp. 370–382. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-65223-3_25
12. dos Santos, F., Costa, L., Varela, L.: Analysis of the effect of global and local objectives in multi-objective distributed job shop scheduling problems. In: Gervasi, O., et al. (eds.) ICCSA 2025. LNCS, vol. 15888, pp. 180–193. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-97596-7_12
13. dos Santos, F., Costa, L.A., Varela, L.: A systematic literature review about multi-objective optimization for distributed manufacturing scheduling in the industry 4.0. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A.C., Garau, C. (eds.) ICCSA 2022, Part II. LNCS, vol. 13378, pp. 157–173. Springer (2022). https://doi.org/10.1007/978-3-031-10562-3_12
14. dos Santos, F., Costa, L.A., Varela, L.: Multiobjective optimization in distributed industry and environmental sustainability: a systematic literature review. Revista Angolana de Ciências **5**(2), e050210 (2024)

15. dos Santos, F., Costa, L.A., Varela, L.: Performance comparison of NSGA-II and NSGA-III on bi-objective job shop scheduling problems. In: Pereira, A.I., Mendes, A., Fernandes, F.P., Pacheco, M.F., Coelho, J.P., Lima, J. (eds.) OL2A 2023. LNCS, vol. 1981, pp. 531–543. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-53025-8_36
16. Santos, F.d., Costa, L., Varela, L.: A multi-objective approach for solving distributed job shop scheduling problems. In: Pereira, A.I., et al. (eds.) OL2A 2024. . LNCS, vol. 2280, pp. 326–339. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-77426-3_22
17. Yenisey, M.M., Yagmahan, B.: Multi-objective permutation flow shop scheduling problem: literature review, classification and current trends. Omega **45**, 119–135 (2014)
18. Zhang, J., Ding, G., Zou, Y., Qin, S., Fu, J.: Review of job shop scheduling research and its new perspectives under industry 4.0. J. Intell. Manuf. **30**, 1809–1830 (2019)