



2026, № 1

Журнал обчислювальної та
прикладної математики
Заснований в 1965 році



Journal of Numerical
& Applied Mathematics

ISSN (Print): 2706-9680, ISSN (Online): 2706-9699

ЄДРПОУ видавця: 02070944

ROR: 02aaqv166

E-mail:
jnameditorialteam@gmail.com

Головний редактор:
Ляшко Сергій Іванович

Осьмак Є.А., Дробязко О.О.

НЕЧІТКА ЗАДАЧА КАЛЕНДАРНОГО ПЛАНУВАННЯ ТИПУ JOB-SHOR: ОГЛЯД ОСТАННІХ ДОСЯГНЕНЬ

Адреса редакції:
Кафедра обчислювальної математики,
факультет комп'ютерних наук та кібернетики,
Київський національний університет імені Тараса Шевченка,
Україна, 01601, Київ, вул. Володимирська, 64/13.

©Київський національний університет імені Тараса Шевченка, 2026

Свідectво про державну реєстрацію KB 4246 від 26.05.2000
Включено в Перелік наукових фахових видань у відповідності до Наказу МОН
України № 409 від 17.03.2020

DOI: 10.17721/2706-9699.2026.1.04

УДК 519.8

MSC 90-02

FUZZY JOB-SHOP SCHEDULING PROBLEM: A SURVEY OF RECENT ADVANCES

Y. A. OSMAC¹, O. O. DROBIAZKO²

¹Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, E-mail: yevhen.osmak@knu.ua,
ORCID: 0009-0007-9081-5527

²Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, E-mail: olexandrdrobiazko@knu.ua,
ORCID: 0009-0006-1867-0792

НЕЧІТКА ЗАДАЧА КАЛЕНДАРНОГО ПЛАНУВАННЯ ТИПУ JOB-SHOP: ОГЛЯД ОСТАННІХ ДОСЯГНЕНЬ

Є. А. ОСЬМАК¹, О. О. ДРОВ'ЯЗКО²

¹Факультет комп'ютерних наук та кібернетики, Київський національний університет імені Тараса Шевченка, Київ, Україна, E-mail: yevhen.osmak@knu.ua,
ORCID: 0009-0007-9081-5527

²Факультет комп'ютерних наук та кібернетики, Київський національний університет імені Тараса Шевченка, Київ, Україна, E-mail: olexandrdrobiazko@knu.ua,
ORCID: 0009-0006-1867-0792

ABSTRACT. The Fuzzy Job-Shop Scheduling Problem (FJSSP) extends the classical Job-Shop Scheduling Problem (JSSP), a well-known NP-hard combinatorial optimization task, by representing uncertainty in processing times, due dates, and related parameters via fuzzy sets. This paper presents a comprehensive literature review of the FJSSP, updating and extending the seminal survey published in 2014. Covering the period from 2014 to 2025, the review classifies FJSSP formulations based on fuzzy representations, including triangular and interval models, Intuitionistic Fuzzy Sets (IFS), and Interval Type-2 Fuzzy Sets (IT2FS). It summarizes constraint variants and objective functions, spanning completion-time criteria, energy and emission costs, robustness, and quality metrics. The review surveys exact, heuristic, and metaheuristic solvers alongside emerging learning-based methods, specifically Deep Reinforcement Learning (DRL) on graph representations. Furthermore, it analyzes benchmark datasets and fuzzification methods, including recent generative protocols for dynamic disruptions. Finally, the paper identifies open gaps, highlighting the need for hybrid fuzzy-stochastic uncertainty

Corresponding author: O. O. Drobiazko (olexandrdrobiazko@knu.ua).

© Yevhen Osmak, Oleksandr Drobiazko, 2026. This is an open-access article distributed under the terms of **Creative Commons Attribution Licence (CC BY)**.

models, the standardization of IT2FS, the integration of structural fuzziness for problem definitions, and tighter coupling with Digital Twins (DT) in service-oriented scheduling.

KEYWORDS: Job-Shop Scheduling Problem, Fuzzy Job-Shop Scheduling Problem, literature review.

АНОТАЦІЯ. Нечітка задача календарного планування типу job shop (FJSSP) розширює класичну задачу календарного планування типу job shop (JSSP), відому NP-складну комбінаторну оптимізаційну задачу, шляхом представлення невизначеності в часі обробки, термінах виконання та пов'язаних параметрах за допомогою нечітких множин. У цій статті представлено вичерпний огляд літератури з FJSSP, який оновлює та розширює основоположний огляд, опублікований в 2014 році. Стаття охоплює період з 2014 по 2025 рік і класифікує формулювання FJSSP на основі нечітких представлень, включаючи трикутні та інтервальні моделі, інтуїціоністські нечіткі множини (IFS) та інтервальні нечіткі множини типу 2 (IT2FS). Огляд підсумовує варіанти обмежень та цільові функції, що охоплюють критерії часу виконання, витрати енергії та викидів, надійність та показники якості. Були розглянуті точні, евристичні та метаевристичні методи, а також нові підходи, засновані на навчанні, зокрема глибоке навчання з підкріпленням (DRL) на графічних представленнях. Крім того, в ньому аналізуються тести продуктивності та методи фузифікації, включаючи останні генеративні протоколи для динамічних порушень. У статті також визначено відкриті прогалини у цій галузі, підкреслюючи необхідність гібридних нечітких стохастичних моделей невизначеності, стандартизації IT2FS, інтеграції структурної нечіткості для визначення проблем, та тіснішого зв'язку з цифровими двійниками (DT) в сервісно-орієнтованому плануванні. КЛЮЧОВІ СЛОВА: задача планування роботи цеху, нечітка задача планування роботи цеху, огляд літератури.

1. BASIC INFORMATION

This paper is a literature review based on the latest advances in the field of the Fuzzy Job Shop Scheduling Problem (FJSSP). It serves as a logical follow-up and systematic extension of a well-known article [1], providing an up-to-date overview of **recent** scientific results in this area.

The **purpose** of this work is the following:

1. to collect and structure topic-specific scientific papers published between 2014 and 2025
2. to highlight articles that were not included in the initial literature review [1] but still have scientific value.

The review **is structured** as follows:

Section 2 provides an introduction to the **JSSP**, tracing its evolution into the fuzzy domain and highlighting the shift from theoretical optimization to data-driven manufacturing.

Section 3 offers an **overview and classification** of FJSSP models, summarizing legacy frameworks and introducing modern advancements such as **Intuitionistic Fuzzy Sets (IFS)** and **Interval Type-2 Fuzzy Sets (IT2FS)**.

Section 4 examines the **variation in constraints and objectives**, detailing new physical factors like transportation time and energy consumption, as well as emerging goals related to sustainability and robustness.

Section 5 categorizes **algorithmic approaches**, documenting the transition from classical metaheuristics to adaptive, data-driven **Deep Reinforcement Learning (DRL)** and **Graph Transformers**.

Section 6 reviews **benchmark datasets**, contrasting traditional fuzzified instances with modern generative protocols and real-world industrial logs.

Section 7 concludes the study by synthesizing **research gaps**, specifically identifying the omission of **structural fuzziness** and the lack of models for **service-oriented scheduling** in high-stakes industries.

Methodology of the survey. To ensure a comprehensive update, the research followed a structured three-phase systematic review process:

Data Collection and Search Strategy: We performed a systematic search across the **Web of Science** and **Scopus** databases using primary keywords: «fuzzy», «job-shop scheduling», «FJSSP», and «uncertainty modeling». The search was primarily bounded between 2014 and 2025 to capture the most recent decade of advancements. The corresponding references lists were analyzed and taken into account.

Article Selection and Filtering: Beyond the primary search, a «backward-looking» selection was performed to identify valuable papers that were published before 2014 but omitted from the original survey, providing a more complete historical record.

Comparative and Structural Analysis: The analysis method involved a **comparative mapping** of new literature against the original 2014 classifications. Each section of the original survey was first summarized as a baseline and then expanded with contemporary citations.

Classification of Advancements: We categorized the findings into four distinct dimensions: sophisticated modeling (e.g., IT2FS, IFS), expanded constraints (e.g., energy, logistics), algorithmic evolution (from metaheuristics to DRL), and benchmark modernization.

Gap Identification: A qualitative synthesis was used to evaluate the change in the «maturity» of the field.

2. INTRODUCTION

JSSP is a critical NP-hard combinatorial optimization problem with significant industrial and operational management implications. In practice, the common assumption of certain (crisp) parameters, especially processing times, rarely holds due to human variability and other sources of uncertainty; likewise, expected due dates often allow for some tolerated lateness. To reflect these factors, the FJSSP models processing times and due dates as fuzzy sets. Since the mid-1960s, thousands of studies have addressed the classical JSSP.

Its fuzzy counterpart emerged in 1995, and in 2014, it consisted of fewer than sixty papers with limited algorithmic diversity, indicating a young field.

One of the first comprehensive maps of FJSSP [1], when Abdullah and Abdolrazzagh-Nezhad classified existing models, proposed a three-stage structure for metaheuristics (preprocessing, initialization, improvement), and identified key gaps.

The past decade justifies a full re-evaluation of the subject area. Research volume and sophistication have increased substantially, and FJSSP has shifted from a largely theoretical optimization task toward a data-driven manufacturing problem. This evolution reflects two forces: an internal broadening of methods in response to the gaps identified in 2014, and an external shift driven by AI and machine learning proliferation that changes the way the problem is formulated and studied. A new, holistic survey is therefore needed to synthesize recent progress, overview current directions, and set a further research agenda.

Thus, the purpose of this study is to collect and summarize the main developments in this field over the past 11 years and to help researchers quickly become familiar with the relevant literature.

3. OVERVIEW AND CLASSIFICATION OF FJSSPs

The following overview categorizes the core FJSSP models originally identified in the [1], serving as the baseline for contemporary advancements. As can be seen in the papers, published during 2014-2025, recent modeling efforts have focused exclusively on JSSPs with fuzzy processing time.

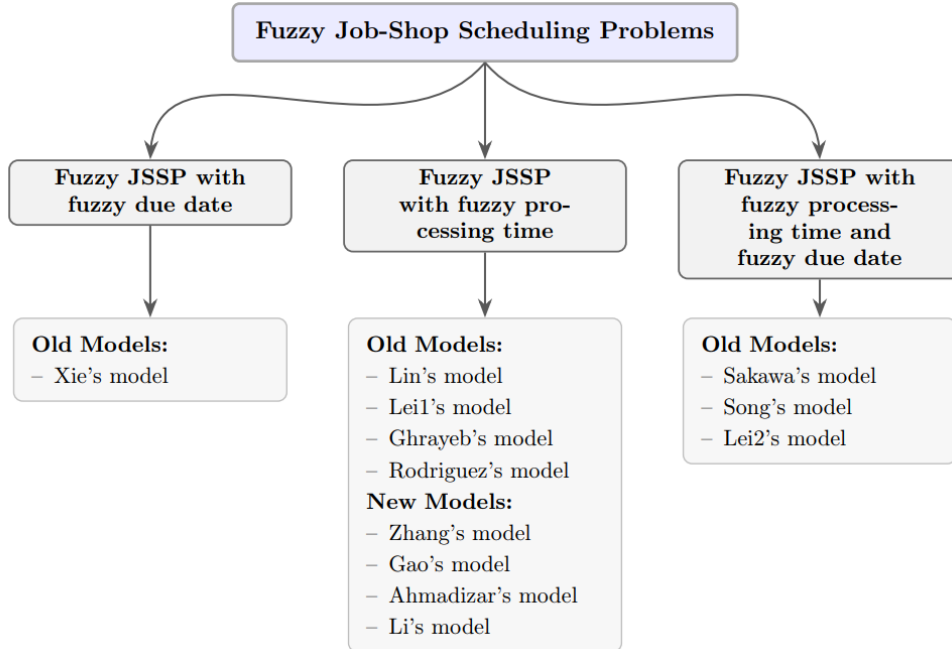


FIGURE 1. Fuzzy JSSP models

1. **Xie's model:** FJSSP with *fuzzy due dates* and *crisp* processing and decision variables. Each job J_i has a double due date $\tilde{D}_i = (d_i^1, d_i^2)$ with satisfaction (membership) for the completion time C_{im} :

$$f_i(C_{im}) = \begin{cases} 1, & C_{im} \leq d_i^1, \\ \frac{d_i^2 - C_{im}}{d_i^2 - d_i^1}, & d_i^1 < C_{im} \leq d_i^2, \\ 0, & C_{im} \geq d_i^2. \end{cases}$$

Objective: maximize the minimum satisfaction over jobs,

$$SD_{\min} = \left(\min_{1 \leq i \leq n} f_i(C_{im}) \right) \rightarrow \max.$$

2. **Lin's model:** Fuzzy *processing times*. Processing of operation O_{ij} is a triangular fuzzy number (TFN) $\tilde{p}_{ij} = (p_{ij}^1, p_{ij}^2, p_{ij}^3)$ and is *de-fuzzed* by the fuzzy mean

$$p_{ij}^* = p_{ij}^2 + \frac{1}{4} \left((p_{ij}^3 - p_{ij}^2) - (p_{ij}^2 - p_{ij}^1) \right)$$

then the (crisp) makespan built from P_{ij}^* is minimized.

3. **Ghrayeb's model:** Fuzzy *processing times*. Uses Sakawa's criterion to approximate the fuzzy max and minimizes two measures of the fuzzy makespan $\tilde{C}_{\max}$:

$$\begin{aligned} TIV^\beta(\tilde{C}_{\max}) &= \frac{1}{2} (\beta C_{\max}^1 + C_{\max}^2 + (1 - \beta) C_{\max}^3), \\ U(\tilde{C}_{\max}) &= C_{\max}^3 - C_{\max}^1, \quad \beta \in [0, 1]. \end{aligned}$$

4. **Rodriguez's model:** Fuzzy *processing times*. Also uses Sakawa's criterion; the objective is the expected fuzzy makespan

$$\mathbb{E}[\tilde{C}_{\max}] = \frac{C_{\max}^1 + 2C_{\max}^2 + C_{\max}^3}{4} \rightarrow \min.$$

5. **Lei's model:** Fuzzy *processing times*. Approximates fuzzy max with *Lei's criterion*. Covers both standard FJSSP and *Flexible* FJSSP (machine alternatives). Objective: minimize fuzzy/expected makespan (e.g., $\min \mathbb{E}[\tilde{C}_{\max}]$).
6. **Sakawa's model:** *Both* fuzzy processing times (triangular) and fuzzy due dates (double). Uses Sakawa's criterion and maximizes the minimum *agreement index*

$$AI_i = \frac{\text{area}(\tilde{C}_{im} \cap \tilde{D}_i)}{\text{area } \tilde{C}_{im}}, \quad AI_{\min} = \min_i AI_i \rightarrow \max.$$

7. **Song's model:** Both fuzzy processing times (triangular) and *trapezoidal* fuzzy due dates $\tilde{D}_i = (d_i^1, d_i^2, d_i^3, d_i^4)$ with

$$\mu_{\tilde{D}_i}(t) = \begin{cases} 0, & t < d_i^1 \text{ or } d_i^4 < t, \\ \frac{t - d_i^1}{d_i^2 - d_i^1}, & d_i^1 \leq t < d_i^2, \\ 1, & d_i^2 \leq t \leq d_i^3, \\ \frac{d_i^4 - t}{d_i^4 - d_i^3}, & d_i^3 < t \leq d_i^4. \end{cases}$$

An objective/approximation method aligned with Sakawa's model (maximize $AI_{\min}$ with Sakawa's criterion).

8. **Lei's model:** Both fuzzy processing times and fuzzy due dates (as in Sakawa/Song). Same agreement-index objective as Sakawa, but the fuzzy max is approximated with *Lei's criterion* instead of Sakawa's.

Omitted and new models for 2014-2025

1. [2] offers an novel extension to the fuzziness framework, using Intuitionistic Fuzzy Sets (IFS), applied to an uncertain processing time. An IFS in X is defined as

$$A = \{ \langle x, \mu_A(x), v_A(x) \rangle \mid x \in X \},$$

with condition $0 \leq \mu_A(x) + v_A(x) \leq 1$ for all $x \in X$. $\mu_A(x)$ and $v_A(x)$ denote, respectively, the degree of membership and the degree of non-membership of the element x in the set A . In addition, $\pi_A(x)$ denotes the degree of uncertainty towards $x \in X$,

$$\pi_A(x) = 1 - \mu_A(x) - v_A(x), \quad x \in X.$$

The paper introduces a generalized IFS approach to process additive operations and to compare two IFSs in the context of FJSSP.

2. [3] considers Fuzzy group shop scheduling (FGSS) with fuzzy processing time and fuzzy job release date, which is stated as follows. There are n jobs to be processed on a set of m machines. Each job has at most one operation to be processed on each machine. The operations of each job are partitioned into groups, such that the operations in the same group are unordered while those in two distinct groups satisfy the predefined precedence constraint between the groups. If each such group consists of only one operation, the FGSS is equivalent to a JSSP. If all operations of each job belong to just one group, the problem is equivalent to an open-shop scheduling problem.
3. [4] considers FJSSP with fuzzy processing time and new job insertion. A model specifically designed for re-manufacturing, which includes two constraints: fuzzy processing time and new job insertion for dynamic scheduling.
4. In [5] consider a type-2 fuzzy sets system (T2FS), which has an enhanced ability to handle of uncertainty. Here the processing time of each job is a nonsymmetric triangular interval T2FS (IT2FS) value.

A type-2 fuzzy set, say $\tilde{A}$, is generally characterized by a membership function $\mu_{\tilde{A}}(x, u)$, where $x \in X$, and $u \in [0, 1]$. Based on the membership function, the definition of the T2FS is given as follows (Considering only the simplified case IT2FS):

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid \forall x \in X, \forall u \in [0, 1]\}, \quad 0 \leq \mu_{\tilde{A}}(x, u) \leq 1.$$

An IT2FS, denoted as $\tilde{A}_I$, is a special case of T2FS, which is generally characterized as

$$\tilde{A}_I = \{((x, u), 1) \mid \forall x \in X, \forall u \in [0, 1]\}.$$

Thus, the sources describe significant progress in modeling fuzzy planning problems, including the use of higher types of fuzzy sets, the integration of complex realistic constraints (transportation time, job insertion, release date), and new production goals (energy efficiency, downtime minimization). At the same time, it should be mentioned that the global disruptions associated with the COVID-19 pandemic (2020-2022) and increased uncertainty from related supply chain disruptions had a profound influence on the research output in this subject area. Considering the number of articles published on this topic over the years, we can say that in 2020 there were approximately the same number as before, but in 2021 and 2022 there was a significant decline in their quantity. At the same time, the pandemic did not affect the choice of topics in this field, nor did it give rise to new ideas.

4. VARIATION IN CONSTRAINTS AND OBJECTIVES INVESTIGATED IN FJSSP

As the original article [1] states, FJSSP have **precedence**, **capacity**, and **release and due date** constraints:

1. **Precedence constraints:**

- 1) Each job should be processed through the machines in a predetermined order
- 2) The machine orders among different jobs are unconfined
- 3) No precedence constraints exist among the operations of different jobs

2. **Capacity constraints:**

- 4) Machines are independent from each other
- 5) Machines require a certain time span between jobs or cannot remain idle
- 6) Each machine can only handle one operation at a time
- 7) Every job can be processed only once on a given machine
- 8) Jobs are independent of one another

3. **Release and due date constraints:**

- 9) negative starting times are not allowed
- 10) the processing times of operations have a given length
- 11) the processing of each operation must not be interrupted

4. **Constraints unique to Flexible FJSSP:**

- 12) Each job operation can be processed by one machine out of a set of candidate machines, which is meant by flexibility

During the studied time frame of 2014-2025, significant extensions to the traditionally considered constraint types were developed. These new types of constraints and their characteristics primarily fall into groups related to **logistics, setup and maintenance; operational, dynamic and quality;** and **environmental and energy requirements**. Here are the new types of constraints discussed.

1. Constraints Related to Logistics, Setup, and Maintenance.

These constraints reflect the physical factors and operational limitations of the manufacturing processes:

- (a) Setup Time: Setup activities as constraints can be critical in practice [6]. This includes:
 - (i) Sequence-Dependent Setup Times: the duration of the setup activity depends on the operation that immediately precedes it on the same machine, which reflects many practical applications [7–9].
 - (ii) Setup Time Available: an upper limit on the total setup times available on a specific machine can be imposed [16].
- (b) Transportation or Move Time: The time required to transport a job between machines is explicitly considered, especially in flexible JSSP [10].
- (c) Machine Availability Constraints: Machines may not be available during the production horizon:
 - (i) Preventive Maintenance (PM) Activity (planned): Operations may be interrupted or delayed by PM, which can be fixed or flexible interval [11].
 - (ii) Machine Breakdown or Unavailability Intervals (unplanned). In [2], this type of constraint is marked as a direction for future research. So, there is definitely a gap in constraints study.
- (d) Storage and Buffers: Constraints regarding where materials wait between operations:
 - (i) Limited Intermediate Buffer Capacity: Available storage space between machine groups may be limited, unlike in classical JSSP, which disregards this condition [12].
 - (ii) Zero-Wait Storage Policy: A constraint where once processing is finished at a machine group, the next operation must start immediately (zero-wait), is addressed in the same paper [12] as the previous one.

2. Operational, Dynamic, and Quality Constraints.

These constraints reflect the need to manage job shop workflow dynamics or ensure specific quality and environmental objectives, which are represented as constraints, not objectives per se:

- (a) Dynamic Scheduling Events or New Job Insertion: In dynamic systems, the arrival and insertion of new jobs into an existing schedule serve as a trigger for rescheduling [4, 13]. It is also mentioned as a direction for future research in [14].

- (b) Lot-Sizing or Batching: Constraints related to splitting jobs into smaller production lots, so-called batches, to facilitate overlapping operations or meet specific customer delivery demands [14].
 - (c) Processing Quality: When dealing with critical orders in a sense orders, prioritization is given to machines with adequate processing quality [15].
 - (d) Part Defect Percentage: Constraints related to the acceptable rate of product defects [16].
3. **Environmental and Energy Constraints.** A growing area of research focuses on green scheduling, introducing constraints related to energy usage and environmental impact:
- (a) Energy Consumption and Idle Time: It aims to minimize objectives such as total non-processing energy, requiring the calculation of machine idle times [17].
 - (b) Greenhouse Gas Emission Limitation: Limitations on total emissions generated by facilities. [16]
 - (c) Total Electricity Consumption Limitation: Explicit restrictions on the maximum amount of electricity consumed are considered to be the same as in the previous paper [16].

Article [1] lists the next types of objective functions:

1. Maximize the minimum satisfaction degree: $SD_{\min} \rightarrow \max$
2. Maximize the minimum agreement index: $AI_{\min} \rightarrow \max$
3. Minimize the defuzzified makespan: $C_{\max}(p^*) \rightarrow \min$
4. Minimize the expected fuzzy makespan: $\mathbb{E}(\tilde{C}_{\max}) \rightarrow \min$
5. Minimize the integral value and uncertainty of the fuzzy makespan:
 $TIV_{\beta}(\tilde{C}_{\max}) \rightarrow \min, \quad U(\tilde{C}_{\max}) \rightarrow \min$

New Objective Functions Considered in the papers, published thereafter are the following:

1. **Environmental, Energy, and Integrated Cost Objectives**
 - (a) **Minimizing Total Energy Consumption / Energy Cost:** This objective is often minimized simultaneously with makespan or tardiness, focusing on reducing the energy consumed. It includes the consumed power cost [18].
 - (b) **Minimizing Total Non-Processing Energy:** A specific energy-related metric, representing the energy consumed when machines are idle, defined and studied in the fuzzy context [17].
 - (c) **Minimizing Total Emitted Carbon Footprint / Greenhouse Gas (GHG) Emission:** Focusing on environmental impact, the objective may involve minimizing the total emitted carbon footprint or minimizing the cost associated with GHG emissions, including taxes, penalties, and discounts [10].

- (d) **Minimizing Total Cost of Production System:** This highly broad objective integrates various financial and environmental factors, including costs related to consumed power, setup, processing, over/under-utilized time, part defects, and produced emissions [16].
- (e) **Minimizing Total Core Idle Time:** This objective specifically minimizes the sum of core idle times across all resources, reflecting efforts to manage non-productive machine states in the flexible JSSP [19].
- (f) **Maximizing Satisfaction of Energy Efficiency:**

$$S_{E_n} = \frac{E_c - E_m}{E_M - E_m}$$

Where E_c denotes the energy consumption of the tasks set, E_M and E_m are maximum and minimum consumptions, respectively. It is used as a cost function in, for instance, real-time embedded systems and is maximized to ensure the system consumes an optimal amount of energy [10, 17, 20].

- (g) **Minimizing Consumption in a Delayed Scenario:** A specialized objective function, particularly in railway operations, which minimizes energy consumption based on a delayed, yet more possible, processing scenario, incorporating the driver's behavioral response to delays and fulfilling punctuality requirements under uncertainty [21].

2. Robustness, Quality, and Uncertainty Metrics

- (a) **Maximizing Production Quality / Total Processing Quality:** A critical objective, especially in high-precision manufacturing, e.g., aerospace shaft parts, that quantifies customer demand and processing resource capacities to prioritize the allocation of high-precision or high-quality machines for crucial orders [15].
- (b) **Maximizing Robustness / Minimizing Robust Deviation:** This focuses on minimizing the predictive error or expected average deviation between the predicted schedule performance and the actual executed schedule performance, aiming for solutions that are stable against uncertainties modeled by fuzzy numbers. It is usually a component of multi-objective optimization [22–24].
- (c) **Maximizing Expected Satisfaction Degree:** Proposed as a new and more accurate measure of due-date satisfaction under uncertainty, this objective is maximized, aiming to address perceived weaknesses in the traditional agreement index criteria [25].

3. Refined Time, Flow, and Due-Date Objectives

- (a) **Minimizing Total Weighted Completion Times:** Used primarily in problems of open shop type, this objective optimizes the sum of completion times weighted by job relative importance. [3, 9, 14].

- (b) **Minimizing Total Delivery Tardiness / Total Weighted Tardiness:** Minimizing the total weighted tardiness, which measures the extra time taken to complete jobs after their due dates, is an objective often combined with the minimization of energy consumption. The attention to this type of objective indicates a increased interest in and importance of this performance indicator. [8, 10, 16, 17, 25, 26].
- (c) **Minimizing Total Earliness/Tardiness Penalties:** This objective minimizes the combined penalty costs associated with finishing jobs too early (earliness) or too late (tardiness), directly addressing the goal of Just-In-Time production. This objective is often expressed as minimizing a single aggregated penalty cost function. [8, 12, 19].
- (d) **Minimizing Maximum Tardiness:** Minimizing the worst late delivery time among all jobs, considered in multi-objective scheduling problems. This approach has not been actively researched lately; only several papers addressed it, which are not included in the original research [26–28].
- (e) **Maximizing Satisfaction Grade of Average Tardiness and Number of Tardy Jobs:** Converting fuzzy objectives, like minimizing average tardiness and minimizing the number of tardy jobs, into fuzzy satisfaction grades, which are then aggregated, e.g., using simple or weighted average operators. This approach has also not been actively researched lately [26].

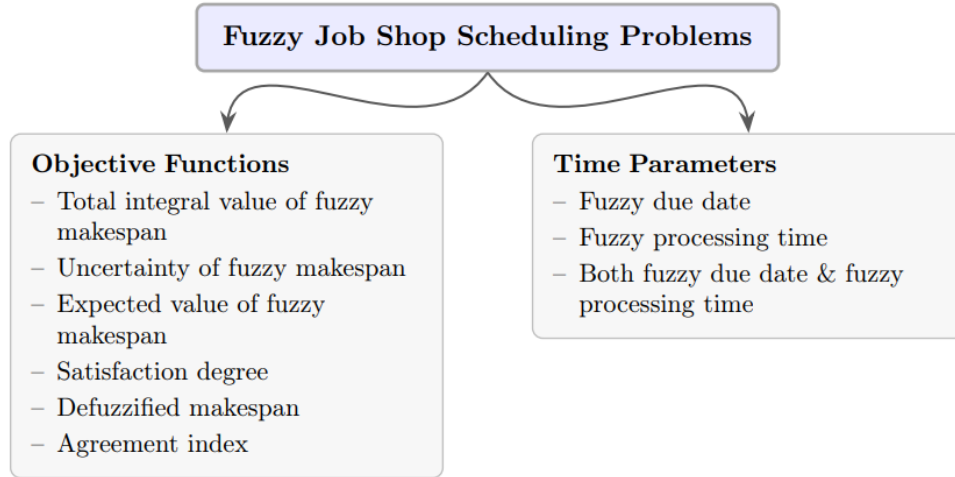


FIGURE 2. Fuzzy JSSPs: Objective Functions and Time Parameters

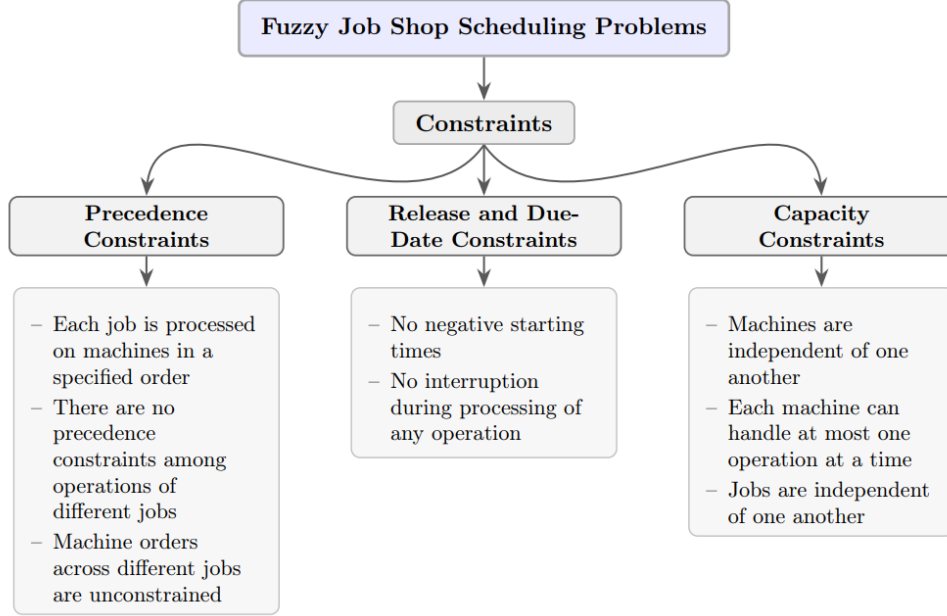


FIGURE 3. Fuzzy Job Shop Scheduling: Constraints

5. EXACT, HEURISTIC, AND META-HEURISTIC ALGORITHMS APPLIED TO FJSSP

All methods for solving FJSSP problems can be grouped into three types:

1. **Exact algorithms:** Solve to proven optimality with certificates; run-time often exponential due to the fact that this problem is NP-hard; best for small or highly structured cases.
2. **Heuristic algorithms:** Problem-specific rules that quickly construct and improve feasible solutions, but not guaranteed to be optimal or convergent; simple and scalable.
3. **Meta-heuristic algorithms:** General search framework that controls how one or more heuristics are applied over time to balance exploration and exploitation, often using stochastic elements; flexible across problems and effective at scale; produces high-quality solutions under time limits without proofs. As defined in [1], they have 3 steps:
 - **Pre-processing:** Select a representation that encodes a schedule into a searchable vector; design a decoder that maps any encoded point to a feasible active schedule by computing fuzzy start/completion times with fuzzy sum and an approximate fuzzy max, enforcing precedence, capacity, and release/due-date constraints, and defining the objective and fitness evaluation for the decoded schedule.

- **Initialization:** Generate an initial population or single seed of encoded solutions with the decoder to ensure feasibility; use randomized construction and/or priority-rule construction to create diverse starting points that cover the search space while keeping decoding simple and fast.
- **Improvement:** Iteratively refine candidates by applying search operators and neighborhood moves; evaluate each decoded schedule with the chosen objective, intensify around high-quality regions with local search, diversify with controlled perturbations, allow hybrid strategies when useful, and stop at convergence or selected limits.

Let us look through meta-heuristic methods:

1. Methods used to solve FJSSP in [1]
 - 1) **GA (Genetic Algorithm):** Encodes schedules, e.g., operation-based or random keys, and evolves a population via selection, crossover, and mutation; a decoder builds feasible fuzzy schedules and fitness targets expected fuzzy makespan or agreement index.
 - 2) **CGA (Co-evolutionary Genetic Algorithm):** Splits the search into co-evolving subpopulations, e.g., machine assignment vs. operation sequencing, that periodically exchange individuals to balance exploration and intensification.
 - 3) **PSO (Particle Swarm Optimization):** Uses discrete particle encodings (swap/insert velocity operators) to move candidate permutations toward personal/global bests; often hybridized with local search to handle sequencing effectively.
 - 4) **ACO (Ant Colony Optimization):** Constructs operation sequences probabilistically using pheromone trails and heuristic «visibility» (machine availability, due-date satisfaction), then decodes to a feasible fuzzy schedule.
 - 5) **BCO (Bee Colony Optimization):** Models employed/ onlooker/ scout bees; elite solutions (food sources) share information via waggle-dance-style selection, while scouts diversify the search; used in hybrid forms for FJSSP.
 - 6) **TS (Tabu Search):** Intensifies a local move-based search (e.g., critical-block swaps) while a tabu list prevents cycling; strong for sequencing with memory-based guidance.
 - 7) **SA (Simulated Annealing):** Accepts worsening moves under a cooling schedule to escape local minima; is simple to implement and often used as a GA/TS hybrid component.
 - 8) **NN (Neural Networks):** Maps scheduling to an energy-minimization network (e.g., Hopfield), encoding machine/order constraints and letting dynamics converge to a feasible schedule.
 - 9) **AIS (Artificial Immune System):** Applies clonal selection and hypermutation to permutations; «memory cells» preserve high-affinity (good) schedules and guide diversification.

- 10) **EDA (Estimation of Distribution Algorithm)**: Learns a probabilistic model over good permutations (e.g., factorized order models) and samples new schedules, capturing precedence patterns directly.
- 11) **DEA (Differential Evolution Algorithm)**: Adapts differential mutation and recombination to discrete encodings (e.g., permutation-aware operators) to explore neighborhoods using population-wise directional differences.
2. Methods that do not solve FJSSP in [1], however, theoretically have the potential to solve FJSSP
 - 12) **EM (Electromagnetism-like Mechanism)**: Population points attract/repel according to objective-induced «charges».
 - 13) **VNS (Variable Neighborhood Search)**: Systematically changes neighborhood structures to escape local minima.
 - 14) **NGA (Neural-Genetic Approach)**: Hybrids where NNs evaluate/repair GA individuals or supply features.
 - 15) **DNAC (DNA Computing)**: Uses biomolecular parallelism to encode and filter combinatorial solutions; discussed conceptually.
 - 16) **CRO (Chemical Reaction Optimization)**: Treats candidate solutions as molecules undergoing reaction-like operators (collision, decomposition) to explore.
 - 17) **GEP (Gene Expression Programming)**: Evolves expression trees from linear chromosomes.
 - 18) **FA (Firefly Algorithm)**: Moves solutions toward brighter (better) ones under attractiveness with randomization.
 - 19) **ICA (Imperialist Competitive Algorithm)**: Models empires assimilating colonies with competition/assimilation moves.
3. In the papers published after 2014, the following methods are used:
 - (a) **GA and Evolutionary Algorithms (EA)**
 Genetic and evolutionary algorithms are the most used methods found in the sources. Advancements in this area focus on hybridization, co-evolution, and controlling parameters using fuzzy logic, which is not the focus of our research, though.
 - (i) **Genetic Algorithm with Tabu Search**
 These papers propose hybridizing GA with TS to balance exploration and exploitation. Advancements include using **heuristic seeding** to initialize populations with high-quality solutions and defining specific **neighborhood structures**, like reversing critical arcs or changing machine assignments, that ensure feasibility and connectivity. [25, 49]
 - (ii) **Co-Evolutionary Algorithms**
 These methods decompose the problem into two distinct populations that evolve cooperatively: one for **machine assignment** and one for **task ordering**. This separation enables use of specific genetic operators for each sub-problem [43].

(iii) **Controlled Genetic Algorithms & Fuzzy Logic Control**

These methods use fuzzy logic to dynamically control the genetic parameters (crossover and mutation probabilities) based on the population's performance or use fuzzy logic to evaluate the fitness of solutions, e.g., using «Approach by Localization» [30, 48, 54].

(iv) **Multi-Objective Evolutionary Algorithms** These approaches adapt the NSGA-II framework to fuzzy environments by incorporating Heuristic Reduction of Energy Consumption strategies and defining fuzzy dominance relationships [6, 17].

(v) **Improved Adaptive GA**

Utilizes a hormone modulation mechanism to adaptively vary crossover and mutation rates based on population fitness variance [18].

(b) **Artificial Bee Colony (ABC) Optimization**

ABC algorithms simulate the foraging behavior of bees. In FJSSP, improvements focus on the initialization of food sources and the movement strategies of the bees.

– **Hybrid and Improved ABC**

These papers introduce **variable neighborhood search** into the ABC framework to enhance local search. They also propose specific initialization heuristics like **MinEnd**, i.e., assigning operations to machines with minimum fuzzy completion time [4, 32].

– **Two-Stage ABC**

Considers **new job insertion** (rescheduling) and fuzzy processing times simultaneously, using a two-stage strategy [4].

– **Modified ABC**

Adopts a **three-point satisfaction-degree model** to calculate the agreement index for multi-objective problems [57].

(c) **Particle Swarm Optimization (PSO)**

In fuzzy scheduling, the continuous nature of PSO requires adaptation to the discrete FJSSP.

– **Pareto Archive PSO**

Converts JSSP into a continuous problem using priority rule-based representation. Combines global best position selection with **archive maintenance** based on crowding measures [37].

– **Robust PSO**

Uses a priority-based representation and a specific decoding algorithm **parameterized G&T** to generate possibly active schedules [23].

(d) **Ant Colony Optimization**

Replaces the standard transition probability function with a **fuzzy rule base** to control the behavior of artificial ants [55].

(e) **Other Nature-Inspired & Physics-Based Meta-heuristics**

Several newer meta-heuristics have been adapted for FJSSP by integrating specific encoding/decoding schemes and local search operators.

- **Spider Monkey Optimization:** Discretizes the continuous Spider Monkey optimization by introducing genetic operators and variable neighborhood search [15].
- **Flower Pollination Algorithm:** Introduces a discrete operator to handle combinatorial optimization [31].
- **Chemical Reaction Optimization:** Defines elementary reactions (decomposition, synthesis) using specific crossover functions and TS-based local search [11].
- **Biogeography-Based Optimization:** Uses **path relinking** as a migration operation and insertion-based local search for mutation [33].
- **Multi-Verse Optimization:** Mimics black/white hole tunnels using path relinking and wormholes using insertion-based heuristics [35].
- **Teaching-Learning-Based Optimization:** Incorporates a bi-phase crossover scheme based on the teaching-learning mechanism [40].

(f) **Estimation of Distribution Algorithms**

Designed for dynamic flexible FJSSP. It transforms dynamic problems into static «non-zero state» problems and uses probability matrix updates to generate solutions [13].

(g) **Hyper-Heuristics**

Uses the backtracking search algorithm as a high-level strategy to manage a pool of **low-level heuristics** (like random swap, adjacent swap, and inverse) [42].

(h) **Application of Artificial Neural Networks and Reinforcement Learning to FJSSP**

The integration of Artificial Neural Networks (ANN) and Reinforcement Learning (RL) into scheduling problems marks a significant paradigm shift from static heuristic rules to adaptive, data-driven decision-making. In the context of the flexible FJSSP, recent literature demonstrates a clear trajectory from tabular Q-learning-assisted metaheuristics to advanced Deep Reinforcement Learning (DRL) frameworks utilizing Graph Neural Networks.

– **RL-Assisted Evolutionary Algorithms**

The initial application of learning mechanisms in Fuzzy scheduling primarily focused on enhancing the search capability of classical evolutionary algorithms. In these hybrid frameworks, RL agents—typically using tabular Q-learning

— are employed to adjust parameters based on the current search state. [59] established a foundational approach by utilizing a weighted Q-learning algorithm to facilitate adaptive scheduling, demonstrating that learning mechanisms could effectively navigate the search space of complex job shops better than static rules. Building on this, [58] proposed a Reinforcement Learning-based RMOEA/D for the bi-objective flexible FJSSP. In this paper, the RL agent guides the multi-objective evolutionary algorithm, balancing the trade-off between convergence and diversity in a fuzzy environment.

Further addressing the dynamic nature of modern manufacturing, [60] developed a Q-learning-based multi-objective immune algorithm. Their study specifically targets FJSSP under dynamic disruptions. Similarly, [62] proposed an RL-enhanced memetic algorithm, emphasizing robustness and adaptability in multi-objective FJSSP scenarios. [41] proposed EMARM (Expectation - Maximization Algorithm-based Autoregressive Model), which uses an autoregressive decoder to generate schedules, effectively solving the «chicken-and-egg» problem of training without ground-truth labels in fuzzy scheduling.

– **Distributed and Flow Shop Extensions**

As manufacturing systems become more decentralized, the scope of learning-based scheduling has expanded to distributed environments. [61] introduced a Reinforcement-Learning-Based 3-D Estimation of Distribution Algorithm (EDA) for the fuzzy distributed hybrid flow-shop.

Furthermore, [63] advanced the learning approach by proposing a Meta-Q-Learning (MQL-MM) approach for the energy-efficient distributed fuzzy hybrid blocking flow-shop.

– **Deep Learning and Graph Transformers**

The most significant recent advancement lies in the transition from tabular RL to DRL, where ANNs are used to approximate value functions or policies for large-scale instances. A notable result is presented by [64], who introduced a framework based on a Scalable Graph Transformer. Unlike previous approaches that rely on hand-crafted state features, this method utilizes the attention mechanisms of the Graph Transformer to automatically extract features from the disjunctive graph representation of the FJSSP. This allows the RL agent to handle large-scale problems effectively, demonstrating superior scalability and generalization capabilities compared to traditional hybrid metaheuristics.

6. BENCHMARK DATASETS FOR FJSSPs

The paper [22] provides the main benchmarks about FJSSPs up to 2016. However, the landscape has expanded significantly thereafter, driven by the requirements of DRL and Industry 4.0 applications:

1. **Common encoding and evaluation used across the benchmark literature** Processing times are typically modeled as triangular fuzzy numbers (TFNs) $\tilde{p}_{ij} = (p_{ij}^1, p_{ij}^2, p_{ij}^3)$ with membership function

$$\mu_{\tilde{p}_{ij}}(x) = \begin{cases} \frac{x - p_{ij}^1}{p_{ij}^2 - p_{ij}^1}, & p_{ij}^1 \leq x \leq p_{ij}^2, \\ \frac{p_{ij}^2 - x}{p_{ij}^2 - p_{ij}^3}, & p_{ij}^2 < x \leq p_{ij}^3, \\ 0, & x < p_{ij}^1 \text{ or } x > p_{ij}^3. \end{cases}$$

A standard scalar summary is the expected value.

$$\mathbb{E}[\tilde{p}_{ij}] = \frac{p_{ij}^1 + 2p_{ij}^2 + p_{ij}^3}{4}.$$

Two common ways to compare TFNs are:

$$\tilde{A} \leq_E \tilde{B} \iff \mathbb{E}[\tilde{A}] \leq \mathbb{E}[\tilde{B}],$$

and the ranking $<_R$ based on indices $c_1(\tilde{N}) = \mathbb{E}[\tilde{N}]$, $c_2(\tilde{N}) = n_2$, $c_3(\tilde{N}) = n_3 - n_1$, compared lexicographically. Arithmetic typically uses

$$(m_1, m_2, m_3) + (n_1, n_2, n_3) = (m_1 + n_1, m_2 + n_2, m_3 + n_3),$$

and an approximated maximum, either interpolation-based (Sakawa's Criterion)

$$\max(\tilde{M}, \tilde{N}) \approx \max_I(\tilde{M}, \tilde{N}) = (\max(m_1, n_1), \max(m_2, n_2), \max(m_3, n_3))$$

or ranking-based (Lei's Criterion)

$$\max(\tilde{M}, \tilde{N}) \approx \max_R(\tilde{M}, \tilde{N}) = \begin{cases} \tilde{M}, & \tilde{N} <_R \tilde{M}, \\ \tilde{N}, & \text{otherwise.} \end{cases}$$

Most benchmark papers then report (at least) $\mathbb{E}[\tilde{C}_{\max}]$ as the makespan measure to minimize.

2. **Original fuzzy instances: hand-crafted FJSSP test beds.** A first benchmark group consists of «original» fuzzy instances, not derived by fuzzifying a crisp test bed, including small classical sets such as 6×6 and 10×10 families (often denoted $S6.*$ and $S10.*$), and larger originals such as 15×10 and 16×16 instances (often denoted Lei01–Lei02 and LP01). These are mainly used to compare metaheuristics under either $(\leq_E, \max_I)$ or $(<_R, \max_R)$ evaluation.

3. **Fuzzified crisp benchmarks, the main scalable source of instances.** A second benchmark group is built by fuzzifying processing times of well-known *crisp* JSSP benchmarks (notably FT, La, ABZ, ORB families), producing fuzzy variants that keep the original routing/structure but replace each crisp duration p_{ij} by a TFN $\tilde{p}_{ij}$.
4. **Fortemps-type fuzzification, often denoted with subscript F .** Across fuzzification methods, the modal value is set to the crisp duration, $p_{ij}^2 = p_{ij}$, while the support bounds are sampled at random in prescribed intervals. A generic template is

$$\tilde{p}_{ij} = (p_{ij}^1, p_{ij}, p_{ij}^3), \quad p_{ij}^1 \sim U([\underline{\delta} p_{ij}, p_{ij}]), \quad p_{ij}^3 \sim U([p_{ij}, \bar{\delta} p_{ij}]),$$

with method-specific choices of $(\underline{\delta}, \bar{\delta})$ and whether sampling is integer or real. The Fortemps-type variant highlighted in the benchmark literature is typically *symmetric*, meaning

$$p_{ij}^3 - p_{ij} = p_{ij} - p_{ij}^1.$$

This yields families such as FTF^* , LaF^* , $ABZF^*$.

5. **Tsujimura instances, often denoted with subscript TT .** Some papers introduce explicit fuzzified instances derived from FT problems (e.g., from FT06 and FT20), but without fully specifying the fuzzification rule. These instances are often evaluated with $(<_R, \max_R)$, but are less standard due to missing generation details.
6. **Non-symmetric fuzzification families, subscripts G, S, Z .** Several methods generate *non-symmetric* TFNs (so $p_{ij}^3 - p_{ij} \neq p_{ij} - p_{ij}^1$), still with $p_{ij}^2 = p_{ij}$, differing mainly in how the sampling intervals are defined and whether bounds are integer or real:
 - **Ghrayeb-type (G):** non-symmetric TFNs (bounds often real-valued).
 - **Song-type (S):** non-symmetric TFNs (introduced in due-date-satisfaction studies).
 - **Zheng-type (Z):** non-symmetric TFNs (bounds often integer-valued), widely reported for ORB/La/ABZ.
7. **Taillard-based fuzzy instances (TaF).** To address the fact that many existing fuzzy instances are nearly solved, a new benchmark is obtained by fuzzifying the hardest Taillard JSP instances, producing TaF21–30 (20×20) and TaF41–50 (30×20) using the Fortemps-type fuzzification idea. Lower bounds for expected makespan are inherited from best-known crisp lower bounds LB , and performance is often summarised by a relative error RE :

$$RE = \frac{|\mathbb{E}[C_{\max}] - LB|}{LB}.$$

8. **Large-Scale and Learning-Based Benchmarks (Fuzzified DMU & Generative Protocols).** The emergence of Neural Combinatorial Optimization (NCO) and Graph Neural Networks (GNNs) necessitates

benchmarks that exceed the scale of classical instances to prevent overfitting and test generalization [64].

- **Fuzzified DMU Instances:** Recent studies [64] utilize fuzzified versions of the Demirkol, Mehta, and Uzsoy (DMU) instances, ranging from 20×15 to 100×20 . These are characterized by tighter machine sequencing and are used to validate Graph Transformer architectures.
 - **Generative Protocols (EMARM):** To support deep reinforcement learning, static files are replaced by generative protocols. Wang et al. [41] introduced the Expectation-Maximization Algorithm-based Autoregressive Model (EMARM), which generates infinite streams of synthetic fuzzy instances to train neural agents for zero-shot generalization.
9. **Type-2 Fuzzy Benchmarks: Interval-Valued Uncertainty.** Addressing the criticism that TFNs model uncertainty too precisely, recent literature [58] introduces Interval Type-2 Fuzzy Sets (IT2FS). Benchmarks in this category, such as the **Li & Gao** and **Zheng** instances, extend the Kacem and Brandimarte flexible JSSP datasets. They define processing times not as single triplets but as Footprints of Uncertainty (FOU) bounded by Upper and Lower Membership Functions, often used in multi-objective energy-efficient scheduling.
 10. **Tarigan Flexible Packaging Dataset.** Introduced by Tarigan et al. [66], this is a FAIR-compliant dataset derived from real manufacturing logs. Unlike static benchmarks, it includes a timeline of dynamic events (machine breakdowns, rush orders, variable setup times), enabling the evaluation of algorithm resilience and stability in rolling horizon simulations.

7. DISCUSSION

This section consolidates the reviewed evidence into four themes and discusses how they collectively explain the field’s movement toward scalable, uncertainty-aware, and sustainability-oriented scheduling.

1. Benchmarks and Evaluation

- **Datasets and Fuzzification:** Standard benchmarks [22] remain prevalent for heuristics but lack the scale for modern learning approaches. Recent studies utilize large-scale datasets, such as fuzzified DMU instances for Graph Neural Networks [64], and generative protocols like EMARM for training neural agents [41]. Real-world datasets with dynamic disruptions have also been introduced [66].
- **Performance Metrics:** Evaluation criteria now prioritize sustainability (total energy consumption, carbon footprint) [17, 20], production quality [15], and robustness against uncertainties [23], moving beyond simple makespan minimization.

- **Scalability:** Current research explicitly addresses scalability using DRL and Graph Transformers with GPU acceleration to solve large-scale instances (100×20) prohibited for traditional methods [64, 65].

2. Trends and Comparative Analysis

- **Algorithmic Dominance:** Hybrid meta-heuristics dominate, with genetic algorithms and tabu search frequently combined to balance exploration and exploitation [25, 33, 49]. Pure heuristics are being replaced by memetic algorithms [6].
- **Learning-Based Scheduling:** A major shift involves integrating Reinforcement Learning (RL). Modern RL agents dynamically adapt scheduling policies [58, 59, 62], replacing static fuzzy logic controllers [30, 48].
- **Uncertainty Modelling:** Research is moving toward Interval Type-2 Fuzzy Sets (IT2FS) to model high-level uncertainty more accurately than triangular fuzzy numbers [5, 20, 52].

3. Applications

- **Manufacturing:** Models address specific constraints in aircraft shaft production [15], engine remanufacturing [18], and textile finishing [8].
- **Logistics:** Constraints now include crane transportation [38] and inter-machine travel [10].
- **Distributed Systems:** The scope includes distributed fuzzy blocking flow-shops for decentralized supply chains [44, 63].

4. Research Gaps and Future Directions

- **Ambiguity in Fuzzy Idle Time:** Inconsistent definitions of idle time in fuzzy environments yield divergent schedules; a standardized calculus is required [19].
- **Scalability of Generative Models:** Future work must focus on scalable generative environments [41] to train zero-shot schedulers, overcoming data bottlenecks [65].
- **Real-Time Adaptation:** Development of robust RL agents is needed for true real-time responses to random breakdowns, moving beyond periodic rescheduling [60, 66].
- **Hybrid Uncertainty:** Models should integrate fuzzy possibility theory and stochastic probability for cyber-physical systems [65].
- **Service-Oriented Scheduling Gaps:** A significant, not researched gap exists regarding service-oriented scheduling in high-stakes industries such as construction, mining, and autonomous power systems, where specialized maintenance is often outsourced to third-party service companies. Current FJSSP models primarily focus on the machine operator's perspective, failing to account for the dual-objective nature of service-provider scheduling: increasing the technical readiness of the equipment for the operator while maximizing the provider's own revenue through optimal resource utilization.

- **Fragmented Adoption of IT2FS:** While research is moving toward Interval Type-2 Fuzzy Sets (IT2FS) to model high-level uncertainty more accurately, their utilization remains fragmented. Future research should move from isolated applications to more comprehensive frameworks that standardize how IT2FS is used across diverse manufacturing and service sectors.
- **Omission of Structural Fuzziness:** Structural fuzziness remains entirely unaddressed within the current FJSSP literature. While processing times and due dates are modeled as fuzzy sets, studied literature does not yet address scenarios where the sets of indices themselves or the governing set of constraints are fuzzy. Developing a framework for structural fuzziness is essential for modeling truly fluid industrial environments where the boundaries of the problem definition are as uncertain as the data within them.
- **Lack of Digital Twin Integration for Closed-Loop Scheduling:** Current literature predominantly treats scheduling in JSSPs as an «open-loop» task—generating a predictive schedule that is executed until a disruption occurs. A critical gap remains in the integration of Digital Twin (DT) technology to enable closed-loop adaptive control. While standard dynamic scheduling reacts to discrete events (e.g., machine failure), a Digital Twin-integrated approach allows for the continuous synchronization of the virtual and physical worlds. Future research must explore how real-time sensor data can dynamically update the membership functions of fuzzy parameters (e.g., narrowing the «footprint of uncertainty» of processing times as an operation progresses) and utilize high-fidelity multi-physics simulations to verify the feasibility of fuzzy schedules against physical constraints before they are released to the shop floor. This technology is already used for Flexible JSSP, but not for FJSSP.

7. CONCLUSION

This literature review updates and extends the research map of the Fuzzy Job Shop Scheduling Problem for 2014–2025, using [1] as a baseline and charting recent model variants, objectives, solution approaches, and benchmarking practices. Over the last decade, the field has shifted from relatively compact fuzzy extensions toward broader, application-driven problem formulations where uncertainty is modeled across multiple elements, not only processing times, and where richer fuzzy constructs, including interval-valued and Type-2 representations, appear more often when ambiguity in the uncertainty itself must be captured [58]. In parallel, the objective landscape has expanded from predominantly fuzzy makespan to multi-objective settings that combine efficiency with robustness, stability, quality, and sustainability-related criteria, reflecting the realities of Industry 4.0/5.0 environments and motivating methods that can handle change rather than optimize a single static plan.

From the viewpoint of algorithms, hybrid metaheuristics remain widely used due to their flexibility, but machine learning-based scheduling has emerged as a major direction, including deep reinforcement learning and graph-based architectures that aim to scale and generalize across instance distributions [60, 64], supported by generative protocols that reduce dependence on scarce labeled data [41]. Benchmarking has also diversified toward fuzzified classical families, large-scale learning-oriented datasets, and dynamic evaluation settings [22, 66].

The authors declare that there is no conflict of interest concerning the publication of this article.

REFERENCES

1. Abdullah S., Abdolrazzaghe-Nezhad M. Fuzzy job-shop scheduling problems: A review. *Information Sciences*. 2014. V. 278. Iss. 9. P. 380–407.
2. Zhang X., Deng Y., Chan F.T.S., Xu P., Mahadevan S., Hu Y. IFSJSP: A novel methodology for the Job-Shop Scheduling Problem based on intuitionistic fuzzy sets. *International Journal of Production Research*. 2013. V. 51. P. 5100–5119.
3. Ahmadizar F., Zarei A. Minimizing makespan in a group shop with fuzzy release dates and processing times. *The International Journal of Advanced Manufacturing Technology*. 2013. V. 66. Iss. 9-12. P. 2063–2074. <https://doi.org/10.1007/s00170-012-4482-8>
4. Gao K.Z., Suganthan P.N., Pan Q.K., Tasgetiren M.F., Sadollah A. Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion. *Knowledge-Based Systems*. 2016. V. 109. No. 10. P. 1–16. <https://doi.org/10.1016/j.knosys.2016.06.014>
5. Li J.Q., Liu Z.M., Li C., Zheng Z.X. Improved Artificial Immune System Algorithm for Type-2 Fuzzy Flexible Job Shop Scheduling Problem. *IEEE Transactions on Fuzzy Systems*. 2021. V. 29. Iss. 11. P. 3234–3248. <https://doi.org/10.1109/TFUZZ.2020.3016225>
6. Wang C., Tian N., Ji Z., Wang Y. Multi-objective fuzzy flexible job shop scheduling using memetic algorithm. *Journal of Statistical Computation and Simulation*. 2017. V. 87. Iss. 14. P. 2828–2846. <https://doi.org/10.1080/00949655.2017.1344846>
7. Behnamian J., Fatemi Ghomi S.M.T. Multi-objective fuzzy multiprocessor flow-shop scheduling. *Applied Soft Computing Journal*. 2014. V. 21. P. 139–148. <https://doi.org/10.1016/j.asoc.2014.03.031>
8. Campo E.A., Cano J.A., Gomez-Montoya R., Rodriguez-Velasquez E., Cortes P. Flexible Job Shop Scheduling Problem with Fuzzy Times and Due-Windows: Minimizing Weighted Tardiness and Earliness Using Genetic Algorithms. *Algorithms*. 2022. V. 15. Iss. 10. P. 334. <https://doi.org/10.3390/a15100334>
9. Noori-Darvish S., Mahdavi I., Mahdavi-Amiri N. A bi-objective possibilistic programming model for open shop scheduling problems with sequence-dependent setup times, fuzzy processing times, and fuzzy due dates. *Applied Soft Computing*. 2012. V. 12. No. 4. P. 1399–1416. <https://doi.org/10.1016/j.asoc.2011.11.019>
10. Chen X., Li J., Han Y., Sang H. Improved artificial immune algorithm for the flexible job shop problem with transportation time. *Measurement and Control*. 2020. V. 53. Iss. 9-10. P. 2111–2128. <https://doi.org/10.1177/0020294020962130>

11. Li J., Pan Q. Chemical-reaction optimization for solving fuzzy job-shop scheduling problem with flexible maintenance activities. *International Journal of Production Economics*. 2013. V. 145. Iss. 1. P. 4–17. <https://doi.org/10.1016/j.ijpe.2012.11.005>
12. Gomes M.C., Barbosa-Pyvoa A.P., Novais A.Q. Optimal scheduling for flexible job shop operation. *International Journal of Production Research*. 2005. Iss. 11. P. 2323–2353. <https://doi.org/10.1080/00207540412331330101>
13. Liu B., Fan Y., Liu Y. A fast estimation of distribution algorithm for dynamic fuzzy flexible job-shop scheduling problem. *Computers & Industrial Engineering*. 2015. V. 87. P. 193–201. <https://doi.org/10.1016/j.cie.2015.04.029>
14. Petrovic S., Fayad C., Petrovic D., Burke E., Kendall G. Fuzzy job shop scheduling with lot-sizing. *Annals of Operations Research*. 2008. V. 159. Iss. 1. P. 275–292. <https://doi.org/10.1007/s10479-007-0287-9>
15. Li J., Guan Z. A multi-objective fuzzy flexible job shop scheduling problem considering the maximization of processing quality. *International Journal of Industrial Engineering Computations*. 2024. V. 15. Iss. 2. P. 491–502. <https://doi.org/10.5267/j.ijiec.2023.12.011>
16. Hemmati Far M., Haleh H., Saghaei A. A fuzzy bi-objective flexible cell scheduling optimization model under green and energy-efficient strategy using Pareto-based algorithms: SATPSPGA, SANRGA, and NSGA-II. *The International Journal of Advanced Manufacturing Technology*. 2019. V. 105. Iss. 9. P. 3853–3879. <https://doi.org/10.1007/s00170-019-03797-w>
17. Gonzalez-Rodriguez I., Puente J., Palacios J.J., Vela C.R. Multi-objective evolutionary algorithm for solving energy-aware fuzzy job shop problems. *Soft Computing*. 2020. V. 24. Iss. 21. P. 16291–16302. <https://doi.org/10.1007/s00500-020-04940-6>
18. Zhao J., Peng S., Li T., Lv S., Li M., Zhang H. Energy-aware fuzzy job-shop scheduling for engine remanufacturing at the multi-machine level. *Frontiers of Mechanical Engineering*. 2019. V. 14. Iss. 4. P. 474–488. <https://doi.org/10.1007/s11465-019-0560-z>
19. Gomez P.G., Gonzalez-Rodriguez I., Vela C.R. Computing Idle Times in Fuzzy Flexible Job Shop Scheduling. *Algorithms*. 2025. V. 18. Iss. 3. P. 137. <https://doi.org/10.3390/a18030137>
20. Shukla A.K., Nath R., Muhuri P.K., Lohani Q.M.D. Energy efficient multi-objective scheduling of tasks with interval type-2 fuzzy timing constraints in an Industry 4.0 ecosystem. *Engineering Applications of Artificial Intelligence*. 2020. V. 87. P. 100747. <https://doi.org/10.1016/j.engappai.2019.103257>
21. Cucala A.P., Fernandez A., Sicre C., Dominguez M. Fuzzy optimal schedule of high speed train operation to minimize energy consumption with uncertain delays and driver's behavioral response. *Engineering Applications of Artificial Intelligence*. 2012. V. 25. Iss. 8. P. 1548–1557. <https://doi.org/10.1016/j.engappai.2012.02.006>
22. Palacios J.J., Puente J., Vela C.R., Gonzalez-Rodriguez I. Benchmarks for fuzzy job shop problems. *Information Sciences*. 2016. V. 329. P. 736–752. <https://doi.org/10.1016/j.ins.2015.09.042>
23. Palacios J.J., Gonzalez-Rodriguez I., Vela C.R., Puente J. Robust swarm optimisation for fuzzy open shop scheduling. *Natural Computing*. 2014. Iss. 2. P. 145–156. <https://doi.org/10.1007/s11047-014-9413-1>
24. Palacios J.J., Gonzalez-Rodriguez I., Vela C.R., Puente J. Satisfying flexible due dates in fuzzy job shop by means of hybrid evolutionary algorithms. *Integrated*

- Computer-Aided Engineering*. 2018. Iss. 1. P. 65–84. <https://doi.org/10.3233/ICA-180583>
25. Vela C.R., Afsar S., Palacios J.J., Gonzalez-Rodriguez I., Puente J. Evolutionary tabu search for flexible due-date satisfaction in fuzzy job shop scheduling. *Computers & Operations Research*. 2020. V. 119. P. 104931. <https://doi.org/10.1016/j.cor.2020.104931>
 26. Saidi Mehrobad M., Pahlavani A. A fuzzy multi-objective programming for scheduling of weighted jobs on a single machine. *The International Journal of Advanced Manufacturing Technology*. 2009. V. 45. Iss. 1-2. P. 122–139. <https://doi.org/10.1007/s00170-009-1947-5>
 27. Hapke M., Jaszkievicz A., Slowinski R. Fuzzy project scheduling system for software development. *Fuzzy Sets and Systems*. 1994. V. 67. Iss. 1. P. 101–117. [https://doi.org/10.1016/0165-0114\(94\)90211-9](https://doi.org/10.1016/0165-0114(94)90211-9)
 28. Giannopoulos N., Moulitanitis V.C., Nearchou A.C. Multi-objective optimization with fuzzy measures and its application to flow-shop scheduling. *Engineering Applications of Artificial Intelligence*. 2012. Iss. 7. P. 1381–1394. <https://doi.org/10.1016/j.engappai.2012.06.011>
 29. Pan Z., Lei D., Wang L. A Bi-Population Evolutionary Algorithm With Feedback for Energy-Efficient Fuzzy Flexible Job Shop Scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2022. V. 52. Iss. 8. P. 2168–2216. <https://doi.org/10.1109/TSMC.2021.3120702>
 30. Hajri S., Liouane N., Hammadi S., Borne P. A controlled genetic algorithm by fuzzy logic and belief functions for job-shop scheduling. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*. 2000. V. 30. Iss. 5. P. 812–818. <https://doi.org/10.1109/3477.875454>
 31. Xu W., Ji Z., Wang Y. A flower pollination algorithm for flexible job shop scheduling with fuzzy processing time. *Modern Physics Letters B*. 2018. V. 32. Iss. 34n36. 1840113. <https://doi.org/10.1142/S0217984918401139>
 32. Wang L., Zhou G., Xu Y., Liu M. A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem. *International Journal of Production Research*. 2013. V. 51. Iss. 12. P. 3593–3608. <https://doi.org/10.1080/00207543.2012.754549>
 33. Lin J. A hybrid biogeography-based optimization for the fuzzy flexible job-shop scheduling problem. *Knowledge-Based Systems*. 2015. V. 78. No. 4. P. 59–74.
 34. Sun L., Lin L., Gen M., Li H. A Hybrid Cooperative Coevolution Algorithm for Fuzzy Flexible Job Shop Scheduling. *IEEE Transactions on Fuzzy Systems*. 2019. V. 27. Iss. 5. P. 1008–1022. <https://doi.org/10.1109/TFUZZ.2019.2895562>
 35. Lin J., Zhu L., Wang Z.-J. A multi-objective fuzzy flexible job shop scheduling problem considering the maximization of processing quality. *Computers & Industrial Engineering*. 2019. V. 127. No. 1. P. 1089–1100.
 36. Soto J., Melin P., Castillo O. A New Approach for Time Series Prediction Using Ensembles of IT2FNN Models with Optimization of Fuzzy Integrators. *International Journal of Fuzzy Systems*. 2018. Iss. 3. P. 701–728.
 37. Lei D. A Pareto archive particle swarm optimization for multi-objective job shop scheduling. *Computers & Industrial Engineering*. 2008. V. 54. Iss. 4. P. 960–971.
 38. Du Y., Li J., Li C., Duan P. A Reinforcement Learning Approach for Flexible Job Shop Scheduling Problem With Crane Transportation and Setup Times. *IEEE Transactions on Neural Networks and Learning Systems*. 2024. Iss. 4. P. 5695–5709. <https://doi.org/10.1109/TNNLS.2022.3208942>

39. Basan N.P., Coccola M.E., Del Valle A.G., Mendez C.A. An Effective Milp-based Decomposition Algorithm for the Integrated Design and Scheduling of Flexible Job-shop Plants. *Chemical Engineering Transactions*. 2019. V. 74. No. 5.
40. Xu Y., Wang L., Wang S., Liu M. An effective teaching-learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *Neurocomputing*. 2015. V. 148. No. 1. P. 260–268. <https://doi.org/10.1016/j.neucom.2013.10.042>
41. Wang Y., Guo T., Liu Z. An Expectation-Maximization Algorithm-based Autoregressive Model for the Fuzzy Job Shop Scheduling Problem. Arxiv. 2025. <https://doi.org/10.48550/arXiv.2502.00018>
42. Lin J. Backtracking search based hyper-heuristic for the flexible job-shop scheduling problem with fuzzy processing time. *Engineering Applications of Artificial Intelligence*. 2019. V. 77. P. 186–196. <https://doi.org/10.1016/j.engappai.2018.10.008>
43. Palacios J.J., Gonzalez-Rodriguez I., Vela C.R., Puente J. Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop. *Fuzzy Sets and Systems*. 2015. V. 278. No. 11. P. 81–97. <https://doi.org/10.1016/j.fss.2014.12.003>
44. Shao Z., Shao W., Pi D. Effective heuristics and metaheuristics for the distributed fuzzy blocking flow-shop scheduling problem. *Swarm and Evolutionary Computation*. 2020. V. 59. P. 100747. <https://doi.org/10.1016/j.swevo.2020.100747>
45. Puente J., Vela C.R., Gonzalez-Rodriguez I. Fast Local Search for Fuzzy Job Shop Scheduling. *ECAI 2010 - 19th European Conference on Artificial Intelligence*. 2010. P. 739–744. <https://doi.org/10.3233/978-1-60750-606-5-739>
46. Fattahi P., Jolai F., Arkat J. Flexible job shop scheduling with overlapping in operations. *Applied Mathematical Modelling*. 2009. V. 33. Iss. 7. P. 3076–3087.
47. Luhandjula M.K. Fuzzy optimization: An appraisal. *Fuzzy Sets and Systems*. 1989. V. 30. Iss. 3. P. 257–282. [https://doi.org/10.1016/0165-0114\(89\)90019-5](https://doi.org/10.1016/0165-0114(89)90019-5)
48. Yun Y.S. Genetic algorithm with fuzzy logic controller for preemptive and non-preemptive job-shop scheduling problems. *Computers & Industrial Engineering*. 2002. P. 623–644. [https://doi.org/10.1016/S0360-8352\(02\)00130-4](https://doi.org/10.1016/S0360-8352(02)00130-4)
49. Palacios J.J., Gonzalez M.A., Vela C.R., Gonzalez-Rodriguez I., Puente J. Genetic tabu search for the fuzzy flexible job shop problem. *Computers & Operations Research*. 2015. V. 54. P. 74–89. <https://doi.org/10.1016/j.cor.2014.08.023>
50. Palacios J.J., Puente J., Gonzalez-Rodriguez I., Vela C.R. Hybrid Tabu Search for Fuzzy Job Shop. *Natural and Artificial Models in Computation and Biology*, 2013. P. 376–385.
51. Bilkay O., Anlagan O., Kilic S.E. Job shop scheduling using fuzzy logic. *The International Journal of Advanced Manufacturing Technology*. 2004. V. 23. Iss. 7-8. P. 606–619. <https://doi.org/10.1007/s00170-003-1771-2>
52. Srinivasan A., Geetharamani G. Linear Programming Problem with Interval Type 2 Fuzzy Coefficients and an Interpretation for Its Constraints. *Journal of Applied Mathematics*. 2016. P. 1–11. <https://doi.org/10.1155/2016/8496812>
53. Kilic S., Kahraman C. Metaheuristic Techniques for Job Shop Scheduling Problem and a Fuzzy Ant Colony Optimization Algorithm. *Fuzzy Applications in Industrial Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. P. 401–425.
54. Kacem I., Hammadi S., Borne P. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic.

- Mathematics and Computers in Simulation*. 2002. V. 60. Iss. 3-5. P. 245–276. [https://doi.org/10.1016/S0378-4754\(02\)00019-8](https://doi.org/10.1016/S0378-4754(02)00019-8)
55. Kilic S. Scheduling a Fuzzy Flowshop Problem with Flexible Due Dates Using Ant Colony Optimization. *Applications of Evolutionary Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. P. 742–751.
56. Dubois D., Fargier H., Fortemps P. Scheduling Under Flexible Constraints and Uncertain Data: The Fuzzy Approach. *Production Scheduling*. Wiley, 2008. P. 301–332. <https://doi.org/10.1002/9780470611050.ch11>
57. Yuguang Z., Fan Y., Feng L. Solving multi-objective fuzzy flexible job shop scheduling problem using MABC algorithm. *Journal of Intelligent & Fuzzy Systems*. 2019. Iss. 2. P. 1455–1473. <https://doi.org/10.3233/JIFS-181152>
58. Li R., Gong W., Lu C. A reinforcement learning based RMOEA/D for bi-objective fuzzy flexible job shop scheduling. *Expert Systems with Applications*. 2022. V. 203. P. 117380. <https://doi.org/10.1016/j.eswa.2022.117380>
59. Wang Y.-F. Adaptive job shop scheduling strategy based on weighted Q-learning algorithm. *Journal of Intelligent Manufacturing*. 2020. V. 31. Iss. 2. P. 417–432. <https://doi.org/10.1007/s10845-018-1454-3>
60. Chen X., Li J., Xu Y. Q-learning based multi-objective immune algorithm for fuzzy flexible job shop scheduling problem considering dynamic disruptions. *Swarm and Evolutionary Computation*. 2023. V. 83. P. 101414. <https://doi.org/10.1016/j.swevo.2023.101414>
61. Deng L., Di Y., Wang L. A Reinforcement-Learning-Based 3-D Estimation of Distribution Algorithm for Fuzzy Distributed Hybrid Flow-Shop Scheduling Considering On-Time-Delivery. *IEEE Transactions on Cybernetics*. 2024. Iss. 2. P. 1024–1036. <https://doi.org/10.1109/TCYB.2023.3336656>
62. Chang X., Jia X., Ren J. A reinforcement learning enhanced memetic algorithm for multi-objective flexible job shop scheduling toward Industry 5.0. *International Journal of Production Research*. 2025. Iss. 1. P. 119–147. <https://doi.org/10.1080/00207543.2024.2357740>
63. Shao Z., Shao W., Chen J., Pi D. MQL-MM: A Meta-Q-Learning-Based Multi-objective Metaheuristic for Energy-Efficient Distributed Fuzzy Hybrid Blocking Flow-Shop Scheduling Problem. *IEEE Transactions on Evolutionary Computation*. 2025. Iss. 4. P. 1183–1198. <https://doi.org/10.1109/TEVC.2024.3399314>
64. Zhang W., Zhao F., Feng B., Mei X. A Reinforcement Learning Control Framework Based on Scalable Graph Transformer for Large-Scale Fuzzy Job Shop Scheduling Problems. *IEEE Transactions on Neural Networks and Learning Systems*. 2025. Iss. 9. P. 16521–16533. <https://doi.org/10.1109/TNNLS.2025.3569868>
65. Smit I.G., Zhou J., Reijnen R., Wu Y., Chen J., Zhang C., Bukhsh Z., Zhang Y., Nuijten W. Graph Neural Networks for Job Shop Scheduling Problems: A Survey. *Arxiv*. 2024. <http://arxiv.org/abs/2406.14096>
66. Tarigan M., Gaol F.L., Gunawan A.A.S., Budiharto W. Advancing Dynamic Flexible Job Shop Scheduling With a Real-World Dataset and Genetic Adaptive Scheduling System. *IEEE Access*. 2025. V. 13. P. 193020–193031. <https://doi.org/10.1109/ACCESS.2025.3630184>

Received: 15.01.2026 / Revised: 28.02.2026 /
Accepted: 13.03.2026 / Published: 24.04.2026