UNIVERSITÀ DELLA CALABRIA

Exploring Flexibility in Job Shop Scheduling: Mathematical Models and Solutions

Mastroianni, Matteo
Paparazzo, Francesco
Cantafio, Matteo

Gestione e Controllo della produzione

Academic Year 2023/2024

Abstract

This paper discusses the Flexible Job-Shop Scheduling Problem (FJSP), an extension of the Job Shop Scheduling Problem (JSP), highlighting its complexity due to the simultaneous consideration of job routing and operation scheduling. The authors propose two Mixed Integer Linear Programming (MILP) models, MILP-1 and MILP-2 to address FJSPs and FJSPs with process plan flexibility (FJSP-PPFs). The paper reviews the literature on FJSP, examining various approaches such as fuzzy logic, genetic algorithms, mathematical models, and metaheuristic algorithms. Additionally, it presents computational results for MILP-2, indicating its success in identifying optimal solutions for certain problem instances. Ultimately, an extension of MILP-1 is presented and developed in Python, taking into account the efficiency parameters of the machines incorporated as input.

1 Introduction

The job shop scheduling problem (JSP) falls within the domain of production scheduling and is considered one of the most challenging combinatorial optimization problems. The primary objective of the job shop scheduling problem is to establish a timetable for jobs with pre-defined operation sequences within a multi-machine setting. The flexible job-shop scheduling problem (FJSP) is an extension of the traditional JSP, allowing operations to be executed on any available machine within a designated set. FJSP encompasses a routing sub-problem, involving the assignment of each operation to a machine from a set of capable machines, and a scheduling sub-problem. The scheduling sub-problem entails sequencing the assigned operations across all machines to achieve a viable schedule that minimizes a predefined objective function. Consequently, solving the FJSP is more challenging due to the simultaneous consideration of job routing and operation scheduling, leading to its classification as an NP-hard problem. FJSP incorporates all the intricacies of JSP and introduces additional complexity by requiring the determination of the assignment of operations to machines. In our paper, the authors adapted a model initially developed by Manne [1] for JSPs to make it applicable for solving FJSPs. The resulting model is denoted as MILP-1. Subsequently, a new mechanism is integrated into MILP-1 to address FJSP-PPFs (FJSP with process plan flexibility), which includes the additional sub-problem of process plan selection. The modified model is referred to as MILP-2.

1.1 Organization of the paper.

This paper is organized as follows. In Section 2 we reported the State-of-Art, where we give an overview of previous research related to the study of FJSP, from past research to the most recent, alleging tables and graphics. In Section 3 we provided a description of the problem proposed by the authors, presenting the mathematical model, with an explanation of constraints and variables. In Section 4 we reported the main results obtained by the authors, applying the model described to test problems. In Section 5 we extended the model by taking in consideration the efficiency parameters of the machines. In Section 6 we have briefly summarized the content of the entire paper, proposing some future research directions related to the discussed problem.

2 State-of-Art

In this section, from the assigned paper, we intend to compile a history to the past of the literature regarding the FJSP. We selected the papers by focusing on applications that were as diversified as possible from each other, in order to delve into different variants of the problem under analysis. **Kacem et al.(2002)** [2] suggests a Pareto strategy based on the hybridization of fuzzy logic (FL) and evolutionary algorithms (EAs). **Gao et al. (2007)** [3] developed a new genetic algorithm hybridized with an innovative local search procedure for the problem (bottleneck shifting) to address FJSP with 3 objectives: minimizing the makespan, minimizing the maximum machine workload and minimizing the total workload. **Fattahi et al. (2007)** [4] addressed job scheduling problems in a parts manufacturing company, considering a mathematical model and heuristic approaches. The mathematical model is employed to obtain an optimal solution for small-sized problems, and two heuristic approaches have been developed involving integrated and hierarchical methods to solve real-sized problems. **Demir et al. (2012)** [5] proposed four important formulations of the FJSP and a new time-indexed model. The evaluation of these formulations is divided into three groups based on the types of binary variables they use to sequence operations on the machines. In **Ebadi et al. (2012)** [6] the model's unique feature is its allowance for preemption, enabling the interruption and resumption of an operation at any time. It utilizes a disjunctive graph, where operations are represented by nodes and time constraints by edges. Precedence is considered among operations of the same job and those of different jobs processed on the same machine. An exact approach is proposed, leveraging a branch and bound algorithm. This algorithm uses a graphical representation of the problem, along with various dominance, bounding, and preference techniques, to effectively reduce the search space and find the optimal solution. The problem is then formulated as selecting disjunctive edges to create an acyclic subgraph. The goal is to minimize the longest path from the source node to the sink node, optimizing the overall scheduling process. **Chen et al. (2022)** [7], modeled the flexible job shop scheduling problem under the process resource preemption scenario, and a two-layer rule scheduling algorithm based on deep reinforcement learning is suggested to meet the goal of shortest scheduling time. **Birgin et al.'s (2013)** model [8] stands out for its general flexibility, accommodating complex job types, including parallel or assembly operations. It's compact and robust, characterized by fewer variables and constraints, and it demonstrates superior linear relaxation quality with a smaller integrality gap. In terms of effectiveness and speed, it outperforms Özgüven et al.'s model by solving more instances more quickly and showing a better ability to provide optimal solutions or upper bounds. **Hansmann et al.'s (2013)** [9] model addresses the flexible job shop scheduling problem with blockages, a unique aspect not present in Birgin's model. This model is applied to the rail car maintenance problem, where an occupied machine blocks access to subsequent ones. Despite its complexity, being weak NP-hard for a single work center with two machines and strongly NP-hard for a single work center, the authors propose a solution using a mixed integer linear optimization model, heuristic methods, and a branch and bound algorithm. **Mokhtari et al. (2015)** [10] presented a model that address the FJSP with preventive maintenance and time-varying failure rates; it's a Mixed-Integer Linear Programming (MILP) model and it considers stochastic parameters for processing times and due dates, and integrates routing, sequencing, and maintenance decisions. The model uses a metaheuristic algorithm combining simulation and optimization, employing a simulated annealing (SA) algorithm for exploring the solution space. This makes it applicable to various production systems requiring flexible manufacturing processes and preventive maintenance. The application of FJSP, however, is not limited to optimizing production efficiency, cost or quality; indeed **Yin et al. (2016)** [11] presented a model that focuses on minimizing energy consumptions and environmental pollution in a work-shop. They proposed a new mathematical approach and the use of a multi-objective genetic algorithm based on a simple lattice design to solve this mixed-integer programming problem. Also for **Meng et al. (2018)** [12] the objective is minimizing the total energy consumption of the workshop. They proposed six new Mixed-Integer Linear Programming models (MILP) with an on/off strategy, divided into 2 families. The first one is characterized by a non-linear objective function, that requires linearization, while the second one has an originally linear objective function. **Shen et al. (2017)** [13] developed a model that use tabu search, an intelligent metaheuristic algorithm, to explore the solution space without getting trapped in local optimum. It defines a neighbourhood structure with movement functions that allow operations to be inserted, removed or swapped between machines or on the same machine, while respecting feasibility conditions. The model is based on some structural properties of the problem, such as the concept of disjunctive graph, the calculation of the heads and tails of the operations, the definition of critical operations and blocks, and the use of preliminary evaluations of the neighbors to identify the most promising ones. **Zhu et al. (2020)** [14] proposed a more sophisticated and advanced solution for a more general scheduling problem for a FJSP with multiple process plans . They considered various realistic flexibilities such as processing flexibility, machine flexibility, and sequence flexibility and they presented an evolutionary approach based on genetic programming to automatically generate effective dispatching rules for the given problem with an evaluation method to assess generated dispatching rules. **Ghaedy-Heidary et al. (2023)** [15] analyzed the application of FJSP in the context of semiconductor production, addressing a stochastic flexible job-shop scheduling problem (SJSSP). They created a mathematical model, and then it's transformed into a Simulation Optimization model that is combined with a computer simulation model that can simulate the photolithography workstation. The Least Work Remaining (LWR) dispatching rule is the basis for the first schedule that the simulation model creates. Additionally, the simulation model determines the SFJSSP's objective function. The last paper we selected is **Liu et. al's (2024)** [16] paper, where they addressed an extension of the flexible manufacturing challenge, that is the problem of flexible job shop batch scheduling. They analyzed the coupling relationship between batching methods and scheduling optimization to build a flexible job shop batch production model that takes the set-up time into account. An enhanced genetic algorithm with a three-layer coding mechanism is created based on the suggested model to address the issue of the solution space enlargement brought on by batch collaboration. The batch strategy, operation selection, and machine selection are components of the three-layer coding process. To increase production resilience and efficiency, an operation overlapping technique is also implemented.

In Table 1 we presented the main features of the selected paper for the state-of-art. We used eight different features to describe each paper. The first one "**Algorithm**" specifies which type of approach was used for the problem. The "−" means that it was used a commercial solver to solve small instances of the presented problems without presenting new solution's methods. In this column there are also "**Heuristic**" and "**Metaheuristic**"; the difference between this two types of approach is that a metaheuristic is a higher-level strategy used to select and manage the lower-level heuristics. This makes metaheuristics more flexible and widely applicable than heuristics. The second features "**Solution Technique**" specifies which kind of method was used to solve the optimization problem. "**Parameters**" indicates the nature of used parameters, meaning if parameters are deterministic or stochastic. "**Model Type**" specifies the kind of mathematical model used to describe the problem "**Objective Function** indicates the objective to optimize, while "**Industrial Application** indicates if the paper's model was applied to a real-world industrial scenario. The column " **Preemption Allowed**" specifies if the tasks in the model can be preempted (interrupted and resumed later). Ultimately, the last column "**Setup Times**" indicates whether setup times were included within the processing times parameter or not.
Later, we also provided a graphical representation of the main keywords for each paper in **Figure 1**

Table 1: Main features of the papers reviewed

Reference	Algorithm	Solution Techniques	Parameters	Model Type	Objective Function	Industrial Application	Preemption Allowed	Setup Times
Kacem et al. (2002) [2]	-	Fuzzy multi-objective evaluation stage, Evolutionary multi-objective optimization stage	Deterministic	Hybrid	Min Cmax, Min. workload of the most loaded machine, Min. total workload of machines	-	No	Not considered
Gao et al. (2007) [3]	Heuristics	Genetic algorithm hybridized with bottleneck shifting	Deterministic		Min. Cmax, Min. maximal machine workload, Min. total workload	-	No	Not considered
Fattahi et al. (2007) [4]	Heuristic	Hierarchical and Integrated Algorithms	Deterministic	MILP	Min. Cmax	Production company for parts for petrochemical industries	No	Not considered
Özgüven et al.(2010) [17]	-	Not specified	Deterministic	MILP	Min. Cmax	-	No	Not considered
Ebadi et al. (2012) [6]	Exact	Branch and bound	Deterministic	Disjunctive Graph	Min. Cmax	-	√	Not considered
Demir et al. (2012) [5]	Heuristic	Not specified	Deterministic	MILP	Min. Cmax		No	Not considered
Birgin et al. (2013) [8]	-	Not specified	Deterministic	MILP	Min. Cmax	-	No	Not considered
Hansmann et al. (2013) [9]	Heuristic	Greedy Algorithm and Branch and bound	Deterministic	MILP	Min. Cmax	Railcar Maintenance	No	Not considered
Mokhtari et al. (2015) [10]	Metaheuristic	Simulated Annealing (SA) and Monte-Carlo (MC) simulation	Stochastic	Simulation-Optimization	Min. total number of tardy jobs	-	No	Not considered
Yin et al. (2016) [11]	Heuristic	Multi-objective genetic algorithm	Deterministic	MILP	Min. Cmax, Min. total energy consumption, Min. noise emission	-	No	Not considered
Shen et al. (2017) [13]	Metaheuristic	Tabu search	Deterministic	MILP and Tabu Search	Cmax	-	No	Sequence-dependent
Meng et al. (2018) [12]	-	Not specified	Deterministic	MILP	Min. total energy consumption	-	No	Not considered
Zhu et al. (2020)[14]	-	Genetic programming	Deterministic	MILP	Min. Cmax	-	No	Not considered
Chen et al. (2022) [7]	-	Proximal policy optimization	Deterministic	Deep reinforcement learning	Min. Cmax	Aircraft carrier deck replenishment	√	Not considered
Ghaedy-Heidary et al. (2023) [15]	Heuristic	Genetic Algorithm	Stochastic	Simulation Optimization	Min. Cmax	Semiconductor manufacturing	No	Considered
Liu et al. (2024) [16]	Heuristic	Genetic Algorithm	Deterministic	Min. Cmax	Batch production	-	Considered	

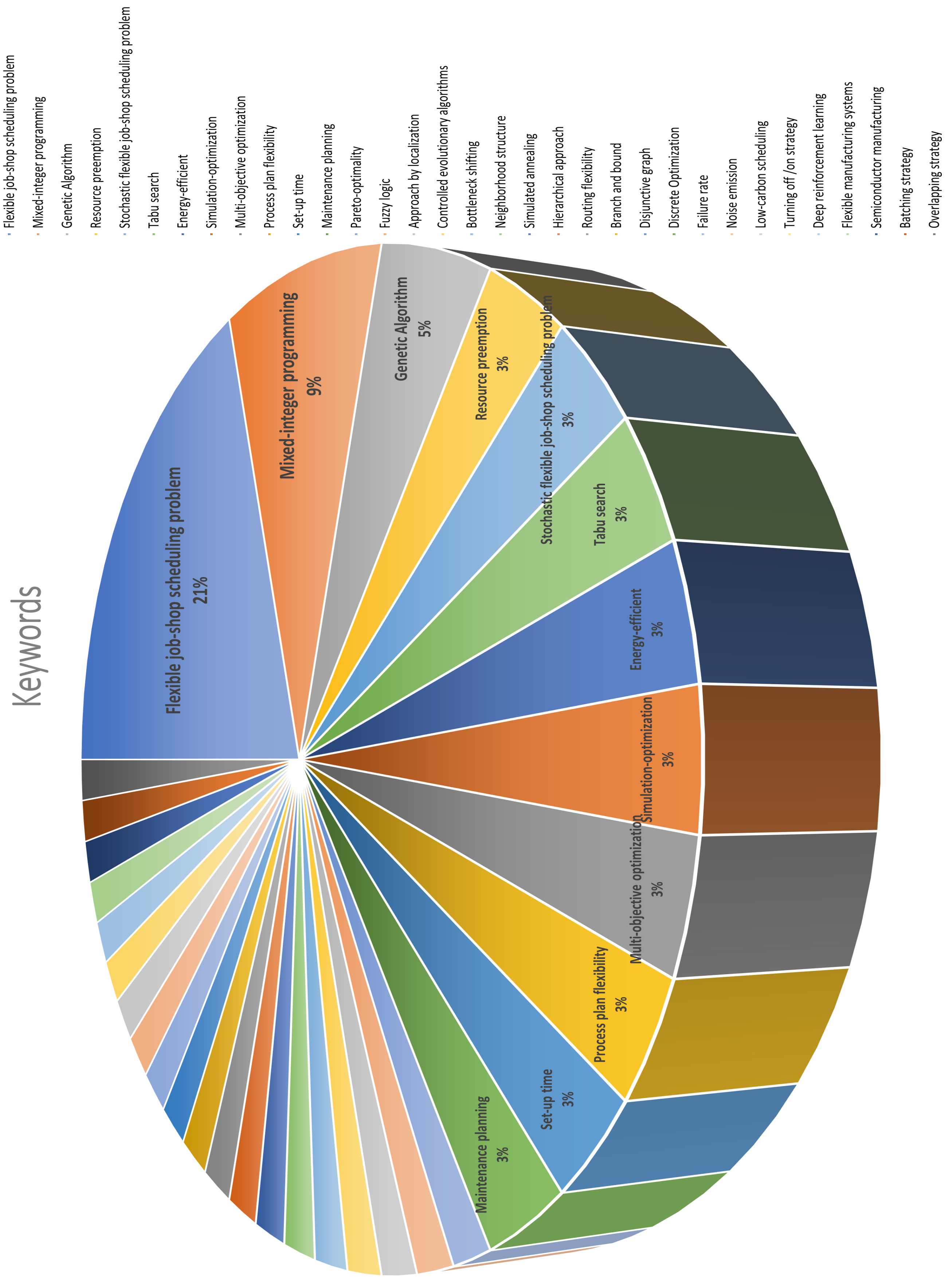

Figure 1: Keywords of the paper reviewed

3 Problem Description

In this section we want to take a more detailed look to the study conducted by **Özgüven et al.(2010) [17]**. Two formulations of the FJSP are proposed: **MILP-1** and **MILP-2**. The second one is an is extension of MILP-1 in order to enable it to handle the process plan flexibility.
The FJSP comprises a collection of n independent jobs $J = \{j_1, \ldots, j_n\}$, each with its unique processing order across a set of m machines $M = \{m_1, \ldots, m_m\}$. To complete job i, a certain number of ordered operations $\{O_{i1}, \ldots, O_{il_{(i)}}\}$ must be executed. In contrast to the JSP, where operation j of job i (O_{ij}) is fixed to a predefined machine $m_j \in M$, the FJSP allows flexibility by permitting operation j to be processed by any machine in a specified set $M_j \subseteq M$ for a given processing time t_{ijk}. The FJSP is similar to multiple unique JSPs due to its routing flexibility. Unlike the JSP, which solely deals with sequencing, the FJSP involves both routing and sequencing challenges: assigning each operation O_{ij} to a machine chosen from set M_j and arranging operations on machines to minimize $C_{\max}$.
The assumptions for FJSPs are as follows: processing times are fixed and known, setup times are either negligible or included in processing times, transportation times are disregarded, all jobs are available at time zero, pre-emption is not allowed, and each machine can process only one operation at a time. Additionally, there is only one feasible process plan for each job.

The FJSP-PPF problem takes into account multiple process plans for each job, thereby removing the final assumption of the traditional FJSP. The problem is defined over a set of n jobs, where each job i has a set of $\tau_{(i)}$ process plans denoted by $P_i = \{\rho_{i1}, \ldots, \rho_{i\tau_{(i)}}\}$. Each process plan $p_{(i)}$ for a job i is a sequence of $l_{ip_{(i)}}$ operations. These process plans are assumed to be predetermined and are characterized by linear precedence relationships. Any operation j in the process plan $p_{(i)}$ of job i (denoted by $O_{ip_{(i)}j}$) can be executed on any machine from a specified set $M_j \subseteq M$, with a given processing time $t_{ip_{(i)}jk}$. The FJSP-PPF problem involves not only the routing and sequencing sub-problems, but also the process plan selection sub-problem. This involves selecting a process plan for each job i from the set P_i, assigning each operation $O_{ip_{(i)}j}$ to a machine from the set M_j, and scheduling the operations on the machines in such a way that the makespan $C_{\max}$ is minimized.

3.1 Description of MILP-1

To define the priority relations between two different jobs, we use the binary variable defined by A.S. Manne [1] $\delta_{ii'k}$ that takes the value 1 if job i precedes job i' on machine k, or 0 if it doesn't.

MILP-1 is the model proposed to solve the FJSP and considers the parameters and variables stored in the following tables.

Indices and Sets	**Description**
$i, i' \in J$	Indices for jobs, where J is the set of all jobs.
$j, j' \in O$	Indices for operations, where O is the set of all operations.
$k \in M$	Index for machines, where M is the set of all machines.
O_i	Ordered set of operations of job i, where $O_i \subseteq O$.
M_j	Set of alternative machines on which operation j can be processed, where $M_j \subseteq M$.
$M_j \cap M_{j'}$	Set of machines on which operations j and j' can be processed.

Table 2: Indices and Sets for MILP-1

Parametres	**Description**
t_{ijk}	the processing time of operation O_{ij} on machine k
L	a large number

Table 3: Parameters for MILP-1

Variables	**Description**
X_{ijk}	1, if machine k is selected for operation O_{ij}; 0, otherwise
S_{ijk}	The starting time of operation O_{ij} on machine k
C_{ijk}	The completion time of operation O_{ij} on machine k
$Y_{iji'j'k}$	1, if operation O_{ij} precedes operation $O_{i'j'}$ on machine k; 0, otherwise
C_i	The completion time of job i
C_{max}	The maximum completion time over all jobs (makespan)

Table 4: Variables for MILP-1

The Mixed Integer Linear Programming model that is formulated for the given problem is presented in the following form:

$$
\begin{aligned}
&\min \quad C_{\max} \\
&\text{s.t.} \\
&\sum_{k \in M_j} X_{ijk} = 1 && \forall i \in J, \forall j \in O_i && \text{(1a)} \\
&S_{ijk} + C_{ijk} \leq X_{ijk} \cdot L && \forall i \in J, \forall j \in O_i, \forall k \in M_j && \text{(1b)} \\
&C_{ijk} \geq S_{ijk} + t_{ijk} - (1 - X_{ijk}) \cdot L && \forall i \in J, \forall j \in O_i, \forall k \in M_j && \text{(1c)} \\
&S_{ijk} \geq C_{i'j'k} - (Y_{iji'j'k}) \cdot L && \forall i < i', \forall j \in O_i, \forall j' \in O_{i'}, \forall k \in M_j \cap M_{j'} && \text{(1d)} \\
&S_{i'j'k} \geq C_{ijk} - (1 - Y_{iji'j'k}) \cdot L && \forall i < i', \forall j \in O_i, \forall j' \in O_{i'}, \forall k \in M_j \cap M_{j'} && \text{(1e)} \\
&\sum_{k \in M_j} S_{ijk} \geq \sum_{k \in M_j} C_{i,j-1,k} && \forall i \in J, \forall j \in O_i \setminus O_{if_{(i)}} && \text{(1f)} \\
&C_i \geq \sum_{k \in M_j} C_{i,O_{il_{(i)}},k} && \forall i \in J && \text{(1g)} \\
&C_{\max} \geq C_i && \forall i \in J && \text{(1h)} \\
&X_{ijk} \in \{0,1\} && \forall i \in J, \forall j \in O_i, \forall k \in M_j \\
&S_{ijk} \geq 0 && \forall i \in J, \forall j \in O_i, \forall k \in M_j \\
&C_{ijk} \geq 0 && \forall i \in J, \forall j \in O_i, \forall k \in M_j \\
&C_i \geq 0 && \forall i \in J \\
&Y_{iji'j'k} \in \{0,1\} && \forall i < i', \forall j \in O_i, \forall j' \in O_{i'}, \forall k \in M_j \cap M_{j'}
\end{aligned}
$$

The set of constraints (1a) assign each operation O_{ij} to a single machine. If operation O_{ij} isn't allocated to machine k, then constraint (1b) impose that the start and end times for it on machine k are set to zero. However, if it is assigned, constraint (1c) ensures that the time difference between the start and end times is at least equal to the processing time on machine k. The constraints (1d) and (1e) manage the requirement that operations O_{ij} and $O_{i'j'}$ cannot occur simultaneously on any machine in the set $M_j \cap M_{j'}$. The constraint (1f) guarantees the precedence relationships between the operations of a job, i.e., operation O_{ij} cannot start until operation $O_{i,j-1}$ has finished.

Ultimately, constraints (1g) and (1h) specify the completion times for the last operations of the jobs and the total makespan, respectively.

3.2 Description of MILP-2

MILP-2 is an extension of MILP-1 that enables it to handle the process plan flexibility. As for the MILP-1 we presented the parameters and variables in the following tables.

Indices	**Description**
$i, i' \in J$	Indices for jobs, where J is the set of all jobs.
$j, j' \in O$	Indices for operations, where O is the set of all operations.
$k \in M$	Index for machines, where M is the set of all machines.
O_i	Ordered set of operations of job i, where $O_i \subseteq O$.
M_j	Set of alternative machines on which operation j can be processed, where $M_j \subseteq M$.
$M_j \cap M_{j'}$	Set of machines on which operations j and j' can be processed.
$p_{(i)}$	Process plans of job i ($p_{(i)} \in P_i$).
P_i	Set of alternative process plans for job i.
$O_{ip_{(i)}}$	An ordered set of operations is denoted as $O_{ip_{(i)}}$ in the process plan $p_{(i)}$ for job i, where $O_{ip_{(i)}} = \{f_{(i)}, \ldots, l_{(i)}\}$, and $f_{(i)}$ represents the first element while $l_{(i)}$ represents the last element of $O_{ip_{(i)}}$.
O_i	The set of operations in all process plans of job i ($O_i = \bigcup_{p_{(i)} \in P_i} O_{ip_{(i)}}$).

Table 5: Indices and Sets for MILP-2

Parameters	**Description**
$t_{ip_{(i)}jk}$	The processing time of operation $O_{ip_{(i)}j}$ on machine k.
L	A large number

Table 6: Parameters for MILP-2

Variables	**Description**
$Z_{ip_{(i)}}$	1, if process plan $p_{(i)}$ of job i is selected; 0, otherwise;
$X_{ip_{(i)}jk}$	1, if machine k is selected for operation $O_{ip_{(i)}j}$; 0, otherwise;
$S_{ip_{(i)}jk}$	The starting time of operation $O_{ip_{(i)}j}$ on machine k;
$C_{ip_{(i)}jk}$	The completion time of operation $O_{ip_{(i)}j}$ on machine k;
$Y_{iji'j'k}$	1, if operation O_{ij} precedes operation $O_{i'j'}$ on machine k; 0, otherwise;
C_i	The completion time of job i;
C_{max}	The maximum completion time over all jobs (**makespan**);

Table 7: Variables for MILP-2

The MILP-2 is presented as follows:

$$\min \quad C_{\max}$$

s.t.

$$\sum_{k \in M_j} X_{ip_{(i)}jk} = Z_{ip_{(i)}} \qquad \forall i \in J, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}} \tag{2a}$$

$$\sum_{p_{(i)} \in P_i} Z_{ip_{(i)}} = 1 \qquad \forall i \in J \tag{2b}$$

$$S_{ip_{(i)}jk} + C_{ip_{(i)}jk} \leq (X_{ip_{(i)}jk}) \cdot L \qquad \forall i \in J, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}}, \forall k \in M_j \tag{2c}$$

$$C_{ip_{(i)}jk} \geq S_{ip_{(i)}jk} + t_{ip_{(i)}jk} - (1 - X_{ip_{(i)}jk}) \cdot L \qquad \forall i \in J, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}}, \forall k \in M_j \tag{2d}$$

$$\sum_{p_{(i)} \in P_i} S_{ip_{(i)}jk} \geq \sum_{p_{(i')} \in P_{i'}} C_{i'p_{(i')}j'k} - (Y_{iji'j'k}) \cdot L \qquad \forall i < i', \forall j \in O_{ip_{(i)}}, \forall j' \in O_{i'p(i')}, \forall k \in M_j \cap M_{j'} \tag{2e}$$

$$\sum_{p'_{(i)} \in P_{i'}} S_{i'p'_{(i)}j'k} \geq \sum_{p_{(i)} \in P_i} C_{ip_{(i)}jk} - (1 - Y_{iji'j'k}) \cdot L \qquad \forall i < i', \forall j \in O_{ip_{(i)}}, \forall j' \in O_{i'p'(i')}, \forall k \in M_j \cap M_{j'} \tag{2f}$$

$$\sum_{k \in M_j} S_{ip_{(i)}jk} \geq \sum_{k \in M_j} C_{ip_{(i)}j-1,k} \qquad \forall i \in J, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}} - O_{ip_{(i)}f_{(i)}} \tag{2g}$$

$$C_i \geq \sum_{k \in M_j} C_{ip_{(i)}, O_{ip_{(i)}l_{(i)}}, k} \qquad \forall i \in J, \forall p_{(i)} \in P_i \tag{2h}$$

$$C_{\max} \geq C_i \qquad \forall i \in J \tag{2i}$$

$$Z_{ip_{(i)}} \in \{0,1\} \qquad \forall i \in J, \forall p_{(i)} \in P_i$$

$$X_{ip_{(i)}jk} \in \{0,1\} \qquad \forall i \in J, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}}, \forall k \in M_j$$

$$S_{ip_{(i)}jk} \geq 0 \qquad \forall i \in J, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}}, \forall k \in M_j$$

$$C_{ip_{(i)}jk} \geq 0 \qquad \forall i \in J, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}}, \forall k \in M_j$$

$$Y_{iji'j'k} \in \{0,1\} \qquad \forall i < i', \forall j \in O_i, \forall j' \in O_{i'}, \forall k \in M_j \cap M_{j'}$$

$$C_i \geq 0 \qquad \forall i \in J$$

Constraints (2a) and (2b) ensure the acceptance of a single process plan for job i and the exclusive assignment of operation $O_{ip_{(i)}j}$ to a particular machine. In cases where operation $O_{ip_{(i)}j}$ is not assigned to machine k, constraints (2c) set both the start and finish times on machine k to zero. If assigned, constraints (2d) ensure that the time difference between start and finish on machine k is equal to or greater than the execution time. To satisfy the condition that operations O_{ij} and $O_{i'j'}$ must not overlap in their respective process plans on any machine in the set $M_j \backslash M_{j'}$, constraints (2e) and (2f) are applied. Constraints (2g) maintain the integrity of precedence relationships between job operations, while (2h) determine job completion times, and (2i) determine the makespan.

4 Computational Study

In this section we want to present the computational results obtained by **Özgüven et al.(2010)** **[17]** through the tests conducted on MILP-2. This test problems are generated by **Özbakır et al.** [18], who has created hypothetical FJSPs with different sizes. For every problem she wanted to analyze the solution capability of MILP-2 on the FJSP-PPF, and each problem size corresponds to four distinct FJSPs, owing to the presence of different levels of process plan and routing flexibility. The authors reported, for each problem, the levels of process plan and routing flexibility, and the approach employed to determine the operations, machine alternatives for operations and processing times of operations for each job. In particular the author did a test problem with five jobs, two process plans, six operations and five machines. The two alternative process plan exist with a predetermined set of ordered operations for each job, along with two alternative machines for each operation. The range for the number operations in the process plans is from two to four. They obtained a solution with a Cmax value of 193. The test problems are executed in the mathematical modelling language AIMMS 3.8 and CPLEX 11.0. MILP-2 successfully identifies optimal solutions only for the initial three problems; for remaining case, due to the problem size's increase, the disparity between the optimal LP bound and the best integer solution widens. Finally, one last observation made is that the number of binary variables and constraints in MILP-2 is primarily sensitive to the number of alternative machines.

5 Extensions or Applications

In this section we propose a new model based on **MILP-1** by Özgüven [17] where the efficiency values of individual machines are inputted, aiming to better represent a real-world scenario. Consequently, the makespan is recalculated, taking into account the efficiency of the overall system. Both the implementation of the original model, and our extension, were carried out in Python and subsequently solved using the CPLEX solver, in the particular scenario of 2 jobs, 2 operations for each job and 2 machines, by using the same dataset as the one used in our original paper in order to test our results (optimum $C_{max} = 66$) . Under the assumption that the efficiencies of the two machines are independent of each other, the constraint (**1c**) is modified as follows:

$$C_{ijk} \geq S_{ijk} + \frac{t_{ijk}}{\alpha_k} - (1 - X_{ijk}) \cdot L \quad \forall i \in J, \forall j \in O_i, \forall k \in M_j$$

with $\alpha_k \in (0, 1)$ parameter of efficiency.

6 Conclusions

In this paper, we have reviewed the literature on the flexible job shop scheduling problem (FJSP) and its extensions, focusing on the mathematical models and solution methods proposed by various authors. We have also presented the MILP-1 and MILP-2 models developed by Ozgüven et al. (2010) for the FJSP and the FJSP with process plan flexibility (FJSP-PPF), respectively. We have summarized the main features and results of these models, as well as their advantages and limitations. We implemented MILP-1 in a Python environment in order to solve a small benchmark instance of the flexible job shop widely used by several authors. Then, we formulated and introduced a modified model derived from MILP-1, designed to enhance the efficiency of each machine. Furthermore, we implemented and tested the newly proposed model. As future research directions also suggest to explore other extensions and applications of the FJSP, such as considering environmental impact, stochastic parameters and novel techniques that adopts machine learning and deep learning techniques.

References

[1] A. s. Manne, "On the job-shop scheduling problem," *Operations Research*, vol. 8, pp. 219–223, 1960.

[2] P. D. I. Kacem, S. Hammadi, and P. Borne, "Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic.," *Mathematics and Computers in Simulation*, vol. 60, pp. 245–276, 09 2002.

[3] J. Gao, M. Gen, L. Sun, and X. Zhao, "A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems," *Computers Industrial Engineering*, vol. 53, no. 1, pp. 149–162, 2007.

[4] P. Fattahi, M. Saidi Mehrabad, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 18, no. 3, 2007.

[5] Y. Demir and S. Kürşat İşleyen, "Evaluation of mathematical models for flexible job-shop scheduling problems," *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 977–988, 2012.

[6] A. Ebadi and G. Moslehi, "Mathematical models for preemptive shop scheduling problems," *Computers Operations Research*, vol. 39, no. 7, p. 1605 – 1614, 2012.

[7] Z. Chen, L. Zhang, X. Wang, and P. Gu, "Optimal design of flexible job shop scheduling under resource preemption based on deep reinforcement learning," *Complex System Modeling and Simulation*, vol. 2, pp. 174–185, 06 2022.

[8] E. Birgin, P. Feofiloff, C. Fernandes, E. Melo, M. Oshiro, and D. Ronconi, "A milp model for an extended version of the flexible job shop problem," *Optimization Letters*, 04 2013.

[9] R. Hansmann, T. Rieger, and U. Zimmermann, "Flexible job shop scheduling with blockages," *Mathematical Methods of Operations Research*, vol. 79, 04 2013.

[10] H. Mokhtari, "Scheduling optimization of a stochastic flexible job-shop system with time-varying machine failure rate," *Computers Operations Research*, vol. 61, 03 2015.

[11] L. Yin, X. Li, L. Gao, C. Lu, and Z. Zhang, "A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem, considering productivity, energy efficiency and noise reduction," *Sustainable Computing: Informatics and Systems*, vol. 13, 11 2016.

[12] L. Meng, C. Zhang, X. Shao, and Y. Ren, "Milp models for energy-aware flexible job shop scheduling problem," *Journal of Cleaner Production*, vol. 210, pp. 710–723, 11 2018.

[13] L. Shen, S. Dauzère-Pérès, and J. Neufeld, "Solving the flexible job shop scheduling problem with sequence-dependent setup times," *European Journal of Operational Research*, vol. 265, 08 2017.

[14] X. Zhu, W. Wang, X. Guo, and L. Shi, "A genetic programming-based evolutionary approach for flexible job shop scheduling with multiple process plans," *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pp. 49–54, 2020.

[15] E. Ghaedy-Heidary, E. Nejati, A. Ghasemi, and S. Torabi, "A simulation optimization framework to solve stochastic flexible job-shop scheduling problems—case: Semiconductor manufacturing," *Computers Operations Research*, vol. 163, p. 106508, 12 2023.

[16] Z. Liu, J. Zha, J. Yan, Y. Zhang, T. Zhao, Q. Cheng, and C. Cheng, "An improved genetic algorithm with an overlapping strategy for solving a combination of order batching and flexible job shop scheduling problem," *Engineering Applications of Artificial Intelligence*, vol. 127, 2024.

[17] C. Özgüven, L. Özbakir, and Y. Yavuz, "Mathematical models for job-shop scheduling problems with routing and process plan flexibility," *Applied Mathematical Modelling*, vol. 34, no. 6, p. 1539 – 1548, 2010.

[18] L. Özbakır, ", modeling, analyzing and solution of multiple objective flexible job shop scheduling problems by using meta-heuristic algorithms," *Ph.D. Dissertation, Erciyes University, Social Sciences Institute, Kayseri, Turkey*, 2004.