

Analysis of the Effect of Global and Local Objectives in Multi-objective Distributed Job Shop Scheduling Problems

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

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

Abstract. In distributed manufacturing systems, the goal is to process jobs on several factories with different number of machines. Each machine has given processing and setup times that depend on the jobs. The problem of seeking solutions minimizing total production time (makespan), reducing tardiness, or maximizing resources utilization is known as Job Shop Scheduling problem. However, in this problem, each factory may have specific priorities or local objectives. On the other hand, global objectives of the organization can also exist. Thus, this paper studies the effect of these factory local objectives in Distributed Job Shop Scheduling Problems (DJSS). Some instances of the DJSS problems are solved using NSGA-III considering various combinations of local objectives with global ones in a multi-objective context. The trade-off solutions obtained are compared and analyzed. These preliminary results indicate the usefulness and validity of this approach.

Keywords: Distributed job shop scheduling problem · global and local objectives · multi-objective optimization · evolutionary algorithms

1 Introduction

The growing complexity of distributed manufacturing systems has revealed the need for efficient approaches to optimizing multiple objectives in distributed factory environments [13]. In the context of distributed job shop scheduling problems, each factory may have different priorities, such as minimizing total production time (makespan), reducing tardiness or maximizing resource utilization, among others. The existence of divergent objectives between factories introduces significant challenges, since local decisions can have a negative impact on the overall performance of the system. In this scenario, coordination between factories is crucial to ensure that individual goals are achieved without compromising collective efficiency.

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

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15888, pp. 180–193, 2026.
https://doi.org/10.1007/978-3-031-97596-7_12

Currently, distributed production systems have been widely adopted in industrial production processes, and in recent years many studies have been carried out on the modeling and optimization of distributed job shop scheduling problems [6]. The Distributed Job Shop Scheduling (DJSS) problem extends the traditional Job Shop Scheduling (JSS) problem by introducing a set of geographically distributed factories, each equipped with a specific number of machines. In this scenario, a certain number of jobs must be processed in these factories. DJSS problems are more complex and challenging than traditional JSS problems, as they require the simultaneous resolution of two interconnected sub-problems: assigning jobs to machines within each factory and determining the optimal sequence of jobs in each factory [3,8]. Like traditional JSS problems, the main objective of DJSS is to find a job distribution and sequencing strategy that minimizes the makespan [15], which remains the most widely used optimization criterion in JSS problems.

Previous works considered objectives such as makespan [1,11,15], the average completion time of the jobs [7,11,14], standard deviation of completion times and the total tardiness [9,15] among others. Some of these works considered DJSS as a single objective optimization problem. However, DJSS problems involve more than one objective to optimize simultaneously. This kind of problems require advanced optimization techniques [12,14], capable of dealing with multiple conflicting objectives [10]. In [11], a multi-objective distributed job shop scheduling model with unrelated parallel machines and sequence-dependent setup times was used to solve small instances using NSGA-III [4], considering the makespan and the average completion time as objectives in a global perspective.

However, it is important to study the effects of considering priorities or local objectives for factories. It should be stressed the relevance of considering the heterogeneity of factories in distributed production systems and therefore, in finding solutions that efficiently balance local and global requirements. Thus, objectives must be reformulated in order to take into account the diversity of factories goals and the interdependence between factories.

Thus, in this paper, an approach for optimizing different objectives in distributed manufacturing systems with global objectives for the entire system and local objectives for factories is proposed. The methodology developed aims to find solutions that harmonize the individual priorities of each factory, while guaranteeing the overall efficiency of the system. Several global and local objectives are formulated. A set of DJSS instances with unrelated parallel machines and sequence-dependent setup times are solved using NSGA-III [4]. The effect of global and local objectives on the solutions obtained is then studied and analyzed.

The structure of the document is as follows. Section 2 briefly presents the distributed job shop scheduling framework, including the description of the general framework implemented and the formulation of global and local objectives. In Sect. 3, the results of several combinations of global and local objectives using

NSGA-III on a set of DJSS instances are analyzed and discussed. Finally, in Sect. 4, some conclusions and future work are addressed.

2 Distributed Job Shop Scheduling Framework

2.1 Framework Description

A general framework for solving Distributed Job Shop Scheduling (DJSS) and Job Shop Scheduling (JSS) problems was developed and implemented in Python programming language. In Fig. 1, it is presented a diagram that describes this general framework. First, the user must define the DJSS problem by indicating the number of factories, the number of machines in each factory, and the number of jobs to be processed. This way, the distributed production systems is defined. It should be noted that a JSS problem can be also defined by indicating a number of factories equal to one, i.e., it is a particular case of a DJSS where the number of factories is one. Next, the number of objectives, type of global or local objectives must be defined. DJSS problems can be formulated with one or more conflicting objectives, this latter case correspond to a multi-objective problem. Moreover, objectives can express global and local goals. Global objectives refer to the overall objectives of the organization. On the other hand, local objectives are related with local criteria of each factory of the organization. Finally, specific problem data have to be provided, namely the setup times (order dependent), the processing times of each job on each machine of each factory, and the transportation times of each job to each factory. All problem data is saved in a file that will be read later by the optimization algorithm.

Afterwards, an optimization algorithm (e.g., GA, NSGA-II, NSGA-III among others) has to be selected taking into account the characteristics or properties of the problem being solved. Meta-heuristics such as evolutionary algorithms [5] are often used to solve these DJSS optimization problems that are combinatorial in nature. Evolutionary algorithms are inspired by the genetic process of biological organisms, which evolve through natural selection. These algorithms are particularly suited to solve combinatorial problems since permutation representation of solutions and specific genetic operators can be used. Furthermore, in general, several conflicting objectives exist and evolutionary algorithms that work with a population of solutions can find an approximation of the Pareto optimal frontier. In this implementation, the optimization algorithms have to be selected by the user from the PyMOO [2] framework.

For all algorithms, the user can modify their parameters and stopping criteria or use the default setup (in this case, default algorithm parameters are used). Next, the optimization algorithm runs and the results are stored in an output file. This file contains data regarding the solutions of the problem and the respective objective functions values. Other data related with the optimization process can be also stored. Finally, a set of tools can be selected to present and visualize the results such as Gantt charts to represent solutions, graph representations of the

Pareto optimal frontier or parallel coordinate plots to identify trade-offs. These visualization tools intend to facilitate the analysis of solutions and also facilitate the decision-making process.

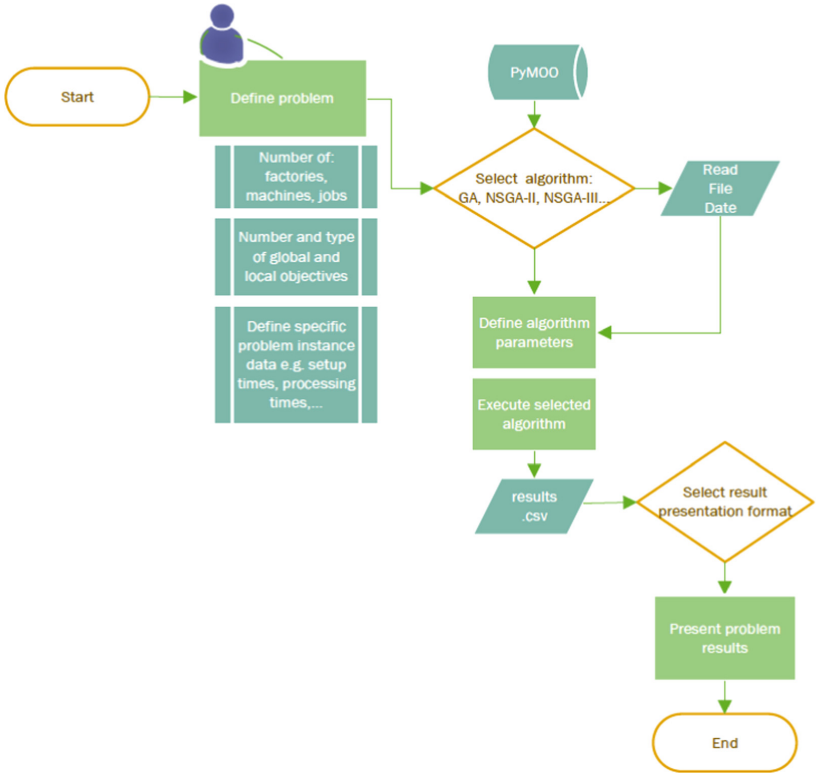


Fig. 1. Diagram of the implemented system for solving DJSS problems.

2.2 Global and Local Objectives Formulation

In DJSS problems, machines are distributed across multiple factories located in different geographic locations. This adds an extra layer of complexity to the problems, differing from traditional JSS problems in that factors such as transportation times of raw materials between factories, as well as processing and setup times for jobs on machines of each factories must be considered. Each factory is considered autonomous and capable of performing any job. Therefore, in order to formulate the optimization problem with global and local objectives, the user must indicate all information to describe the distributed manufacturing system.

First, the following sets must be defined by the user in order to characterize the DJSS problem:

- $\mathbb{N}$ is the set of N jobs, i.e., $\mathbb{N} = \{n_1, n_2, n_3, ...n_N\}$;
- $\mathbb{F}$ is the set of F factories, i.e. $\mathbb{F} = \{f_1, f_2, f_3, ...f_F\}$;
- $\mathbb{M}_u$ of M_u is the set of of machines for each factory $u \in \mathbb{F}$, i.e. $\mathbb{M}_1 = \{m_{1,1}, \ldots, m_{1,M_1}\}, \mathbb{M}_2 = \{m_{2,1}, \ldots, m_{2,M_2}\}, \ldots, \mathbb{M}_F = \{m_{F,1}, \ldots, m_{1,M_F}\}$.

The following assumptions are considered: for each factory u, each machine M_u can only execute one job N at each instant; all machines are always available; there is no precedence in the N jobs, each of them being independent; all jobs are available to be executed at the initial moment; and, no preemption is allowed, i.e., the processing of a job in a machine cannot be interrupted.

The following data must defined by the user: processing times $p_{u,i,j}$ for each machine i on factory u, and each job j; setup times $s_{u,i,j,k}$ for each machine i on factory u, to process job j after job k; and, transportation times $t_{u,j}$ to transport job j to factory u.

The following nomenclature and notation for the DJSS problem is considered: π is the feasible sequence of all jobs assigned to machines in each factory, i.e., the vector of all feasible subsequences $\pi_{u,i}$ on each machine $i \in \mathbb{M}_u$ of factory $u \in \mathbb{F}$; and $c_{u,i}(\pi)$ is the total completion time of jobs on machine $i \in \mathbb{M}_u$ of factory $u \in \mathbb{F}$ for the feasible sequence π. So, the goal in a DJSS optimization problem is to determine a feasible sequence of jobs on each machine in the different factories, in order to simultaneously minimize one or more objectives or criteria.

Therefore, after indicating all parameters of the DJSS problem, it is required to formulate objectives, such as minimizing the maximum completion time (makespan), minimizing the average completion times, minimizing the standard deviations of completion times, minimizing the total delay, minimizing the average delay, among others. In general, these objectives are formulated aiming to optimize the overall efficiency and performance of the distributed system, i.e., they can be regarded as global objectives across all factories. In [11], the makespan and the average completion time of the jobs were considered as objectives in a global perspective. However, local objectives for factories can be different due to its heterogeneity in distributed production systems. Thus, it is crucial to formulate objectives that considers these two perspectives and allow to find solutions that efficiently balance local and global requirements.

The following nomenclature is adopted to facilitate the formulation of objectives that considers both global and local objectives of the DJSS problems: C_G^L represents an objective in which G is the global objective and L is the local objective of completion times. Several different functions can be defined for G and L. In this work, $G, L \in \{\max, \mu, \sigma\}$ where max, μ and σ are the maximum, average and standard deviation functions, respectively. For instance, in [11], the objectives considered were minimizing the maximum completion time (makespan), and minimizing the average completion times, i.e., $\min C_{\max}^{\max}(\pi) = c_{\max}(\pi)$ and $\min C_{\mu}^{\mu}(\pi) = \bar{c}(\pi)$, respectively.

Let us start by defining objectives in which global and local objectives coincide. These objectives can be expressed by the following equations:

$$C_{\max}^{\max}(\pi) = c_{\max}(\pi) = \max_{u=1,\ldots,F} \left(\max_{i=1,\ldots,M_u} (c_{u,i}(\pi)) \right) \tag{1}$$

$$C_{\mu}^{\mu}(\pi) = \bar{c}(\pi) = \sum_{u=1}^{F} \frac{1}{M_u} \sum_{i=1}^{M_u} c_{u,i}(\pi) \tag{2}$$

$$C_{\sigma}^{\sigma}(\pi) = c_{\sigma}(\pi) = \sqrt{\frac{\sum_{u=1}^{F} \sum_{i=1}^{M_u} (c_{u,i}(\pi) - \bar{c_u}(\pi))^2}{\sum_{u=1}^{F} M_u}} \tag{3}$$

for the maximum completion time (makespan), the average completion time, and the standard deviation completion time, respectively. It should be noted that in each of these equations the global and local objectives are the same.

Let us now consider situations in which global and local objectives are different. Several combinations of global and local can be defined. For instance, the following objective combines the makespan as global objective with the average completion time as local objective for factories:

$$C_{\max}^{\mu}(\pi) = \max_{u=1,\ldots,F} \left(\frac{1}{M_u} \sum_{i=1}^{M_u} c_{u,i}(\pi) \right) \tag{4}$$

On the other hand, the following objective combines the makespan as global objective with the standard deviation of the completion time as local objective for factories:

$$C_{\max}^{\sigma}(\pi) = \max_{u=1,\ldots,F} (\sigma_u(\pi)) = \max_{u=1,\ldots,F} \left(\sqrt{\frac{\sum_{i=1}^{M_u} (c_{u,i}(\pi) - \bar{c_u}(\pi))^2}{M_u}} \right) \tag{5}$$

Other combinations of global and local objectives can be defined. In the above examples, the local objectives are equal for all factories. However, it should be noted that a different local objective for each factory could be defined. Different combinations of these objectives result in different single objective or multi-objective DJSS problems.

3 Experimental Results

3.1 Implementation Details

In this paper, some DJSS problem instances were taken from a previous work [11] in order to study the effect of global and local objectives. The instances are P_1, P_2 and P_3, and their characteristics are summarized below:

- P_1 - instance with 2 factories, both with 2 machines, and 6 jobs ($F = 2$, $M_1 = 2$, $M_2 = 2$, and $N = 6$);
- P_2 - instance with 2 factories, one with 2 machines and the other with 4 machines, and 11 jobs ($F = 2$, $M_1 = 2$, $M_2 = 4$, and $N = 11$);
- P_3 - instance with 3 factories, two of them with 2 machines and the other with 4 machines, and 11 jobs ($F = 3$, $M_1 = 2$, $M_2 = 2$, $M_3 = 4$, and $N = 11$).

The following four bi-objective DJSS problems were considered for instances P_1, P_2 and P_3:

$$\min \mathbf{F_1}(\pi) = \left(C^{\max}_{\max}(\pi), C^{\mu}_{\mu}(\pi)\right)^T \tag{6}$$

$$\min \mathbf{F_2}(\pi) = \left(C^{\max}_{\max}(\pi), C^{\mu}_{\max}(\pi)\right)^T \tag{7}$$

$$\min \mathbf{F_3}(\pi) = \left(C^{\max}_{\max}(\pi), C^{\sigma}_{\sigma}(\pi)\right)^T \tag{8}$$

$$\min \mathbf{F_4}(\pi) = \left(C^{\max}_{\max}(\pi), C^{\sigma}_{\max}(\pi)\right)^T \tag{9}$$

where π is a feasible sequence belonging to the feasible set Ω (the set of all possible permutations) and $\mathbf{F_1}(\pi)$, $\mathbf{F_2}(\pi)$, $\mathbf{F_3}(\pi)$, and $\mathbf{F_4}(\pi)$ are the vectors of objective functions defined in the objective space for each problem.

As in [11], NSGA-III was selected from PyMOO [2] framework to solve all DJSS problems. The population size was 100 chromosomes. The feasible sequence of jobs on the machines in different factories, i.e. the π sequence, is represented by a chromosome as a permutation of size $N + \sum_{u=1}^{F} M_u - 1$. Positions in the chromosome with genes larger than N divide the chromosome into subsequences $(\pi_{u,i})$ representing the set of jobs and the corresponding order assigned to each machine $i \in \mathbb{M}_u$ in the factory $u \in \mathbb{F}$. The genetic operators used were similar to the ones used in [11], i.e., binary tournament selection operator to select chromosomes for reproduction, and the order-based crossover and the inversion mutation to generate offspring from parents. All genetic operators guarantee the feasibility of all solutions during the search. The stopping criteria was the same used in [11], i.e., a maximum number of generations of 10,000 and a period at 100 generations for testing the nonexistence of improvement in the search. Due to stochasticity of the NSGA-III, 20 independent runs were performed.

3.2 Results for F_1 and F_2 Problems

In this section, the results for instances P_1, P_2, and P_3 for F_1 and F_2 problems are presented and discussed. In these problems, one of the objectives correspond to the minimization of the makespan both globally and locally ($C^{\max}_{\max}$). The other objective is the minimization of the average completion time globally and locally (problem F_1) or the makespan globally and the average completion time locally (problem F_2). Figure 2, Fig. 3, and Fig. 4 show the approximations to the Pareto optimal frontier for P_1, P_2, and P_3 instances, respectively. In all these figures, the X-axis corresponds to $C^{\max}_{\max}$ objective and the Y-axis indicates the values of C^{μ}_{μ} and $C^{\mu}_{\max}$ objectives. All these figures also includes two pay-off tables indicating the values of the objectives for each solution for F_1 and F_2 problems. The table closest to the top right corner of the graph is the pay-off table for problem F_1 (the other is for problem F_2). The best approximations to the Pareto optimal set obtained by NSGA-III are presented. So, the solutions obtained in the run with the highest hypervolume (i.e., the best approximation to the Pareto optimal set) among the 20 independent runs are shown.

Instance P_1 In Fig. 2, the approximations to the Pareto optimal frontier for F_1 and F_2 problems for instance P_1 are represented. In this instance, for the objectives of F_1 problem, three different non-dominated solutions (solutions A, B, and C) were obtained. For the objectives of F_2 problem, four non-dominated solutions were obtained (solutions D, E, F and G). When comparing these two sets of solutions as a whole, it can be said that solutions D to G are dominated by solutions A to C.

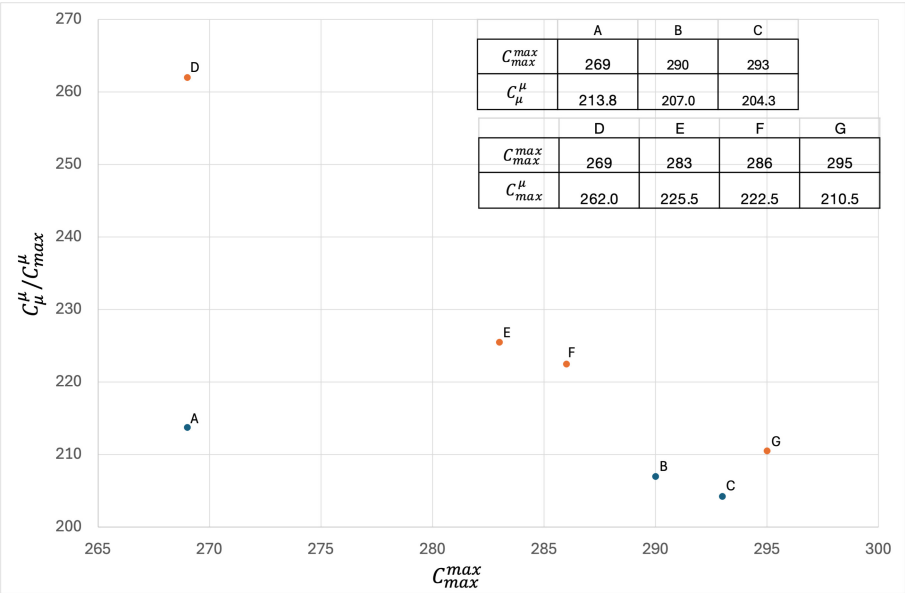


Fig. 2. Representation of the Pareto optimal frontier approximation for F_1 and F_2 problems for instance P_1.

It is worth analyzing with more detail solutions A and D since they have the same value of $C^{\max}_{\max}$. However, solution A results of prioritizing the global goal of reducing the overall average of completion times in all the manufacturing system, while solution B aims to locally reduce this value in each factory. Let us inspect the sequences of jobs assigned to each machine of each factory. Since this instance has two factories ($F = 2$), each one with two machines ($M_1 = M_2 = 2$), this sequence can be expressed as $\pi = \langle < \pi_{1,1}, \pi_{1,2} >, < \pi_{2,1}, \pi_{2,2} > \rangle$. So, solution A and solution D correspond to the following sequences of jobs, completion times, and objective values:

– Solution A: $\pi^A = \langle < [4,1], [2,6] >, < [3], [5] > \rangle$ with total completion times of $c_{1,1}(\pi^A) = 255$, $c_{1,2}(\pi^A) = 269$, $c_{2,1}(\pi^A) = 205$ and $c_{2,2}(\pi^A) = 126$, and therefore $C^{\max}_{\max}(\pi^A) = 269$, and $C^{\mu}_{\mu}(\pi^A) = 213.75$;

– Solution D: $\pi^D = \langle < [4,1],[2,6] >, < [5],[3] > \rangle$ with total completion times are $c_{1,1}(\pi^D) = 255$, $c_{1,2}(\pi^D) = 269$, $c_{2,1}(\pi^D) = 185$ and $c_{2,2}(\pi^D) = 174$, with $C^{\max}_{\max}(\pi^D) = 269$, and $C^{\mu}_{\max}(\pi^D) = 262$.

Solutions A and D are very similar, differing just in the assignment of jobs in the machines of factory 2. When these solutions are evaluated in terms the other objectives, solution A has $C^{\mu}_{\max}(\pi^A) = 262$ and solution D has $C^{\mu}_{\mu}(\pi^D) = 220.75$. Thus, solution D is dominated by solution A in a global perspective since $C^{\mu}_{\mu}(\pi^A) < C^{\mu}_{\mu}(\pi^D)$. The opposite can not be stated, i.e., solution D does not dominate solution A locally due to $C^{\mu}_{\max}(\pi^A) = C^{\mu}_{\max}(\pi^D)$.

Instance P_2. Figure 3 shows the approximations to the Pareto optimal frontier for F_1 and F_2 problems for instance P_2. Solutions A to D are the non-dominated solutions for problem F_1. Solutions E to G are the non-dominated solutions obtained for problem F_2. Taking into account the comparison of these two sets of solutions as a whole, t can be said that solutions F and G are dominated by solution A. However, solution E is not dominated by any other.

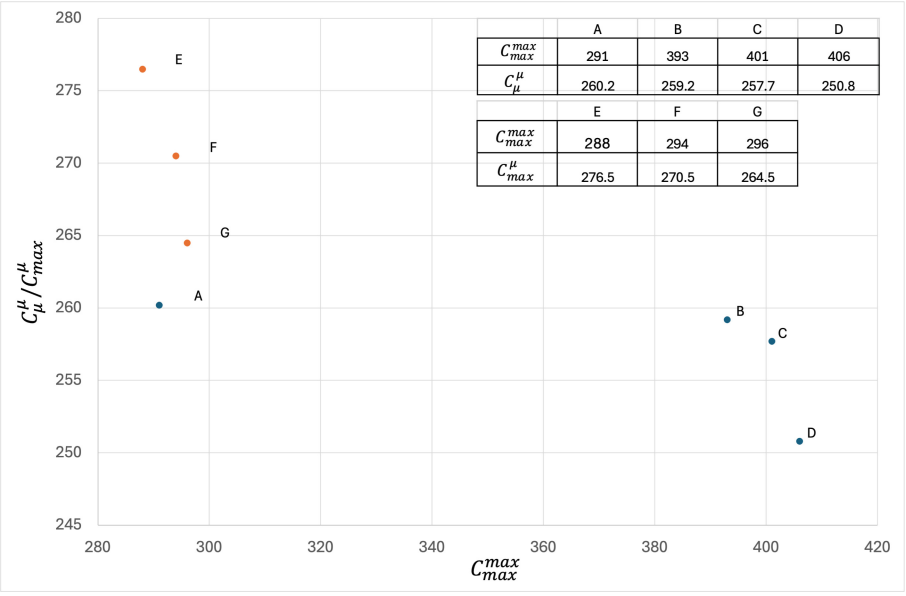


Fig. 3. Representation of the Pareto optimal frontier approximation for F_1 and F_2 problems for instance P_2.

Solution E is the solution with lower value of $C^{\max}_{\max}$ (even inferior to the one of solution A). It is interesting to inspect the scheduling plans defined the sequences of these two solutions. This instance P_2 has two factories ($F = 2$), one with two machines ($M_1 = 2$) and the other with four machines ($M_2 = 4$), this sequence can be expressed as $\pi = \langle < \pi_{1,1}, \pi_{1,2} >, < \pi_{2,1}, \pi_{2,2}, \pi_{2,3}, \pi_{2,4} > \rangle$:

– Solution A: $\pi^A = \langle < [4,8],[7,10] >, < [1],[3,2],[11,9],[5,6] > \rangle$;
– Solution E: $\pi^E = \langle < [4,5],[7,10] >, < [8,1],[2],[11,9],[3,6] > \rangle$.

As expected, it can be observed that there exist some similarities between the two solutions. On the other hand, it is worth emphasizing that, in this case, using a local perspective allows to obtain a solution that is better in terms of $C^{\max}_{\max}$.

Instance P_3. Figure 4 shows the approximations to the Pareto optimal frontier for F_1 and F_2 problems for instance P_3. Solutions A to D and solutions E to H are the non-dominated solutions for F_1 and F_2 problems, respectively.

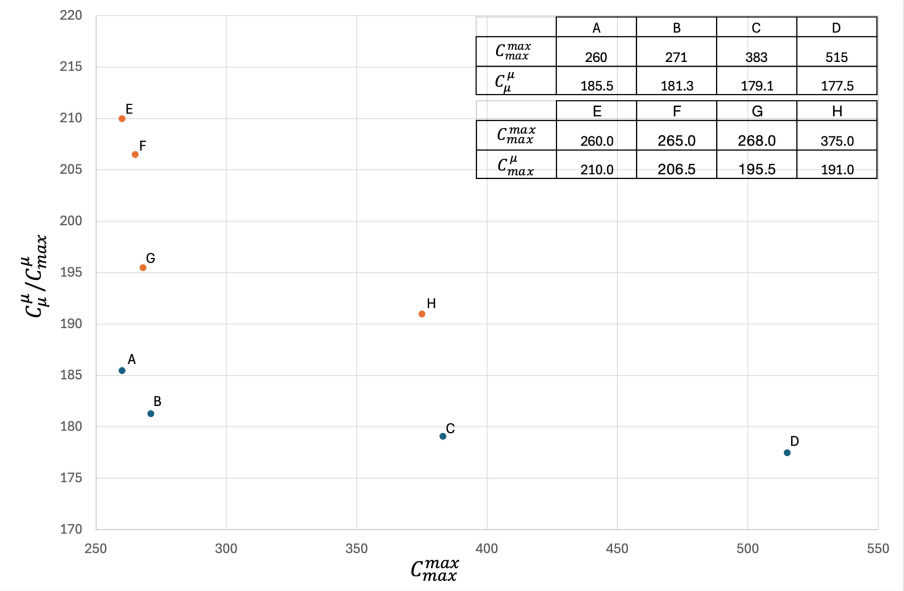


Fig. 4. Representation of the Pareto optimal frontier approximation for F_1 and F_2 problems for instance P_3.

When comparing the two sets of solutions, it can be seen that solutions of F_1 problem dominate solutions of F_2 problem. Nevertheless, it is worth analyzing solutions A and E since they have the same value of $C^{\max}_{\max}$. For instance P_3, the sequences of jobs assigned to each machine of each factory can be expressed as $\pi = \langle < \pi_{1,1},\pi_{1,2} >, < \pi_{2,1},\pi_{2,2} >, < \pi_{3,1},\pi_{3,2},\pi_{3,3},\pi_{3,4} > \rangle$:

– Solution A: $\pi^A = \langle < [7,2],[8] >, < [1,6],[9] >, < [3],[5,4],[10],[11] > \rangle$;
– Solution E: $\pi^E = \langle < [7,3],[1] >, < [9,11],[2] >, < [6],[5,4],[8],[10] > \rangle$.

These two solutions does not have any subsequences in common, except for the second machine of the third factory. In a global perspective, solution A seems to be preferable.

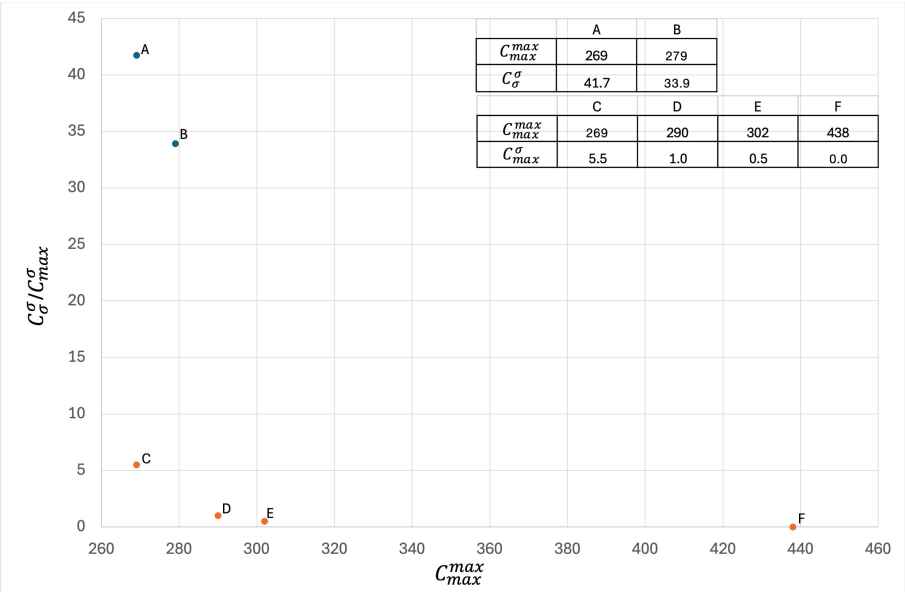
3.3 Results for F_3 and F_4 Problems

The results for instances P_1, P_2, and P_3 for F_3 and F_4 problems are presented and discussed in this section. Like in F_1 and F_2 problems, one of the objectives is the minimization of the makespan both globally and locally ($C_{\max}^{\max}$). However, the other objective is the minimization of the standard deviation of the completion time both globally and locally (problem F_3) or the makespan globally and the standard deviation of the completion time locally (problem F_4). Figure 5, Fig. 6, and Fig. 7 depict the approximations to the Pareto optimal frontier for P_1, P_2, and P_3 instances, respectively. As before, in all these figures, the X-axis corresponds to $C_{\max}^{\max}$ objective, but now the Y-axis indicates the values of C_{σ}^{σ} and $C_{\max}^{\sigma}$ objectives. In each figure, the two pay-off tables indicate the values of the objectives for each solution for F_3 and F_4 problems. The approximations to the Pareto optimal set obtained by NSGA-III with highest hypervolume among the 20 independent runs are presented.

Instance P_1. Figure 5 shows the approximations to the Pareto optimal frontier for F_3 and F_4 problems for instance P_1. There are two non-dominated solutions for problem F_3 (solutions A and B). Solutions C to F are the non-dominated solutions for problem F_4. The solutions with the smallest values of $C_{\max}^{\max}$ are solutions C and A. However, globally, the set of solutions A and B are dominated by the set of solutions C to F. It should be noted that the maximum standard deviation of the completion times is zero. This means that the load associated to the jobs is balanced between machines in the two factories. Overall, in this instance, it can be observed that local prioritization allowed to achieve solutions the correspond to better balance of loadings between machines in each factory.

Instance P_2. In Fig. 6, the approximations to the Pareto optimal frontier for F_3 and F_4 problems for instance P_2 are shown. Four non-dominated solutions were found for problem F_3. For problem F_4, two solutions were obtained. These two solutions have extrema values of $C_{\max}^{\max}$. When the two sets of solutions for F_3 and F_4 problems are compared, it can be seen that they are incomparable each other. So, neither set dominates the other. Overall, in this instance, solution E has the lowest makespan. Conversely, solution F has the worst makespan, but the best workload balance.

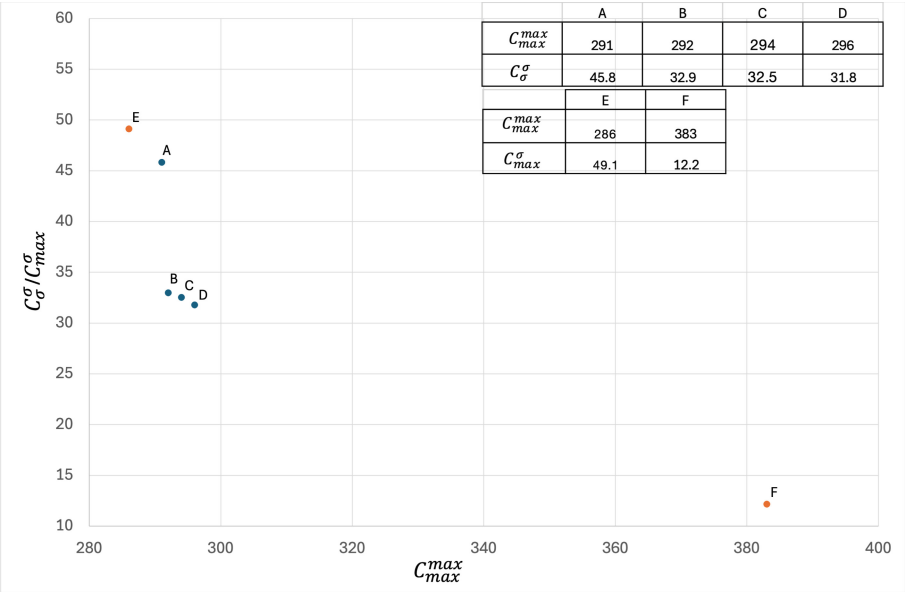
Instance P_3. The approximations to the Pareto optimal frontier for F_3 and F_4 problems for instance P_3 are represented on Fig. 7. For problem $F3$, a single solution was found (solution A). This means that the two objectives of problem F_3 are not conflicting in this instance. Nevertheless, for problem F_4, this conflict exist since four solutions (solutions B to E) were achieved. Solutions A and B have the same makespan value, but first one seems to be preferable in terms workload balance. Solutions D and E have large makespan values when compared with the remaining ones.



	A	B
C_{max}^{max}	269	279
C_{σ}^{σ}	41.7	33.9

	C	D	E	F
C_{max}^{max}	269	290	302	438
C_{max}^{σ}	5.5	1.0	0.5	0.0

Fig. 5. Representation of the Pareto optimal frontier approximation for F_3 and F_4 problems for instance P_1.



	A	B	C	D
C_{max}^{max}	291	292	294	296
C_{σ}^{σ}	45.8	32.9	32.5	31.8

	E	F
C_{max}^{max}	286	383
C_{max}^{σ}	49.1	12.2

Fig. 6. Representation of the Pareto optimal frontier approximation for F_3 and F_4 problems for instance P_2.

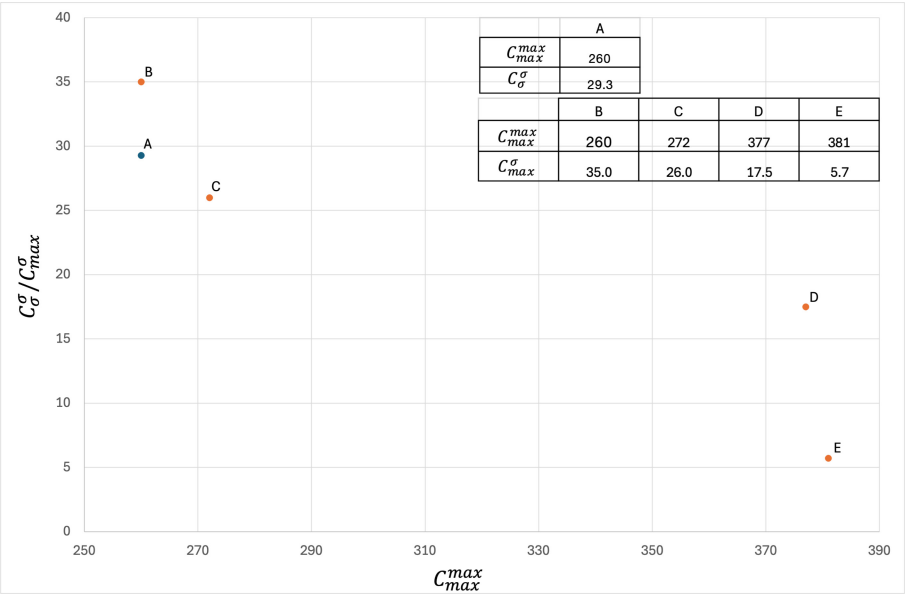


Fig. 7. Representation of the Pareto optimal frontier approximation for F_3 and F_4 problems for instance P_3.

4 Conclusions and Future Work

In this paper, a DJSS framework is proposed to tackle DJSS problems. In particular, the effect of global and local objectives in DJSS problems is studied. Several optimization objectives are formulated and proposed to reflect global and/or local goals related with the distributed manufacturing systems. Some DJSS instances (borrowed from [11]) with unrelated parallel machines and sequence-dependent setup times were solved using NSGA-III to test and analyze how the objectives affect the solutions obtained for the distributed manufacturing system. For this purpose, various combinations of global and local objectives were considered, namely related with makespan, average completion times and standard deviation of the completion times.

The preliminary results indicate that this approach can be useful for decision-making since combines global and local goals, i.e., takes into account simultaneously a global perspective of the organization as a whole system and the particular priorities of the factories. As shown by the results, the solutions have meaning and represent different alternatives that can not be assessed by the decision-maker if just global objectives were formulated.

Moreover, it is clear that the decision-making process is facilitated and may involve both global and local decision-makers. They can analyze trade-offs both in terms of global and local objectives and the scheduling plans they represent.

Future work will include formulating more objectives expressing other priorities both global and local, involving local decision-makers in the process, apply-

ing this framework to other instances of DJSS problems, and investigating the use of bi-level approaches for DJSS problems.

References

1. Abdelmaguid, T.F.: Representations in genetic algorithm for the job shop scheduling problem: a computational study. J. Softw. Eng. Appl. **3**(12), 1155 (2010)
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. Manufact. **22**(5–6), 493–504 (2006)
4. 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)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
6. Fu, Y., Hou, Y., Wang, Z., Wu, X., Gao, K., Wang, L.: Distributed scheduling problems in intelligent manufacturing systems. Tsinghua Sci. Technol. **26**(5), 625–645 (2021). 10.26599/TST.2021.9010009
7. González-Neira, E.M., Urrego-Torres, A.M., Cruz-Riveros, A.M., Henao-García, C., Montoya-Torres, J.R., Molina-Sánchez, L.P., Jimenez, J.F.: Robust solutions in multi-objective stochastic permutation flow shop problem. Comput. Indust. Eng. **137**, 106026 (2019)
8. 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)
9. Ruiz, R., Allahverdi, A.: Minimizing the bicriteria of makespan and maximum tardiness with an upper bound on maximum tardiness. Comput. Oper. Res. **36**(4), 1268–1283 (2009)
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: International Conference on Computational Science and its Applications, pp. 495–507. Springer (2023)
11. dos Santos, F., Costa, L., Varela, L.: A multi-objective approach for solving distributed job shop scheduling problems. In: International Conference on Optimization, Learning Algorithms and Applications, pp. 326–339. Springer (2024)
12. 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: Computational Science and Its Applications–ICCSA 2022 Workshops: Malaga, Spain, July 4–7, 2022, Proceedings, Part II, pp. 157–173. Springer (2022)
13. 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 (Abr 2024)
14. dos Santos, F., Costa, L.A., Varela, L.: Performance comparison of NSGA-II and NSGA-III on bi-objective job shop scheduling problems. In: International Conference on Optimization, Learning Algorithms and Applications, pp. 531–543. Springer (2024)
15. Yenisey, M.M., Yagmahan, B.: Multi-objective permutation flow shop scheduling problem: literature review, classification and current trends. Omega **45**, 119–135 (2014)