10.24425/acs.2025.156308

Archives of Control Sciences
Volume 35(LXXI), 2025
No. 3*, pages 485–504*

A novel heuristic method for flexible job-shop scheduling problem with sequence-dependent setup time, transportation time and machine efficiency constraints

Jolanta KRYSTEK and Adam GEPPERT

This paper presents a novel heuristic approach to the Flexible Job-Shop Scheduling Problem (FJSP), incorporating sequence-dependent setup and transportation times, as well as machine efficiency constraints. The objective of the optimization was to minimize the total makespan of the job scheduling process. To solve this problem, a computational tool was developed, which implemented a new heuristic method based on a sequential algorithm, alongside selected priority assignment rules (FIFO, LIFO, LPT, SPT, EDD, LWR). The effectiveness of the generated schedules was evaluated using multiple criteria. Comparative numerical results were presented for various problem parameters and priority rules, demonstrating the performance and advantages of the proposed approach.

Key words: flexible job-shop scheduling problem (FJSP), heuristic method, sequence-dependent setup times, transportation times, machine efficiency, dispatch rules

1. Introduction

In modern manufacturing systems, the demand for greater flexibility, efficiency, and adaptability has significantly increased due to evolving market conditions, dynamic customer requirements, and the continuous pursuit of cost reductions. Industrial environments are facing the challenge of maintaining high

J. Krystek (corresponding author, e-mail: jolanta.krystek@polsl.pl) is with Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland.

A. Geppert (e-mail: adam@websited.pl) is with Websited sp. z o.o., Ligonia 1/1, 44-100 Gliwice, Poland.

This work has been supported by 02/040/BK_25/1070 grant.

Received 7.05.2025. Revised 7.09.2025.

production efficiency while simultaneously adapting to unforeseen changes such as urgent job arrivals, machine failures, or fluctuations in resource availability. Consequently, the need for advanced scheduling techniques that can effectively handle such complexities has become more pressing. Scheduling optimization plays a critical role in ensuring that production systems operate smoothly and meet tight deadlines while minimizing costs and resource consumption.

The Job-Shop Scheduling Problem (JSP) [12] is a well-known combinatorial optimization problem that has been the focus of extensive research due to its practical significance in manufacturing systems. The traditional JSP model assumes a fixed allocation of resources, where each job must be processed on a specific machine. However, as industrial requirements evolve, the limitations of this rigid approach become apparent. The Flexible Job-Shop Scheduling Problem (FJSP) [3] emerged as a natural extension of the JSP, offering greater flexibility in resource allocation. As a variation of the classic job-shop problem, FJSP removes the limitation of resource uniqueness by allowing each operation to be processed on multiple machines. This flexibility not only enhances the adaptability of the production process but also brings it closer to real-world manufacturing environments, where machine capabilities may vary, and resource allocation must be dynamic.

The flexibility provided by FJSP offers significant advantages in terms of efficiency and resilience. In addition to improving production scheduling, FJSP also better accommodates the need for responsiveness in real-time operations. As a result, it is particularly suited to address the challenges posed by urgent job arrivals or sudden changes in the market environment [18]. This makes FJSP an ideal model for industries that must operate in highly dynamic settings, where schedules need to be continuously adjusted to meet new demands and constraints.

In recent years, the FJSP has garnered considerable attention in literature, with numerous studies exploring various extensions and refinements of the model. Researchers have incorporated additional constraints, such as time lag, availability, batching, blocking, setup times, transportation and many others, to more accurately represent the complexities of real-world manufacturing systems [5]. Moreover, multi- and many-objective optimization approaches and energy-efficient scheduling methods have been proposed to further improve the sustainability and cost-effectiveness of production processes. However, many of these studies focus on solving specific sub-problems, with optimization techniques tailored to particular constraints or application domains.

This paper addresses this gap by presenting a novel heuristic method for solving the Flexible Job-Shop Scheduling Problem with constraints. The proposed approach integrates sequence-dependent setup times, transportation times, and machine efficiency constraints into a single optimization framework (FJSP-SDSTTTME), which, to the best of our knowledge, has not yet been described in the existing literature. By doing so, it aims to provide a more realistic representation of the complexities involved in job scheduling within industrial environments. The study offers a general approach that balances the various constraints and seeks to minimize the total makespan, thus improving the overall efficiency of the scheduling process. The computational experiments conducted in this work demonstrate the effectiveness of the proposed method, providing new insights into the potential of FJSP for real-world applications.

2. Formulation of the FJSP-SDSTTTME

The Flexible Job-Shop Scheduling Problem with sequence-dependent setup times, transportation times, and machine efficiency constraints can be formalized as follows.

The problem describes a set of n jobs $J = \{J_1, J_2, \ldots, J_n\}$ and a set of m machines $M = \{M_1, M_2, \ldots, M_m\}$. Each job J_i consists of a set of o operations $O^i = \{O^i_1, O^i_2, \ldots, O^i_o\}$, each of which has a duration $p^i_k > 0$ and must be performed within a specified time window, between the ready-time r_i and due-time d_i. Each operation O^i_k requires a machine $M(O^i_k)$ from a subset of compatible machines $M(O^i_k) \subseteq M_{CM}$. The operation must be performed on the selected machine $M(O^i_k)$ without interruption. Furthermore, for two consecutive operations O^i_k and O^i_{k+1} the following condition must hold:

$$M(O^i_k) \neq M(O^i_{k+1}). \tag{1}$$

The machine efficiency factor, denoted as $MEF(O^i_k)$, represents the effectiveness of a machine in executing an operation O^i_k. The machine efficiency $ME(O^i_k)$ is a value between 0 and 1, where 1 corresponds to 100% efficiency, meaning the operation is completed in its base duration p'^i_k. If the machine is less efficient (i.e., $ME\left(O^i_k\right) < 1$), the operation's duration p^i_k is increased proportionally to the decrease in efficiency.

$$MEF(O^i_k) = \left(1 + \left(1 - ME(O^i_k)\right)\right), \tag{2}$$

$$p^i_k = p'^i_k * MEF\left(O^i_k\right). \tag{3}$$

The start time S^i_k denotes the moment when an operation O^i_k can begin using the assigned resources, while the end time C^i_k represents the completion time of this operation, where:

$$C^i_k = S^i_k + p^i_k, \tag{4}$$

$$S^i_k + p^i_k \leqslant S^j_h, \tag{5}$$

$$S_h^j + p_h^j \leqslant S_k^i . \tag{6}$$

The time required to set up a machine before processing a new operation depends on the characteristics of the previous operation and is known as sequence-dependent setup time $ST_{jh}^{ik}\big|_{M(O_k^i)}$, which must be accounted for when scheduling consecutive operations [2]. The setup time impacts the start time of the subsequent operation, as the machine may need to be reconfigured before it can process the next operation.

When operations are performed on different machines, there is also a transportation phase between operations that occurs after one operation is completed and before the next one begins. This transportation time $TT_{M(O_{k-1}^i)}^{M(O_k^i)}$ depends on the distance and handling time between machines and must be included in the scheduling process [13]. Just like setup times, transportation times are accounted for when determining the start time of a subsequent operation.

If operation O_k^i is to be executed directly after operation O_h^j on the same machine $M(O_k^i) = M(O_h^j)$, a sequence-dependent setup time $ST_{jh}^{ik}\big|_{M(O_k^i)}$ and transportation time $TT_{M(O_{k-1}^i)}^{M(O_k^i)}$ directly impact the start time $S_k^i\big|_{M(O_k^i)}$ of operation O_k^i on the respective machine. The relationship is expressed as:

$$S_k^i\big|_{M(O_k^i)} = \max\left(\left(C_h^j + ST_{jh}^{ik}\big|_{M(O_k^i)}\right), \left(C_{k-1}^i + TT_{M(O_{k-1}^i)}^{M(O_k^i)}\right)\right) \tag{7}$$

The scheduling is feasible if, for any two consecutive operations O_k^i and O_{k+1}^i of the same job, condition (1) holds, and for any two operations O_k^i and O_h^j, where $M(O_k^i) = M(O_h^j)$, constraints (4) or (5) are satisfied. The objective is to find a schedule that minimizes the makespan $C_{\max}$.

$$C_{\max} = \max_{i=1,..,n} \left(C_k^i\right) . \tag{8}$$

Additionally, the problem requires the consideration of the following assumptions:

- All machines are available at time zero.
- Each machine can process only one operation at a time.
- Operations are non-preemptive.
- Operations of the same job have precedence relationships.
- Input and output buffers of machines are unlimited.
- Setup times may vary depending on the type of operation.

- Machine setup time from task A to task B can differ from the reverse setup time.
- Machines cannot perform other tasks during setup.
- Transportation resources are not limited.
- Transportation operations cannot be interrupted.
- Loading and unloading times in the transportation process are neglected.
- The first operation of the first job on a machine is placed directly into the input buffer, without requiring transportation.
- Machines are properly prepared for the first operations of the first jobs on them, eliminating the need for setup at the start of processing.

3. A new heuristic method for the FJSP-SDSTTTME

The Job-Shop Scheduling Problem is well-known in literature as an NP-hard problem and its complexity increases significantly when extended to the Flexible Job-Shop Scheduling Problem. When additional constraints, such as sequence-dependent setup times, transportation times, and machine efficiency limitations, are incorporated, the problem becomes even more challenging. These added layers of complexity necessitate the consideration of new solution approaches, as traditional optimization techniques may struggle to provide feasible or efficient solutions in reasonable time.

Given the increased complexity of the problem, it would be natural to consider approaches like Mixed-Integer Linear Programming (MILP) [11] or Genetic Algorithms (GAs) [4]. However, these methods are typically computationally expensive, especially for large-scale instances, and may not be suitable for real-time or practical industrial applications where rapid solutions are required. MILP formulations, while providing exact solutions, often face scalability issues, particularly when dealing with numerous constraints and decision variables [16]. Similarly, while GAs offer flexibility and adaptability, they are heuristic in nature and may require extensive parameter tuning to achieve satisfactory results [14]. In contrast, priority rules allow for lightweight, interpretable scheduling that can quickly construct feasible solutions, even in highly constrained or fragmented search spaces. This makes them especially effective in contexts where identifying admissible solutions is difficult and where decision-making must occur within strict time limits.

To address these challenges, a new heuristic approach was developed based on a sequential algorithm combined with selected priority rules [9]. This method was chosen for its ability to efficiently handle the constraints introduced by the

FJSP-SDSTTTME while maintaining a balance between solution quality and computational efficiency. The sequential approach, in conjunction with simple yet effective priority rules, allows for a flexible and robust scheduling framework that can be tailored to specific industrial needs. This heuristic method provides a practical solution for industries requiring fast, approximate scheduling results, even in the face of complex, real-world constraints.

The algorithm is executed iteratively, processing discrete time intervals corresponding to events, such as the arrival of jobs into the system or the completion of operations, and continues until all jobs are scheduled. Each iteration begins by handling the job that has become available for machine allocation. If the job still has pending operations, it is added to the queue of waiting jobs. Subsequently, the algorithm checks whether any available machine type has pending jobs awaiting assignment. If such jobs exist, the pre-assignment process is initiated. This process is repeated for as many pending jobs as there are machines of the respective type available at that specific time.

The pre-assignment process starts by resolving conflicts in the job queue based on the selected priority rule. A conflict arises when more than one job is awaiting to start on a machine at the same time. Six priority rules have been implemented to resolve these conflicts [8, 15]:

- FIFO (First In First Out) – the job that entered the queue first is assigned to the machine.
- LIFO (Last In First Out) – the job that entered the queue last is assigned to the machine.
- SPT (Shortest Processing Time) – priority is given to the job that requires the shortest processing time on the machine at that moment.
- LPT (Longest Processing Time) – priority is given to the job that requires the longest processing time on the machine at that moment.
- LWR (Least Work Remaining) – conflicts are resolved in favor of the job with the least remaining work (excluding already completed operations).
- EDD (Earliest Due Date) – priority is given to the job with the earliest predicted completion time, denoted as d_i, which is the time when the job is expected to finish after all operations are completed, starting from time 0.

In the event that no clear winner emerges from the above priority rules, a secondary custom rule is applied, where the job with the lower identification number is assigned priority.

Following the resolution of conflicts, the sequence proceeds to select a machine from the available machines of the chosen type. The primary selection criterion is to choose the machine with the highest machine efficiency coefficient *ME*. If multiple machines have the same *ME* value, the final decision is made based on a secondary rule, where the machine with the lower identification number is given priority.

The subsequent step is to determine whether the selected job is the first operation to be executed on the selected machine. If this is the case, the setup time is set to $ST^{ik}_{jh}\big|_{M(O^i_k)} = 0$, in accordance with the assumptions of the problem. Otherwise, the procedure to determine the setup time begins, which is calculated as the first argument of the max function (7) when determining the start time for the operation on the selected machine $S^i_k\big|_{M(O^i_k)}$.

The sequence then proceeds to the analysis of transportation times. It begins by checking whether the operation to be assigned is the first operation of the job. If so, transportation time does not apply, and the transportation time is set to $TT^{M(O^i_k)}_{M(O^i_{k-1})} = 0$, according to the problem assumptions. If the operation is not the first, the procedure to determine the end of the transportation process begins, and this value is considered as the second argument of the max function (7) when calculating the start time for the operation on the machine $S^i_k\big|_{M(O^i_k)}$.

Based on the results of the previous two steps, the start time for the operation on the selected machine $S^i_k\big|_{M(O^i_k)}$ is calculated using the formula (7). Once the start time is determined, the operation can be assigned to the machine, and the machine's availability, along with the job's next available time, is updated based on the calculated completion time C^i_k.

4. Evaluation of the schedule

In scheduling problems, the effectiveness of a solution is typically assessed using various performance metrics that quantify different aspects of the scheduling process. For the FJSP-SDSTTTME, these metrics are crucial in evaluating the quality of the schedule generated by the proposed heuristic method. The primary objective is to minimize overall processing time, optimize resource utilization, and reduce operational delays, thereby enhancing the efficiency of the scheduling process. The following key criteria are employed to assess the quality of the schedules produced by the heuristic method.

4.1. Completion time of all jobs

One of the fundamental optimality criteria in the considered problem is the completion time of all jobs, denoted as $C_{\max}$, also referred to as the system completion time. This criterion defines the moment at which the last job in the system is completed. Minimizing this metric is essential for improving production efficiency, as a reduction in total job completion time directly translates to lower operational costs and better resource utilization [12]. In industrial scheduling, a lower makespan indicates improved throughput and enhanced production system performance, making it a critical objective in optimization strategies. The completion time of all jobs is mathematically defined as follows:

$$C_{\max} = \max_{i=1,\ldots,n} \left(C_i\right), \tag{9}$$

where C_i represents the completion time of the i-th job, and n denotes the total number of jobs in the system. Therefore, the completion time of all jobs corresponds to the maximum completion time among all scheduled jobs.

4.2. Job flow time

The job flow time F_i is a vital metric for evaluating scheduling efficiency, as it quantifies the duration a job remains in the system from the moment it becomes available for processing until its completion. More formally, flow time is defined as the difference between the completion time C_i and the ready-time r_i, which represents the earliest possible start time for a given job [10]:

$$F_i = C_i - r_i\,. \tag{10}$$

Prolonged flow times can have a cascading effect on the entire scheduling process, delaying subsequent jobs and affecting overall resource utilization. Consequently, minimizing flow time is a fundamental objective in production scheduling, as it leads to reduced waiting times, improved responsiveness, and enhanced resource balancing.

4.3. Job waiting time

The job waiting time W_i refers to the period during which a job is present in the system but remains idle, awaiting the availability of the necessary resources. Unlike flow time, which includes both active and idle periods, waiting time specifically measures the inefficiencies introduced by resource unavailability. It is computed as the difference between the job flow time F_i and the cumulative processing time of all operations assigned to that job [7, 10]:

$$W_i = F_i - \sum_{k=1}^{o} O_k^i\,. \tag{11}$$

Excessive waiting times can lead to reduced system efficiency, production bottlenecks, and increased lead times. As a result, minimizing job waiting time is essential for improving system throughput, achieving better synchronization between jobs, and optimizing resource allocation.

4.4. Machine idle time

The machine idle time I represents the total period during which machines are not engaged in any operation within a given schedule. This metric directly measures the degree of resource underutilization, making it a crucial indicator of scheduling efficiency. Machine idle time is calculated individually for each machine as the difference between the completion time of the last assigned operation E_i and the cumulative processing duration of all operations executed on that machine O_i. If a machine has not been assigned any operations, its idle time is considered zero and is excluded from the final evaluation [17]:

$$I = \sum_{i=1}^{m} \max\left(0,\ (E_i - O_i)\right). \tag{12}$$

5. Experimental study and computational results

In order to evaluate the performance of the proposed heuristic for the FJSP-SDSTTTME, a selection of benchmark test instances from the literature was utilized, specifically those developed by Adams et al. [1], Fisher et al. [7], and Demirkol et al. [6]. These instances, originally designed for the classical Job Shop Scheduling Problem, were systematically adapted to align with the newly formulated FJSP-SDSTTTME model. To transform the JSP instances into FJSP instances, the original number of machines was interpreted as the number of machine types. The actual number of machines available for each type was then randomly sampled from a normal distribution with values ranging from 2 to 4, introducing an additional layer of flexibility in machine assignments. Furthermore, to better reflect the complexities encountered in real-world manufacturing environments, additional constraints such as sequence-dependent setup times and transportation times were incorporated. The setup and transportation times were randomly drawn from a normal distribution within the range of 1 to 5, simulating the variability in industrial scheduling scenarios.

To establish a baseline for performance evaluation, an initial experimental configuration was implemented in which all machines were assumed to operate at full efficiency. Specifically, the machine efficiency coefficient, denoted as $ME(O_k^i)$, was set to 1 for all machines, indicating optimal operational conditions where no additional delays were introduced due to machine efficiency.

Computational experiments were conducted for each of the proposed priority rules to assess their impact on scheduling performance. These rules, as described in Section 3, were applied sequentially, allowing for an isolated evaluation of their individual effectiveness in solving the scheduling problem. Each experiment systematically examined the influence of distinct rule-based decision-making strategies on key scheduling performance indicators.

The FT06 instance from Fisher [7] was selected as a representative case study for detailed analysis. For this instance, the FIFO priority rule, which yielded the best performance in terms of makespan minimization, was chosen for a more in-depth visualization of the scheduling process. The following graphical representations (Figs. 1–4) were generated for this instance.

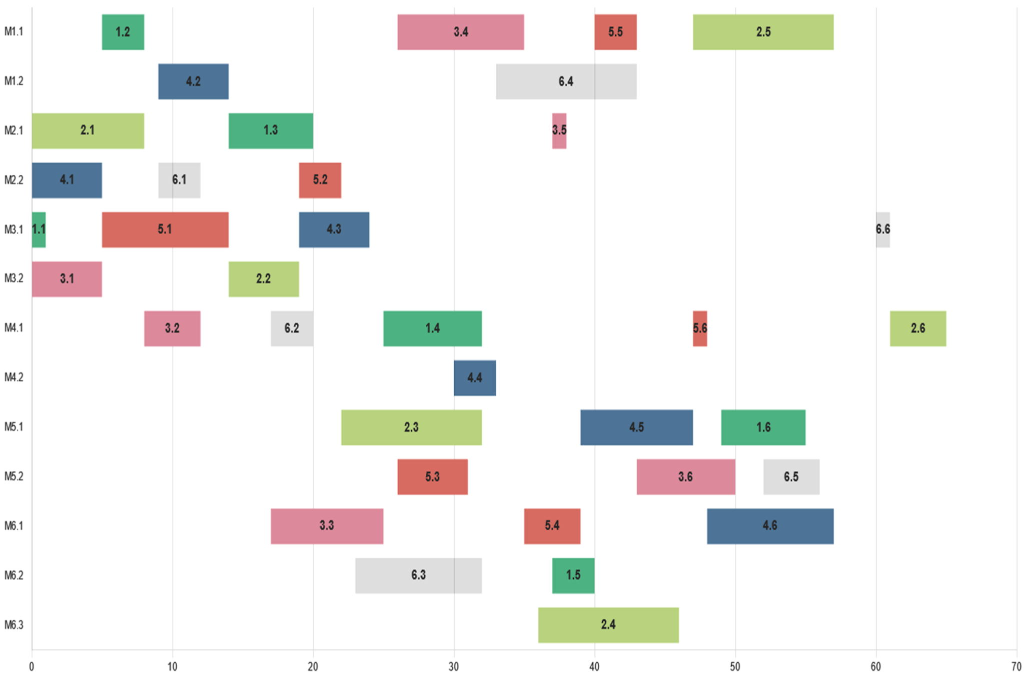

Figure 1: Gantt chart illustrating job scheduling for the FT06 instance (FIFO) with machines operating at full efficiency ($ME(O_k^i) = 1$)

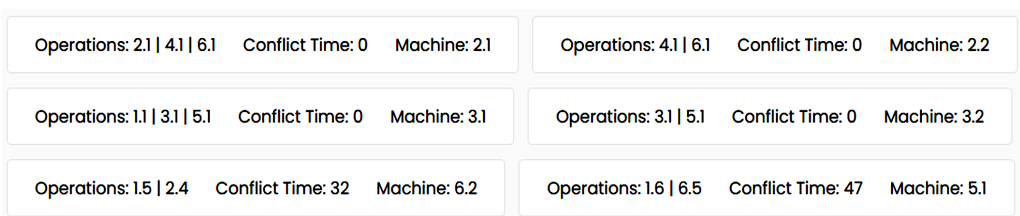

Figure 2: List of job conflicts in the FT06 instance (FIFO) with machines operating at full efficiency ($ME(O_k^i) = 1$)

Figure 3: Gantt chart representing sequence-dependent setup times for the FT06 instance (FIFO) with machines operating at full efficiency $(ME(O_k^i) = 1)$

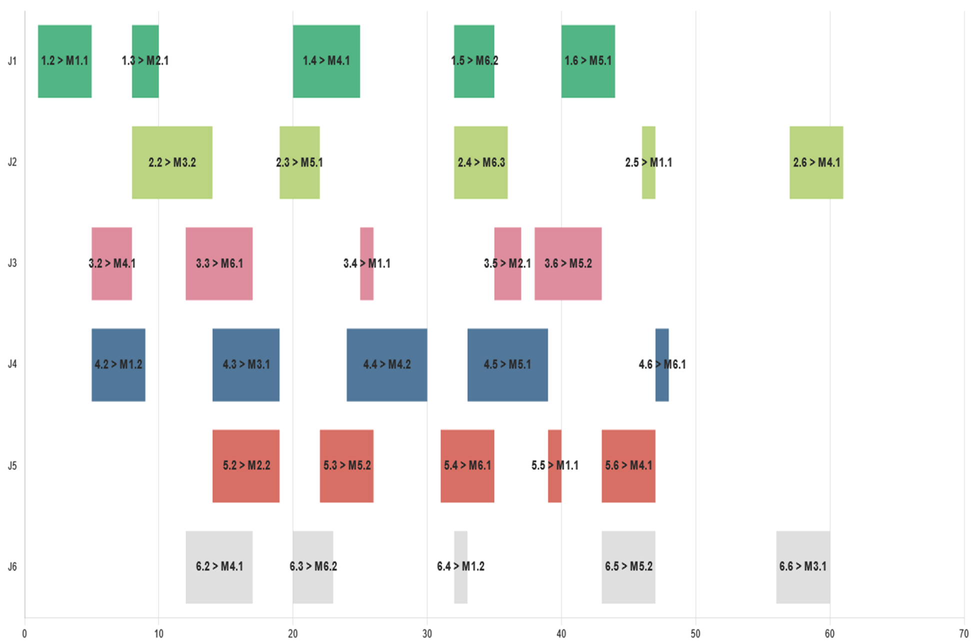

Figure 4: Gantt chart illustrating transportation operations in the FT06 instance (FIFO) with machines operating at full efficiency $(ME(O_k^i) = 1)$

These visualizations provide valuable insights into the dynamics of the scheduling process, demonstrating how job sequencing, setup dependencies, and transportation constraints influence the final schedule. The results for all test instances were subsequently structured in tabular format (Table 1), enabling a comprehensive comparative analysis of the heuristic's efficiency under various operational conditions.

Table 1: Job assignment and routing for the FT06 instance (FIFO)

Job	Job routing	r_i	d_i
1	M3(1) - M1(3) - M2(6) - M4(7) - M3(6) - M5(6)	0	0
2	M2(8) - M3(5) - M5(10) - M6(10) - M1(10) - M4(4)	0	0
3	M3(5) - M4(4) - M6(8) - M1(9) - M2(1) - M5(7)	0	0
4	M2(5) - M1(5) - M3(5) - M4(3) - M5(8) - M6(9)	0	0
5	M3(9) - M2(3) - M5(5) - M6(4) - M1(3) - M4(1)	0	0
6	M2(3) - M4(3) - M6(9) - M1(10) - M5(4) - M3(1)	0	0

In the second phase of the computational experiments, the machine efficiency factor was introduced to simulate more realistic operational conditions, reflecting the performance variations commonly observed in industrial manufacturing environments. Specifically, for each machine type, the efficiency coefficient for the first machine was set to $ME(O^i_k) = 1$, indicating maximum efficiency. The efficiency for each subsequent machine was systematically reduced by 0.25, with a lower bound constraint ensuring that no machine operated with an efficiency below 0.5. This adjustment was designed to model the effects of wear, aging, and operational degradation that occur over time in real-world production systems, where machines do not consistently maintain their optimal performance levels. Following this modification, the same set of benchmark test instances was reevaluated under the updated efficiency constraints. The proposed heuristic was again applied for each of the priority rules, and its performance was assessed based on key scheduling metrics.

As in the first phase, the FT06 instance from Fisher [7] was analyzed in detail for the previously chosen FIFO priority rule. The following visualizations (Figs. 5–8) were generated to illustrate the impact of machine efficiency degradation.

These results, systematically presented in tabular format (Table 2, Table 3), illustrate the impact of varying machine efficiency on scheduling outcomes, with a particular focus on makespan minimization. The comparative analysis of the first and second phases highlights the effects of machine efficiency degradation on scheduling flexibility, job conflicts, and overall resource utilization.

Table 2: Comparison of scheduling performance across different priority rules under base machine efficiency conditions ($ME(O_k^i) = 1$)

Instance	Size	Priority rule	Cmax	Fmax	Favg	Wmax	Wavg	I
FT06	$6 \times 6 \times 6$	FIFO	65	65	56	31	23.17	395
		LIFO	74	74	58.67	36	25.83	390
		SPT	73	73	57	26	24.17	358
		LPT	71	71	56.83	31	24	400
		EDD	74	74	57.5	39	24.67	390
		LWR	74	74	57.5	39	24.67	390
FT10	$10 \times 10 \times 10$	FIFO	742	742	591.1	145	80.2	8283
		LIFO	794	794	584	144	73.1	7327
		SPT	787	787	586.9	149	76	7287
		LPT	729	729	613.3	196	102.4	8076
		EDD	849	849	580.8	194	69.9	7443
		LWR	840	840	580.3	185	69.4	7177
FT20	$20 \times 5 \times 5$	FIFO	777	777	577.15	505	321.7	1670
		LIFO	768	768	489.45	531	234	1946
		SPT	812	812	450	437	194.55	1568
		LPT	732	732	554.2	527	298.75	1568
		EDD	813	813	453.4	438	197.95	1963
		LWR	845	845	448.65	470	193.2	1604
ABZ9	$20 \times 15 \times 15$	FIFO	526	526	458.05	133	85.95	11741
		LIFO	653	653	448.15	196	76.05	11330
		SPT	616	616	459.7	163	87.6	12254
		LPT	538	538	458.2	199	86.1	11820
		EDD	650	650	452.4	193	80.3	11950
		LWR	650	650	452.25	193	80.15	11953
RCMAX_30_15_1	$30 \times 15 \times 15$	FIFO	2500	2500	1849.8	812	403.2	32699
		LIFO	2699	2699	1853.93	1167	407.33	36883
		SPT	2833	2833	1821.3	1045	374.7	35780
		LPT	2327	2327	1879.67	970	433.07	35481
		EDD	2582	2582	1854.57	751	407.97	36197
		LWR	2631	2631	1848.97	898	402.37	36299
RCMAX_40_15_5	$40 \times 15 \times 15$	FIFO	2776	2776	2321.05	1254	813.33	44145
		LIFO	3526	3526	2242.45	1958	734.73	50400
		SPT	3267	3267	2104.05	1591	596.33	45552
		LPT	3540	3540	2388.53	2206	880.8	48827
		EDD	3889	3889	2159.23	2016	651.5	51321
		LWR	3753	3753	2172.28	1880	664.55	59189
RCMAX_50_15_3	$50 \times 15 \times 15$	FIFO	3073	3073	2486.86	1453	898.96	41953
		LIFO	4201	4201	2333.62	2594	745.72	53582
		SPT	4001	4001	2300.44	2290	712.54	45096
		LPT	3890	3890	2511.24	2473	923.34	63635
		EDD	4041	4041	2318.74	2177	730.84	51524
		LWR	4140	4140	2314.1	2280	726.2	61044

Table 3: Comparison of scheduling performance across different priority rules under variable machine efficiency conditions $\left(ME(O_k^i) \in \langle 0.5; 1\rangle\right)$

Instance	Size	Priority rule	Cmax	Fmax	Favg	Wmax	Wavg	I
FT06	$6\times6\times6$	FIFO	75	75	60.67	35	27.83	461
		LIFO	83	83	61	36	28.17	416
		SPT	76	76	59.67	36	26.83	411
		LPT	73	73	59.83	36	27	471
		EDD	76	76	59.67	36	26.83	377
		LWR	76	76	59.67	36	26.83	377
FT10	$10\times10\times10$	FIFO	813	813	671.1	237	160.2	9052
		LIFO	886	886	653.5	231	142.6	8289
		SPT	825	825	643	194	132.1	7828
		LPT	862	862	690.5	263	179.6	9313
		EDD	861	861	646.7	206	135.8	7819
		LWR	934	934	663.5	279	152.6	8714
FT20	$20\times5\times5$	FIFO	892	892	646.85	643	391.4	1834
		LIFO	789	789	535.45	541	280	1767
		SPT	888	888	527.15	513	271.7	1870
		LPT	848	848	653.6	645	398.15	2071
		EDD	879	879	515.15	504	259.7	2111
		LWR	925	925	504.3	550	248.85	2048
ABZ9	$20\times15\times15$	FIFO	628	628	514.55	226	142.45	13065
		LIFO	705	705	516.65	248	144.55	12586
		SPT	641	641	515.75	224	143.65	12968
		LPT	609	609	514.6	237	142.5	12529
		EDD	675	675	519.55	260	147.45	13302
		LWR	675	675	518.65	260	146.55	12848
RCMAX_30_15_1	$30\times15\times15$	FIFO	2762	2762	2121.7	969	675.1	38140
		LIFO	2756	2756	2123.83	1327	677.23	43210
		SPT	3038	3038	2061.07	1224	614.47	39509
		LPT	2810	2810	2138.33	1205	691.73	38539
		EDD	2949	2949	2047.33	1242	600.73	40471
		LWR	3029	3029	2058.57	1322	611.97	41434
RCMAX_40_15_5	$40\times15\times15$	FIFO	3132	3132	2546.03	1767	1038.3	42165
		LIFO	3471	3471	2439.43	1782	931.7	48204
		SPT	3634	3634	2414.1	1945	906.38	46597
		LPT	3956	3956	2637.93	2449	1130.2	60318
		EDD	3564	3564	2407.07	1840	899.35	45520
		LWR	3638	3638	2412.35	1949	904.63	52017
RCMAX_50_15_3	$50\times15\times15$	FIFO	3344	3344	2786.38	1806	1198.48	40860
		LIFO	4001	4001	2755.64	2227	1167.74	55155
		SPT	3656	3656	2660.84	2131	1072.94	42144
		LPT	4563	4563	3001.24	3053	1413.34	63394
		EDD	4611	4611	2727.32	2747	1139.42	60937
		LWR	4042	4042	2688.34	2247	1100.44	55132

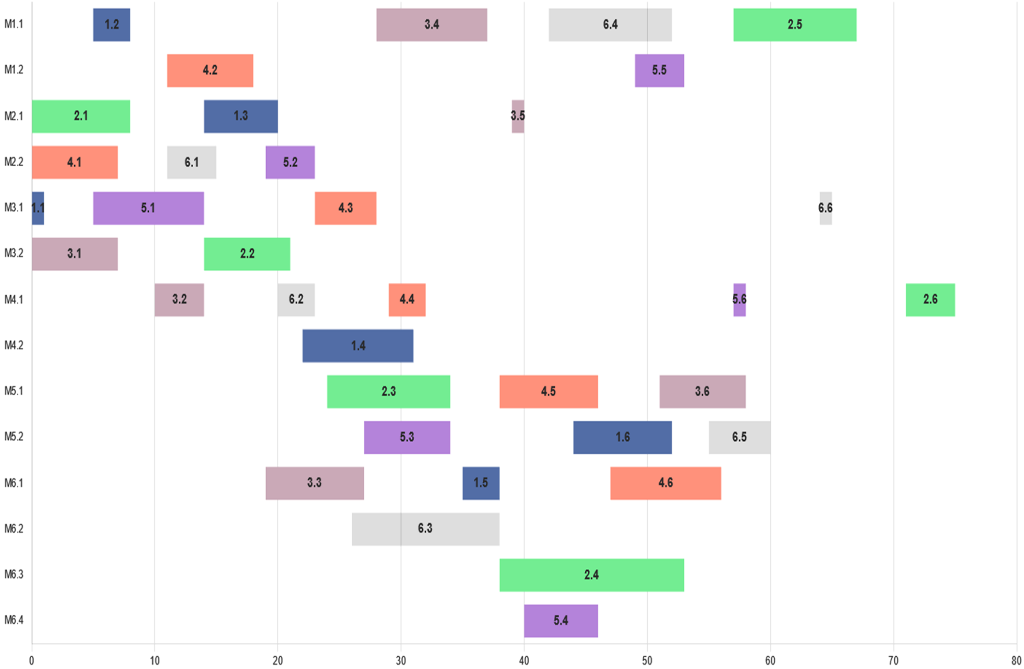

Figure 5: Gantt chart illustrating job scheduling for the FT06 instance (FIFO) with variable machine efficiency $\left(ME(O_k^i) \in \langle 0.5; 1\rangle\right)$

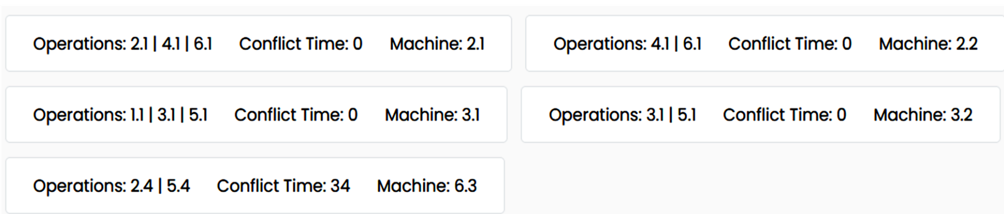

Figure 6: List of job conflicts in the FT06 instance (FIFO) with variable machine efficiency $\left(ME(O_k^i) \in \langle 0.5; 1\rangle\right)$

6. Computational results analysis and conclusion

The analysis of the performance of various priority rules in solving the FJSP-SDSTTTME was conducted across two experimental phases: one with full machine efficiency ($ME(O_k^i) = 1$) and the other with decreasing machine efficiency $\left(ME(O_k^i) \in \langle 0.5; 1\rangle\right)$. This study provides critical insights into the influence of machine performance on scheduling outcomes.

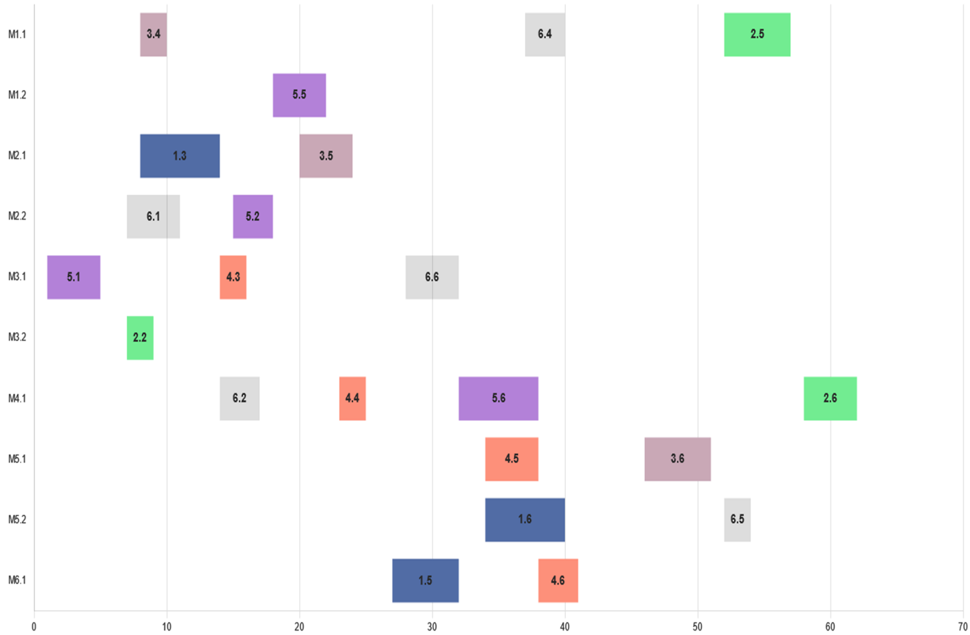

Figure 7: Gantt chart representing sequence-dependent setup times for the FT06 instance (FIFO) with variable machine efficiency $\left(ME(O_k^i) \in \langle 0.5; 1\rangle\right)$

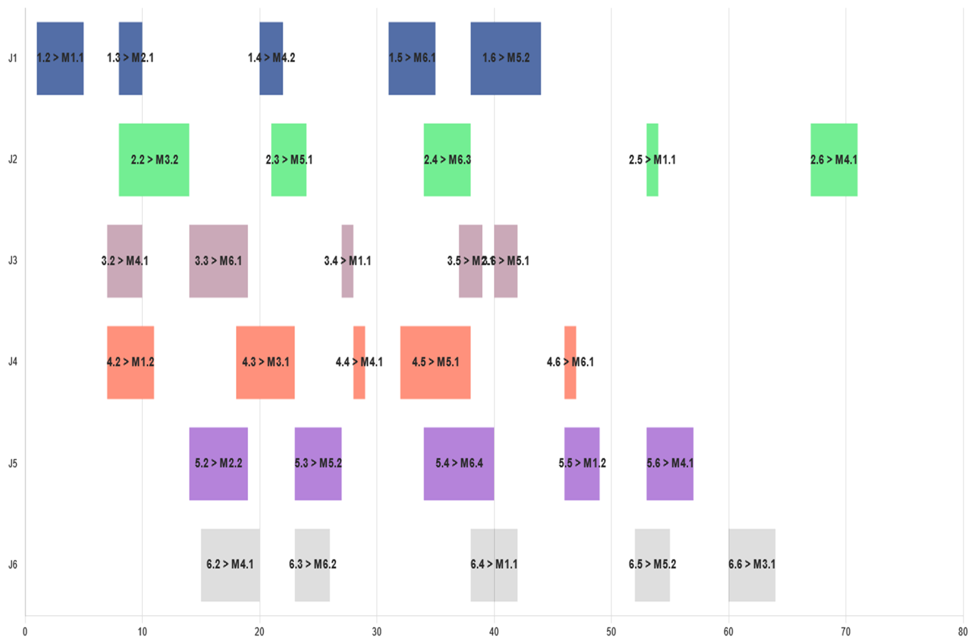

Figure 8: Gantt chart illustrating transportation operations in the FT06 instance (FIFO) with variable machine efficiency $\left(ME(O_k^i) \in \langle 0.5; 1\rangle\right)$

In the first phase, where all machines operated at full efficiency, the FIFO priority rule demonstrated superior performance in minimizing makespan. FIFO consistently outperformed other priority rules by maintaining a relatively balanced distribution of workload across machines. It achieved the lowest makespan in most instances. Other rules, such as LIFO, SPT, and LPT, produced competitive results but were less consistent. These rules often led to higher makespans and more unbalanced workload distributions. Although SPT and LPT were effective in minimizing completion time for individual jobs, they occasionally resulted in resource underutilization or bottlenecks, which caused delays and overall inefficiencies in the schedule.

The second phase of the experiments, where machine efficiency progressively decreased for each subsequent machine, highlighted the significant impact of machine degradation on scheduling performance. As machine efficiency declined, the performance of all priority rules was adversely affected. However, FIFO remained the most effective rule, although with a noticeable increase in makespan. For instance, the makespan for FIFO increased from 65 in the first phase to 75 in the second phase for FT06. While this increase was substantial, it was smaller than the performance degradation observed for other priority rules.

It is worth noting, however, that the consistent superiority of FIFO may be partially influenced by the structure of the test data or specific characteristics of the model. Future studies could apply statistical analyses to verify the robustness of this outcome and confirm whether it generalizes across broader problem classes.

The reduction in machine efficiency had a profound effect on scheduling performance, emphasizing the critical role of machine degradation in real-world manufacturing environments. As efficiency deteriorated, priority rules that had been effective in the first phase experienced significant performance declines. This was particularly evident in instances with a larger number of machines, where the machines were less able to maintain their initial processing speeds. The reduction in machine efficiency led to increased scheduling delays, and these rules produced less balanced schedules, resulting in longer completion times and overall lower system performance.

The rules that had performed poorly in the first phase, such as LWR and EDD, continued to exhibit suboptimal performance in the second phase. These rules, which are designed to optimize based on earliest due dates or remaining workload, did not offer significant advantages under conditions of efficiency degradation. In fact, their makespans further increased, and their ability to optimize resource utilization was diminished. Their performance proved to be less resilient to fluctuations in machine capabilities, thereby reinforcing the importance of selecting priority rules that can withstand operational variances and adapt to the changing performance characteristics of machines.

7. Summary

This paper presents a heuristic algorithm designed to solve the Flexible Job-Shop Scheduling Problem with sequence-dependent setup times, transportation times and machine efficiency constraints, a complex and realistic extension of the classical Job-Shop Scheduling Problem. The proposed heuristic has demonstrated exceptional efficiency in addressing the FJSP-SDSTTTME, consistently producing high-quality results in a relatively short computational time. This efficiency is especially remarkable when considering the added complexity of the three constraints that are commonplace in real-world manufacturing environments.

A major contribution of this study lies in the incorporation of machine efficiency as a dynamic and influential factor within the scheduling model. Our computational experiments reveal the substantial impact of machine efficiency on scheduling performance. In particular, the model accounts for machine performance degradation, which is represented by a decreasing machine efficiency coefficient. This addition highlights the critical role of machine degradation in shaping scheduling outcomes, particularly in terms of makespan and workload distribution. The results underscore the importance of factoring in efficiency degradation when designing scheduling algorithms, as well as the need for adaptive scheduling strategies that can accommodate these fluctuations in machine performance.

In addition to the integration of machine efficiency, the proposed heuristic algorithm has proven its capacity to handle large-scale instances of the FJSP-SDSTTTME. The algorithm effectively addressed problem instances as large as $50 \times 15 \times 15$, which represents a total of 750 operations distributed across 15 machine types, with the number of machines per type ranging from 2 to 4. Notably, the algorithm solved these large instances without difficulty, maintaining relatively short execution times, with the maximum observed runtime being approximately 600 ms. These results not only demonstrate the scalability of the algorithm but also highlight its potential for practical applications in industrial environments, where large-scale scheduling problems are a common occurrence.

In conclusion, the proposed heuristic algorithm offers a robust and effective solution to the Flexible Job-Shop Scheduling Problem with sequence-dependent setup times, transportation times and machine efficiency constraints, providing both high performance and computational efficiency, even for large-scale instances. The integration of machine efficiency into the scheduling model has proven to be a pivotal factor in enhancing overall scheduling outcomes, positioning the algorithm as a valuable tool for addressing real-world manufacturing challenges.

While the model captures several important aspects of real-world systems, it does not yet account for factors such as uncertain processing times, operator-related variability, or multiple transport modes, which may be relevant in specific industrial contexts. The inclusion of such practical factors, along with the validation of the algorithm in an industrial case study, is left as a direction for future work.

As a result, the proposed method offers significant promise for practical applications in industries where flexible and efficient scheduling solutions are required and provides a solid foundation for future research aiming to address even more complex scheduling environments.

References

[1] J. Adams, E. Balas and D. Zawack: The shifting bottleneck procedure for job shop scheduling. *Management Science*, **34**(3), (1988), 391–401. DOI: 10.1287/mnsc.34.3.391

[2] A. Allahverdi: The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, **246**(2), (2015), 345–378. DOI: 10.1016/j.ejor.2015.04.004

[3] P. Brandimarte: Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, **41**(3), (1993), 157–183. DOI: 10.1007/BF02023073

[4] L. Davis: Job shop scheduling with genetic algorithms. *Proceedings of the First International Conference on Genetic Algorithms*, **5** (1985), 136–140.

[5] S. Dauzère-Pérès, J. Ding, L. Shen and K. Tamssaouet: The flexible job shop scheduling problem: A review. *European Journal of Operational Research*, **314**(2), (2024), 409–432. DOI: 10.1016/j.ejor.2023.05.017

[6] E. Demirkol, S. Mehta and R. Uzsoy: Benchmarks for shop scheduling problems. *European Journal of Operational Research*, **109**(1), (1998), 137–141. DOI: 10.1016/S0377-2217(97)00019-2

[7] H. Fisher and G.L. Thompson: Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules. In *Industrial Scheduling*, J.F. Muth and G.L. Thompson (Eds). Prentice-Hall, Englewood Cliffs, (1963), 225–251.

[8] O. Holthaus and C. Rajendran: Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics*, **48**(1), (1997), 87–105. DOI: 10.1016/S0925-5273(96)00068-0

[9] K.-C. Jeong and Y.-D. Kim: A real-time scheduling mechanism for a flexible manufacturing system: Using simulation and dispatching rules. *International Journal of Production Research*, **36**(9), (1998), 2609–2626. DOI: 10.1080/002075498192733

[10] S. Knopp, S. Dauzère-Pérès and C. Yugma: A batch-oblivious approach for complex job-shop scheduling problems. *European Journal of Operational Research*, **263**(1), (2017), 50–61. DOI: 10.1016/j.ejor.2017.04.050

[11] C. Özgüven, L. Özbakir and Y. Yavuz: Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Applied Mathematical Modelling*, **34**(6), (2010), 1539–1548. DOI: 10.1016/j.apm.2009.09.002

[12] M.L. Pinedo: *Scheduling: Theory, Algorithms, and Systems*. Springer, 2016.

[13] W. Ren, Y. Yan, Y. Hu and Y. Guan: Joint optimization for dynamic flexible job-shop scheduling problem with transportation time and resource constraints. *International Journal of Production Research*, **60**(18), (2022), 5675–5696. DOI: 10.1080/00207543.2021.1968526

[14] A. Sadegheih: Scheduling problem using genetic algorithm, simulated annealing and the effects of parameter values on GA performance. *Applied Mathematical Modelling*, **30**(2), (2006), 147–154. DOI: 10.1016/j.apm.2005.03.017

[15] C. Tsiushuang: Dispatching rules for manufacturing job-shop pperations. *IFAC Proceedings Volumes*, **26**(2), (1993), 975–978.

[16] L. Urbanucci: Limits and potentials of mixed integer linear programming methods for optimization of polygeneration energy systems. *Energy Procedia*, **148**, (2018), 1199–1205. DOI: 10.1016/j.egypro.2018.08.021

[17] X. Wu and Y. Sun: A green scheduling algorithm for flexible job shop with energy-saving measures. *Journal of Cleaner Production*, **172**, (2018), 3249–3264. DOI: 10.1016/j.jclepro.2017.10.342

[18] H. Zhang, G. Xu, R. Pan and H. Ge: A novel heuristic method for the energy-efficient flexible job-shop scheduling problem with sequence-dependent set-up and transportation time. *Engineering Optimization*, **54**(10), (2021), 1646–1667. DOI: 10.1080/0305215X.2021.1949007