BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **25**, No 2
Sofia • 2025 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2025-0009

Methods and Algorithms for Flexible Job Shop Scheduling – A State of the Art

Vassil Guliashki[1] ✉ *[0000-0003-3221-7469]*, *Leoneed Kirilov*[1] *[0000-0001-8424-9070]*
Galia Marinova[2] *[0000-0002-8804-5662]*

[1]*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia 1113, Acad. G. Bonchev str. 2, Bulgaria*
[2]*Technical University of Sofia, Sofia, Bulgaria*
E-mails: vassil.guliashki@iict.bas.bg leoneed.kirilov@iict.bas.bg gim@tu-sofia.bg

Abstract: *The Job Shop Scheduling Problem* (*JSSP*) *attracts many researchers due to its combinatorial nature and its discovery in numerous practical applications. This type of problem is characterized by high computational complexity; therefore, solving large-sized problems is not accessible with exact optimization methods. Very often, real JSSP problems can be presented as Flexible Job Shop Scheduling Problems* (*FJSSP*)*. For these problems, there are single-criterion and multi-criteria mathematical models. On the other hand, the ways to solve this type of problems include exact methods and heuristic or metaheuristic algorithms. This paper the aim to review the progress of research in the field of solving FJSSP over the last 10 years, as well as to show current trends for future scientific developments in this area.*

Keywords: *Flexible job shop problems, Optimization models, Exact methods* (*Meta-Heuristic algorithms*)*, Trends and directions for future research.*

1. Introduction

Usually, the JSSP considers a set N of n jobs, $N = \{N_1, N_2, ..., N_n\}$, and a set M of m machines, $M = \{M_1, M_2, ..., M_m\}$. The job i consists of a set of n_i operations in a determined sequence, denoted by $O_{i;j}$ ($i = 1, 2, ..., n; j = 1, 2, ..., n_i$), where i is the job number and j is the number of the operation in this job. The operation $O_{i;j}$ should be processed on one predefined machine $M_k \in M$. The processing times for each operation $O_{i;j}$ on a given machine – $t_{i;j,k}$, are predefined. In the real world, very often a production flexibility is desired and necessary. Many real-world problems can be formulated as flexible job shop problems [1-3]. Different variants of these tasks are known, especially concerning optimization criteria and the initial and/or final conditions for the execution of the jobs or the machine park description. For this reason, the class FJSSP is a broader class of problems, compared to the classical JSSP. The Flexible Job Shop Scheduling problem is a complex scheduling problem that extends the classic Job Shop Scheduling Problem by introducing flexibility in the assignment of operations to machines. Each operation in a job can be processed on multiple machines, each with different efficiencies or processing times. More precisely, each operation in the FJSSP is performed on a machine M_k from the subset M_{ij} of machines $M_k \in M_{ij}$, where $M_{ij} \subset M$. In case that $M_{ij} \subset M$ is fulfilled for at least one operation, there is available a Partial flexibility FJSSP (P-FJSSP); and when $M_{ij} \equiv M$ for each operation, there is Total flexibility FJSSP (T-FJSSP). The goal is typically to optimize objectives like minimizing the *makespan* (this is the maximal completion time), *total completion time*, or *tardiness*, while adhering to constraints such as machine availability and operation precedence.

Several different criteria can be defined for FJSSP [4]. They can be classified into three groups depending on the parameters with which they are measured: the times when the execution of the jobs according to the schedule is completed, the predefined times for the completion of the jobs and machine load and usage estimates.

The first group includes criteria based on the time determining the completion of the execution of the jobs:

1. Maximum time to complete the execution of all jobs in the schedule (makespan) – $C_{\max} = \max\limits_{j=1,.n}\{C_j\}$;

2. Maximum time that the jobs must wait for the start of processing,

$$W_{\max} = \max_{j=1,n}\{W_j\} = \max_{j=1,n}\{r_j - S_j\} = \max_{j=1,n}\{C_j - r_j - p_j\},$$

where r_j is the readiness time of the job N_j for processing, p_j is the execution time of job N_j;

3. Maximum time between the entry and completion of the jobs (flow time),

$$F_{\max} = \max_{j=1,n}\{F_j\} = \max_{j=1,n}\{C_j - r_j\},$$

4. Estimates of the total times, average times, or if weights are set for each job, and weighted total or average estimates of the above times for all jobs in the schedule are also used.

The second group includes criteria based on the use of machines or equipment, which criteria must be minimized:

1. Average number of jobs/operations that have been waiting for a machine to be released, $\overline{N}_w$.

2. Average downtime of machines, $\overline{I}$.

3. Maximum downtime of any of the machines, $I_{\max}$.

4. Average evaluation of machine utilization, $\overline{U} = \sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}\frac{p_{ij}}{mC_{\max}}$, where p_{ij} is the execution time of operation $O_{i;j}$.

The third group combines criteria based on the set times for completing jobs, which must be minimized:

1. Maximum deviation of the completion time of jobs from their preset time in the schedule, $L_{\max} = \max\limits_{j=1,.n}\{L_j\} = \max\limits_{j=1,.n}\{C_j - d_j\}$, where d_j is the planned time for completion of the job N_j (due date);

2. Maximum delay time of completing jobs in the schedule,

$$T_{\max} = \max_{j=1,n}\{T_j\} = \max_{j=1,n}\{0, L_j\};$$

3. Maximum time of early completion of jobs compared to the preset time in the schedule, $E_{\max} = \max_{j=1,n}\{E_j\} = \max_{j=1,n}\{0,-L_j\}$;

4. Estimates of the total above times, average times, or weighted (if weights are set for each activity) total or average estimates of the above times for all jobs in the schedule are also used.

Predefined evaluation functions of the various parameters can also be used as criteria [5, 6]. A broad overview of solution techniques (exact, heuristic and metaheuristic) is presented in the book [5], and it is useful for understanding the landscape of job-shop scheduling research. Other surveys on methods and algorithms for scheduling and flexible scheduling problems are [7, 8, 59]. Exact methods and heuristic algorithms for different scheduling problems are considered in details in the book [6]. It focuses more on presenting scheduling algorithms and evaluation metrics rather than a specific problem on the topic. The book outlines several types of evaluation functions commonly used in job-shop scheduling, such as:

- Makespan $C_{\max}$ (the total time to complete all jobs);
- Total tardiness (total delay from due dates);
- Total completion time;
- Flowtime (total time a job spends in the system).

These evaluation functions are used to assess the quality of scheduling solutions and can be incorporated into various scheduling approaches, including exact and heuristic methods.

It is proven that JSSP is a NP-hard optimization problem [9]. FJSSP as an extension of JSSP is also NP-hard [10]. Multiple objective FJSSP is also NP-hard [11].

2. Exact methods

These methods guarantee optimal solutions but they are computationally expensive for examples of large size, due to computational complexity of the problems (NP-hard). For this reason, they are unattractive for solving problems in a real time, and are not applicable for large-sized tasks. They are useful when the time for problem solving is not restricted. Also, the optimal solutions generated by the exact methods can serve for comparison with the solutions received by approximate and heuristic algorithms when an evaluation of their solution quality is necessary.

2.1. Mixed-Integer Nonlinear or Linear Programming (MINP or MILP)

The formulation of the FJSSP-model is in the form of a linear mixed-integer optimization problem. Improvements in MILP solvers have made these approaches more viable for medium-sized instances. Several research works that focus on MILP formulations and techniques for the FJSSP are [12-20]. The work [13] discusses real-world applications and integrates insights on flexibility in scheduling. Paper [14] proposes a MILP model for FJSSPs with separable sequence-dependent setup times.

This model is very economic because the number of constraints and the binary and continuous variables is greatly reduced.

Paper [15] is a study of energy-aware FJSSPs. In these problems, a shutdown (on/off) strategy is applied when the idle time for the machine(s) is too long. The authors propose a MILP model. The goal is to minimize makespan and total energy consumption. The MILP model is a development of the models of Choi and Choi (2002) and Meng et al. [37]. A constrained programming model for the same problem is also formulated. The experiments show better performance for CP.

Two exact approaches – MILP and CP are proposed for solving FJSSPs with partial or total flexibility in [16]. The objective is to minimize the total completion time. The efficiency of the methods is tested on a large set of instances. In particular, the CP approach outperforms the MILP approach for large-sized instances. The work [17] presents a MILP model tailored to FJSSP without setup and transportation times. The objective is makespan. This model is more compact in comparison with the model proposed by Özgüven-Özbakır-Yavuz (2010) and it proposes better performance. An innovative MILP formulation that may improve computational efficiency compared to traditional methods is proposed in [18]. This study is focused on solving specific FJSSP challenges. The MILP model proposed in [19] incorporates sequence-dependent setup times, making the model more realistic for industrial applications. It enhances the ability to handle complex scheduling constraints. The disadvantage of this model is that its computational complexity increases due to the inclusion of setup times. The MILP model in [20] integrates tool assignment into the FJSSP framework, addressing a critical aspect of real-world scheduling. This study provides a detailed and flexible approach to solving complex scheduling problems. One disadvantage of the model is that the tool assignment adds additional constraints, increasing computational demands. Also, it may require additional domain-specific customization for practical applications.

In [21], several MILP models are proposed for solving "Distributed" FJSSP (DFJSP) with one objective – to minimize makespan (maximum completion time). The first model is called a sequence-based model. It is an extension of the MILP model for FJSSP proposed in Roshanaei-Azab-ElMaraghy (2013). This model introduces three decision variables: for the precedence relationship between two operations, for machine selection, and factory selection. The second model is called a position-based model. It is based on the ideas presented in Naderi-Azab (2014) for solving the Dynamic Job Shop Scheduling Problem (DJSSP) and in Fattahi-Saidi Mehrabad-Jolai (2007) for solving FJSSP. This model introduces a position decision variable for machine selection and sequencing problems. The third model develops the idea for time-indexed modeling following the ideas in Matta (2009) – MILP model for MultiProcessor Open Shop (MPOS) scheduling problem, and [26] – MILP model for Energy-Aware Scheduling (EAS) problem. This third model introduces two time-indexed binary decision variables to determine the machine selection problem and the sequencing problem. Another important parameter is max planned time. The number of constraints and the number of variables depend on this parameter. The fourth model is called the adjacent sequence-based model. It develops the ideas presented in Naderi-Azab (2014) – MILP model for Distributed JSSP and

Mousakhani-Morteza (2013) – MILP model for FJSP. An adjacent precedence decision variable is introduced to define precedence between two adjacent operations on the one machine. A lower bound for *C*max is added to the above models according to the model in De Giovanni-Pezzella (2010). From the computational experiments, it follows that the "sequence-based MILP model" is the most efficient.

A variant of FJSSP with several sequence-dependent constraints (setup times, transportation, and assignment restrictions) is solved in [22]. The objective is to minimize the sum of the production times of the jobs. A MILP model is developed and solved by the Gurobi solver. A similar problem is also solved in [23]. The authors consider FJSSP with sequence-dependent setup and transportation times. The objective is to minimize the total energy consumption, considering the turn off/on strategy. This criterion is nonlinear, and therefore it is linearized and simplified. In this way, a MILP model is constructed. It is solved by the branch-and-cut method (a combination of cutting plane and branch-and-bound methods) from CPLEX solver. The experiments show that the solving time exponentially increases with instance size, i. e., this approach is suitable for small-sized instances.

Several studies consider FJSSPs with operators' participation. A real application for the textile industry (sewing process) is done in [24]. A FJSSP problem in the MILP format is proposed with several setup times (machine-change for operators, color-change, and configuration-change). The objective is to minimize the total tardiness of jobs. The model is solved by CPLEX. It is shown that this approach is suitable for small-sized instances. Further, increasing the skills of operators leads to better solutions. Two models – MILP and CP are formulated for FJSSP, including machine operators in [25]. The objective is to minimize makespan. The computational experiments on CPLEX show that CP outperforms by speed and quality of solutions MILP model for small instances. It is shown also that CP outperforms the metaheuristic approach "Knowledge-guided fruit fly optimization algorithm – KF" proposed by Zheng and Wang (2016). The operators' participation and the influence of the learning effect on processing times are studied in [27, 28]. The authors formulate FJSSP with sequencing flexibility and position-based learning effect. It is assumed that the precedence relations between operations are nonlinear. The problem is modelled as MILP and CP formats, and two constructive heuristics are proposed for building a feasible solution. The objective is to minimize makespan. Solving CP models with an exact method (CPLEX) yields an optimal solution to a large set of instances. Another type of integrated FJSSP (job scheduling and operator scheduling) is studied in [29]. MILP and CP models are formulated. The objective function is makespan. Accordingly, they are solved by the Gurobi solver and the IBM ILOG CP optimizer. Here, too, both models do not work well for large-sized instances, although the MILP model gives better solutions.

A time-indexed model and competitive iterative procedure for solving FJSSP are presented in [30]. The model is in MILP format. The objective function is a linear combination of makespan and tardiness. But the experiments are done separately for makespan and tardiness. The authors emphasize that the choice of an objective function that is appropriate for the real problem being solved is very important for the scheduling model. It is also shown that the time-indexed model solved with an appropriate iterative method can give much better optimal solutions than other models, and for large-sized instances.

A FJSSP with repeated jobs and operations is studied in [31]. The authors model a real-life problem for electronic components production in this form. A CP formulation for the problem is presented and solved by CPLEX with makespan as the objective function. Some seven simple heuristics are also proposed. The experiments show better performance of CP compared to heuristics.

A MILP model with two objectives to minimize – makespan and carbon emissions is proposed in [32] for FJSSP. The model takes into account energy consumption and operator learning effects. To solve the model, the authors propose an Improved Multiobjective Sparrow Search Algorithm (IMOSSA). But the model is also solved by the exact method on CPLEX.

A FJSSP with two flows of jobs is considered in [33] – direct flow from the first stage to the last stage and reverse flow of jobs from the last stage to the first. A MILP model with an objective makespan is proposed. It is solved by the exact branch and bound method and by the meta-heuristic Vibration Damping Optimization (VDO). It is shown that for small-sized instances, both methods find optimal solutions, but the exact method works faster. For large-sized instances, VDO has better performance. In the second case, VDO is also compared with a Genetic Algorithm (GA), and it shows again better performance.

Paper [34] proposes a Mixed Integer Goal Programming (MIGP) model with two objective functions – makespan and the total machining time. The model is solved by applying a pre-emptive goal programming approach and the branch-and-bound method. The experiments show that this exact approach is competitive with metaheuristic methods for small-sized instances.

In [35], an extension of FJSSP is proposed. It refers to a more generalized representation of the precedence relation between operations forming a given job. Usually, the precedence relation is linear. The authors define the precedence relation in terms of a directed acyclic graph. Then, a MILP model is proposed. The objective function is makespan. A variant of FJSP is considered in [36]. The set of work centers forming the shop consists of a machine set, which is linearly ordered and has restricted accessibility. When a machine is busy in the work center, the succeeding machines are not accessible. This is called FJSP with blockages. A MILP model is formulated for this problem, and two methods for solving heuristic-greedy method and the exact branch and bound method. The objective function is makespan.

Several MILP models for solving FJSP with minimizing total energy consumption are proposed in [37]. They are developed by modelling idle energy (linear) and idle time (it is non-linear and subsequently linearized by adding decision variables and constraints). Three models are based on idle energy, and three models are based on idle time. They are studied by the CPLEX solver. Experiments show their advantages over other existing models. In general, computational scalability is a common challenge for MILP-based methods. Hybrid approaches, while promising, often require fine-tuning and may not generalize across all instances.

The multiple objective FJSSP with Controllable Processing Times (CPT) is studied in [38]. The authors consider two objectives: to minimize "*C*max and total energy consumption. A Mixed Integer Linear Programming (MILP) model is developed, and then the epsilon method is used to obtain the optimal Pareto front for small-scale instances. To obtain approximate Pareto fronts for medium- and large-sized problems, an efficient multi-objective hybrid shuffled frog-leaping algorithm (MOsingle bondHSFLA) is proposed.

2.2. Constraint Programming (CP)

A constrained programming model is proposed in [21]. It leverages constraint satisfaction techniques to find feasible schedules by pruning the search space effectively and to find an optimal solution for large-sized problems in a relatively short time. It works by introducing an interval decision variable and a sequence decision variable [39]. Further, the CP model outperforms many methods known in the literature, and it is suitable for small- and large-sized problems. A CP approach is also proposed by Ham and Cakici [40] for FJSSP with parallel batch processing machines. An enhanced MIP model is proposed; to decrease decision space, several inequalities are added to this model, and a CP model is formulated. The testing shows a significant reduction in computational time and the superiority of the CP model. Fekih et al. [41] propose MILP and CP models for partial and total FJSSP. Testing the models shows their effectiveness and the superiority of the model for large-sized FJSSPs.

A comparison of computational performance between MIP and CP approaches for solving scheduling problems is done by Ku and Beck [42]. The results show similar behaviour of both approaches for scheduling problems with up to medium size. But CP outperforms MIP for large-sized problems.

Many authors claim the advantages of the Constrained Programming approach when solving FJSSP. In this regard, Müller et al. [43] propose an algorithm to choose the best CP solver depending on the specific FJSSP being solved. The algorithm is based on deep neural networks and decision trees. Five CP solvers are tested, and IBM ILOG CPLEX CP Optimizer and Google's OR-Tools are selected as the best ones.

Aschauer et al. [44] suggest an approach for easier managing a "multi-constraint scheduling task". They consider FJSSP with "no-wait constraints" and "multiple complex constraints", minimizing *C*max. The test results are promising.

2.3. Branch and Bound (B&B)

Systematically explores possible schedules using bounds to eliminate suboptimal solutions.

A parallel "Branch and Bound (B&B) algorithm" for solving "multi-objective FJSSP" is proposed in [45]. Several important contributions of the method are: one vector presentation of the order of execution and the operation-machine allocation; novel grid representation of the solution space; a concurrent priority queue for storing the pending computing tasks (sub-problems for chosen area in the search space); Generalized Integer-Vector-Matrix (EIVM) representation for MO-FJSSP. The Pareto front of thirteen instances from the literature is computed for the first time. The authors consider minimizing three objectives: makespan, max workload, and total workload.

2.4. Dynamic Programming (DP)

Breaks the problem into smaller subproblems to compute the optimal solution incrementally.

Over the last decade, FJSP has been a rich area of research, especially with advancements in optimization techniques, including Dynamic Programming (DP) [46-49]. The paper [46] discusses using DP for solving FJSP with certain simplifications to make the problem computationally feasible. In [47], the use of DP in the context of unexpected machine failures is extended. This approach is more realistic and applicable to industrial scenarios. The dynamic programming framework enables adaptive rescheduling when machines fail. Incorporating failure considerations improves scheduling reliability and minimizes disruptions. The main disadvantage of this approach is that the state-space explosion due to incorporating machine failure scenarios limits its application to smaller or medium-sized FJSSPs. The authors of [48] investigate multi-objective optimization in FJSP using dynamic programming. The model simultaneously optimizes multiple conflicting objectives, such as minimizing makespan, tardiness, and energy consumption, providing a balanced solution for real-world applications. The approach is flexible and can incorporate additional objectives or constraints, depending on specific scheduling needs. The inclusion of energy consumption as an objective makes it relevant to modern sustainable manufacturing practices. Main disadvantages of this approach are: (i) Multi-objective optimization adds layers of complexity to the DP model, making it slower and more resource-intensive, the approach is inapplicable to large problems; (ii) Balancing multiple objectives requires careful parameter tuning or prioritization, which can be challenging for practitioners.

DP methods for handling more complex scheduling scenarios are considered in [49]. The inclusion of parallel machines increases the flexibility and applicability of the model to real-world shop floors. The authors introduce optimization strategies to reduce the computational complexity of the DP algorithm, making it more practical for larger instances. The method is adaptable to dynamic environments, as parallel machines provide redundancy and flexibility in scheduling. Disadvantages: (i) While the model addresses parallel machines, the complexity of coordination between these machines can still be computationally intensive; (ii) The approach may not fully address other aspects of FJSSPs, such as dynamic changes in job priorities or machine breakdowns; (iii) The optimizations introduced to enhance DP efficiency might compromise the ability to explore all potential scheduling solutions.

In conclusion, DP approaches are highly effective in providing accurate and structured solutions for FJSSPs, especially in scenarios with complex constraints like machine failures or parallel machines. The main disadvantage is that DP's computational intensity and lack of scalability make it more suitable for small-to-medium problem sizes unless enhanced or hybridized.

3. Heuristic algorithms

Heuristics are designed for faster solutions, often sacrificing optimality for efficiency. Some studies from the last decade focus on heuristic algorithms based on

dispatching rules, such as Shortest Processing Time (SPT) or Earliest Due Date (EDD), for solving the FJSSP. These algorithms are often used as part of heuristic approaches in job shop scheduling to achieve near-optimal solutions in practical settings. Malika and Kalla [50] have proposed several fault-tolerant scheduling heuristics, minimizing the schedule length in the case of the absence or presence of faults. They can be applied to solving FJSSP.

3.1. Shortest Processing Time (SPT)

The study [51] considers the JSSP with sequence-dependent setup times. Two models are formulated: 1) single objective with makespan minimization; 2) multiple objective with a bi-criteria objective function. A hybrid algorithm combining a Genetic Algorithm (GA) and a Variable Neighborhood Search (VNS) is proposed. The effectiveness of this algorithm is evaluated by comparing its results with the results of other methods, "demonstrating the superiority of the developed algorithm over the existing algorithms" in terms of solution quality. The paper [52] investigates a hybrid heuristic method incorporating the SPT rule, among others, to solve the FJSP. The authors integrate sequence-dependent setup times and propose a heuristic for optimizing job processing sequences. The heuristic algorithm is tested on benchmark instances, showing its effectiveness and competitiveness compared to other existing algorithms. The FJSSP model, including due windows, sequence-dependent setup times, and uncertainty in processing and setup times, is considered in [53]. The authors use a genetic algorithm to solve this problem and integrate fuzzy logic to minimize the weighted penalties for tardiness/earliness. The developed algorithm is used in a fabric finishing production system. The results are compared with four heuristics: Monte Carlo simulation, shortest processing time, critical reason, and earliest due date. In more than 30% of the cases, the proposed algorithm outperforms these heuristics.

3.2. Earliest Due Date (EDD)

During the last decade, there have not works focusing solely on the Earliest Due Date (EDD) approach for JSSP. Important factors that influence the performance of dispatching rules are the average flow allowance, due-date assignment algorithms, and the progress milestones implementation. These factors interact with the dispatching rules for generating effective JSSP solutions. The role and effectiveness of the EDD rule in flexible job shop scheduling within dynamic and uncertain environments are considered in detail in [54-56]. The study [54] introduces a dynamic scheduling method for flexible job shops utilizing a "MachineRank" algorithm. The approach aims to prioritize machines based on certain criteria to optimize scheduling efficiency. By ranking machines, the method can effectively allocate tasks to the most suitable machines, potentially reducing overall processing time. The approach can adjust to changing conditions in the job shop, such as machine availability or job urgency, enhancing responsiveness. Prioritizing machines helps in balancing the workload, leading to better utilization of resources. The main disadvantage of this approach is that developing and maintaining a dynamic ranking system may require sophisticated algorithms and real-time data processing. Second, the effectiveness of the MachineRank system relies heavily on the accuracy and timeliness of data regarding machine performance and job requirements. The research work [55] applies Deep Reinforcement Learning (DRL) to address scheduling in flexible job-shop environments characterized by dynamic changes, such as machine breakdowns or varying job priorities. DRL enables the system to learn and adapt to unforeseen changes in the environment, maintaining scheduling efficiency. The approach can handle multiple objectives and constraints, finding optimal or near-optimal solutions in complex scenarios. Through learning mechanisms, the system can improve its performance over time. Disadvantages: (i) Training DRL models demands significant computational resources and time; (ii) Effective learning requires a large amount of data representing various scenarios, which may not always be available; (iii) In highly dynamic or novel situations, the DRL model might produce suboptimal or unexpected scheduling decisions. The study [56] explores the use of machine learning techniques to tackle job shop scheduling problems under uncertainty, focusing on developing robust schedules that can accommodate variability in job processing times and other unpredictable factors. The approach aims to create schedules that remain effective despite variations and uncertainties in the job shop environment. By leveraging historical data, the model can identify patterns and make informed scheduling decisions. Disadvantages: (i) The success of the model depends on the availability and quality of data capturing the uncertainties present in the environment; (ii) Developing a model that accurately captures the complexities of uncertainty in scheduling can be challenging and resource-intensive; (iii) The model may struggle to generalize to scenarios significantly different from those encountered during training.

3.3. Greedy algorithms

These algorithms prioritize operations based on predefined criteria, optimizing one decision at a time. During the last 10 years, several studies have applied greedy algorithms to address the FJSSP. Some of them are listed below:

A modified Iterated Greedy (IG) algorithm to tackle the FJSSP is proposed in [57]. The classical IG is divided into two phases, each addressing a sub-problem of FJSSP: sequencing and routing. Dispatching rules are employed in the construction phase for sequencing and machine selection. Experiments on benchmark instances demonstrate that the proposed algorithm is competitive, often finding global optima, and is simpler and less computationally intensive compared to more complex methods.

The study [58] introduces a meta-heuristic method combining genetic and greedy algorithms to optimize performance criteria in FJSSP. To improve the efficiency of the genetic algorithm, the initial population is generated using a greedy algorithm, and several elitist operators are applied to generate better solutions. The greedy algorithm prioritizes cells and jobs within each cell, yielding quality solutions. Testing on the P-FJSP dataset shows that the proposed method outperforms Non-dominated Ranked Genetic Algorithm (NRGA) and Non-dominated Sorting Genetic Algorithm II (NSGA-II) in terms of diversity, spacing, quality, and runtime.

These works illustrate the effectiveness of incorporating greedy algorithms, either standalone or in combination with other meta-heuristics, in solving FJSSP.

4. Metaheuristic algorithms

These complex heuristics balance exploration and exploitation to escape local optima and find near-optimal solutions.

4.1. Genetic Algorithms (GAs)

GAs mimic natural selection to iteratively improve a population of solutions. Several studies have applied GAs to address the FJSSP in the past decade. A comprehensive survey of Gas and hybrid Gas for solving the FJSSP is presented in [59]. Some recent examples are listed below:

A self-learning genetic algorithm for FJSSP is proposed in [60]. It integrates an improved population initialization procedure and an optimized crossover strategy. The authors improved the single mutation approach of the genetic algorithm. Four mutation operators were proposed based on process coding and machine coding. Their selection of mutation operators was iteratively adjusted. As a result, the local search procedure was enhanced, and the convergence speed was accelerated. The novel algorithm generates good-quality solutions on a sample of benchmark examples. Advantages: (i) The proposed algorithm enhances convergence speed compared to traditional GAs by using advanced genetic operators and selection techniques; (ii) The approach generates high-quality solutions for complex scheduling problems by effectively exploring the search space. Disadvantages: (i) GAs can be computationally expensive for large-sized problems due to their iterative nature and population-based approach; (ii) The algorithm's performance depends on carefully chosen parameters (e.g., population size, mutation rate), which can require extensive tuning; (iii) Despite improvements, there is still a possibility of the algorithm converging to suboptimal solutions in highly complex problem spaces.

A genetic algorithm with improvements is proposed in [61] to overcome the weak searching ability and long running time. First, a new generation mechanism is proposed to produce the initial population, which leads to better convergence speed. Second, the mutation operation is changed to avoid the generation of illegal solutions. In this manner, the running time of the algorithm is reduced. The comparison with other algorithms showed the better performance of this novel improved algorithm. Disadvantages: (i) While both improvements reduce the computational overhead, the overall runtime can still be high for large-scale problems; (ii) The effectiveness of the algorithm can vary depending on the specific problem instance and the chosen parameter settings.

The work [62] introduces a genetic algorithm with priority-based representation for FJSSP, one of the most challenging operations research problems. The study examines the impact of the proposed representation schema on FJSSP. Each gene on the chromosome represents the priority of an operation, utilized by a constructive algorithm during decoding. To enhance solutions, iterated local search is applied post-reproduction. By guiding the GA with priorities, the approach reduces unnecessary exploration and accelerates convergence. The method applies to a wide range of scheduling scenarios, including multi-objective and dynamic environments. Benchmarking against widely used FJSSP datasets indicates that the proposed GA performs comparably or better in terms of makespan relative to existing literature.

Disadvantages: (i) The algorithm's performance heavily depends on the choice and design of priority rules, which may not generalize across all problem instances; (ii) Integrating priority rules adds complexity to the GA, requiring careful design and validation; (iii) For large-sized problems, the combined computational demands of the GA and priority rule evaluation may limit its efficiency.

4.2. Simulated Annealing (SA)

Uses a probabilistic approach to escape local optima by allowing worse solutions temporarily.

This kind of algorithm uses a probabilistic approach to escape local optima by allowing worse solutions temporarily. Some research works on SA for solving FJSSP are considered briefly as follows:

An accelerated simulated annealing algorithm enhanced by a partial scheduling mechanism and a cooling schedule based on standard deviation is presented in [63]. The approach aims to rapidly converge to high-quality solutions for the FJSSP. The acceleration techniques reduce the computational time required to achieve such (near) optimal solutions. Experimental results on benchmark instances indicate that the proposed method achieves faster convergence to optimal or near-optimal solutions compared to traditional simulated annealing algorithms. The algorithm performs well in avoiding local optima, thanks to the stochastic nature of simulated annealing. It is adaptable to various FJSSP configurations and constraints, making it suitable for diverse industrial scenarios. Simulated annealing is relatively easy to implement, and the accelerated version retains this simplicity. Disadvantages: (i) The performance depends on careful tuning of parameters, such as cooling schedules and initial temperatures; (ii) Despite the acceleration, the algorithm may struggle with very large problems due to computational intensity; (iii) The quality of the initial solution can significantly affect the algorithm's performance.

In [64], an FJSSP with limited buffers and step-deteriorating jobs is considered, including multiple non-identical parallel machines. The authors formulated a mixed integer programming model, where the makespan and total tardiness have to be minimized simultaneously. An effective hybrid meta-heuristic algorithm is proposed, named GVNSA. It combines GA, SA, and Variable Neighborhood Search (VNS). The obtained results demonstrate that GVNSA generates better quality solutions compared to other heuristic and meta-heuristic algorithms. The hybrid approach effectively addresses multiple constraints, including limited buffers and job deterioration, making it suitable for complex real-world problems. Disadvantages: (i) Combining multiple techniques increases the complexity of the algorithm, making it harder to implement and understand; (ii) The performance depends heavily on proper parameter tuning for each combined component.

The authors of [65] introduce a simulated-annealing-based hyper-heuristic (SA-HH) for the FJSSP, focusing on assembling Heuristic Schemes (HS) that consist of Machine Assignment Rules (MARs) and Job Sequencing Rules (JSRs) alongside problem state features. Two variants of SA-HH are investigated: one incorporating problem state features and one without. The proposed approach demonstrates superior performance in minimizing makespan compared to benchmark algorithms on standard datasets. Advantages: (i) The hyper-heuristic framework dynamically selects and combines heuristics, improving flexibility and adaptability; (ii) The combination of simulated annealing and hyper-heuristic techniques ensures robust and high-quality solutions; (iii) The hyper-heuristic framework reduces the need for manual algorithm design and tuning. Disadvantages: (i) The framework's adaptability and dynamic heuristic selection can result in significant computational overhead; (ii) While the method adapts well within the FJSSP context, its generalizability to other scheduling problems might be limited without significant adjustments.

These studies highlight the effectiveness of simulated annealing algorithms in solving the flexible job shop scheduling problem over the past decade.

4.3. Particle Swarm Optimization (PSO)

The PSO algorithms are inspired by the social behaviour of birds and adjust the solutions based on the best positions found by a swarm. Some recent works on PSO techniques for FJSSP are considered in brief below:

A new algorithm, named EPSO, for solving FJSSP based on PSO is proposed in [66]. EPSO includes features such as particle life cycle and a discrete position update mechanism to expand the solution space and avoid premature convergence. The objective is to minimize makespan, and benchmarking against 20 well-known instances demonstrates that EPSO performs equally well or better than existing methods. Enhancements in the PSO algorithm improve the balance between exploration (global search) and exploitation (local refinement). The algorithm demonstrates faster convergence compared to traditional optimization techniques. It is designed to minimize computational overhead while maintaining solution quality. Disadvantages: (i) The performance heavily depends on the proper tuning of PSO parameters, such as inertia weight and learning coefficients; (ii) There is a risk of the algorithm converging prematurely to local optima, especially for complex or large-scale FJSSPs.

The authors of [67] apply a distributed PSO algorithm to solve FJSSP with the aim of minimizing makespan. Various benchmark data, including Partial and Total FJSSP, are tested. The received results demonstrate that the novel PSO is effective and efficient. Additionally, the study enables real-time decision-making in response to resource states and unforeseen events. Advantages: (i) Parallel processing significantly reduces computation time, making it suitable for large-scale FJSSPs; (ii) Distributed agents increase the diversity of the search, reducing the likelihood of premature convergence; (iii) The algorithm consistently produces high-quality solutions for complex scheduling problems. Disadvantages: (i) The distributed nature of the algorithm increases implementation complexity and requires specialized hardware or software; (ii) The performance depends on the availability of computational resources for parallelization; (iii) Managing communication and synchronization between distributed agents can introduce additional computational overhead.

A variable neighborhood descent hybrid genetic algorithm (VND-hGA) for FJSSP is proposed in [68] to overcome the low convergence speed and to improve the accuracy of the genetic algorithm. The algorithm integrates a hybrid heuristic initialization strategy, A BareBones Particle Swarm Optimization (BBPSO)-based mutation operator, and VND based on an improved multilevel neighborhood structure. The advantages of BBPSO are maximized by means of a real-number-based chromosome representation, special crossover method, coding, and decoding. The comparison with existing algorithms shows superior solution accuracy and convergence performance of the novel algorithm.

These studies highlight the application of PSO algorithms in solving the Flexible Job Shop Scheduling Problem in the last ten years. In many cases PSO is used to improve the convergence speed and to reduce the computation time.

4.4. Ant Colony Optimization (ACO)

This technique uses the behaviour of ants to explore multiple scheduling pathways and reinforce optimal solutions. Some examples for its application on FJSSP are listed as follows:

The work [69] proposes an improved ACO algorithm to solve the FJSSP, aiming to minimize makespan. The approach introduces a novel pheromone updating mechanism and a local search strategy to enhance solution quality. Experimental results on benchmark instances demonstrate the effectiveness of the proposed method in finding optimal or near-optimal solutions. This is a progress compared to basic ACO, characterized by low computational efficiency and generating a local optimum.

The authors of [70] present the application of an ACO algorithm for scheduling operations in flexible job shop systems with multi-resource requirements. The study focuses on operations that require multiple resources, such as machines and personnel, and proposes a metaheuristic schema to find optimal scheduling solutions under these constraints.

The paper [71] develops a hybrid algorithm combining Ant Colony System (ACS) and Iterated Local Search (ILS) to solve the FJSSP with the objective of *C*max minimizing. The proposed algorithm is tested on benchmark instances, demonstrating its effectiveness in finding high-quality solutions compared to existing methods.

The study [72] addresses a multiple objective FJSSP with multiple time constraints, including setup time, transportation time, and delivery time. The objective is simultaneously minimizing the total workload, the workload of critical machine, the maximum completion time, and the penalties of earliness/tardiness. An improved ACO algorithm is proposed, featuring a distributed coding approach and initialization methods to enhance solution diversity and quality. The algorithm's performance is validated through experiments on 28 benchmark instances, showing its effectiveness in solving complex FJSSP instances.

The study [73] proposes a model of the JSSP as a complete graph and uses ACO to explore the search space. The results on several instances illustrate the good performance of the model with the corresponding selection of parameters.

These studies highlight the ongoing research efforts in applying and enhancing ACO algorithms to effectively solve the complexities associated with flexible job shop scheduling problems over the past decade.

4.5. Tabu Search (TS)

This metaheuristic uses a memory-based approach to avoid revisiting recently explored solutions. Some works devoted to TS for FJSSP are listed as follows:

A hybrid algorithm combining Genetic Algorithm (GA) and Tabu Search (TS) is proposed for the flexible job shop scheduling problem with sequence-dependent set-up times and job lag times in [74]. The genetic algorithm is used for a global search, and the Tabu search – for precise local search. Very good performance of the hybrid GA-TS algorithm is shown on two classes of instances based on classical data sets. The results show that the proposed hybrid algorithm is efficient for such kind of problems. Advantages: (i) The algorithm effectively addresses sequence-dependent set-up times and job lag times; (ii) Combining genetic algorithms with local search methods improves the exploration and exploitation balance, leading to high-quality solutions; (iii) The framework can be adapted to various real-world scheduling scenarios. Disadvantages: (i) The hybrid approach increases algorithmic complexity, making implementation and parameter tuning more challenging; (ii) The performance may vary depending on the specific scheduling constraints.

The evolutionary searching ability of GA with local improvement ability of TS are combined in [75] to balance the exploration and exploitation. The resulting hybrid algorithm is effective for FJSSP with objective function minimizing the makespan. Six benchmark instances and 201 open problems have been tested to evaluate the performance of the developed heuristic. The experimental Gantt charts demonstrate that the proposed algorithm has significant improvement of the solution accuracy and the computational time compared to other algorithms for FJSSP. Advantages: (i) The combination of GA and TS ensures a robust search process, with GA providing global exploration and TS refining the solutions locally; (ii) The approach can handle multiple objectives and complex problem constraints effectively; (iii) The use of Tabu search reduces the likelihood of the algorithm getting trapped in local optima. Disadvantages: (i) The hybridization introduces more parameters (e.g., GA crossover rate, TS tabu list length) that require careful tuning for optimal performance; (ii) The combination of two metaheuristics can result in higher computational costs compared to single-heuristic approaches; (iii) The algorithm may struggle with scalability when applied to extremely large or highly complex problems due to its intensive search process.

A decentralized model based on tabu search is formulated in [76] to solve the Distributed and Flexible Job shop Scheduling Problem (DFJSP). The performance of the model is validated on popular benchmark instances.

4.6. Other metaheuristic approaches

A robust flexible job-shop scheduling problem with uncertain operation processing times with an uncertainty budget is studied in [77]. The objective function is makespan. A two-stage robust optimization approach is proposed. At the first stage the assignment and the sequence of operations on machines are defined and at the second stage the operation start-times resp. Three approaches are proposed for solving the problem. For the first one the authors develop Mixed Integer Linear Programming (MILPext) extended robust model which is an extension of the sequence-based model proposed in [78]. For the second one, a Constraint Programming (CPext) is proposed following [25]. The third one is column and Constraint Generation approach (CCG) [79] with different combination of the above

MILPext and CPext. All approaches are tested on a large set of instances. The best combination turns out to be CCG-CPext-MILPext.

A distributed FJSSP for modeling multi-factory production is studied in [80]. The objective function is makespan. The authors propose a new Mixed-Integer Linear Programming (MILP) model. Its solution by exact methods is suitable for small-sized instances. A hybrid approach GA-VNS-CP including genetic algorithm (GA), Variable Neighborhood Search (VNS) and Constraint Programming (CP) methods is also proposed. Computational experiments show the outperformance of the new hybrid method in comparison with other ones. With the new hybrid method, improved solutions have been found for a number of benchmark instances.

A hybrid framework for solving FJSSP is proposed in [81]. The objective is to minimize Total Energy Consumption (TEC). The hybrid framework includes Gene Expression Programming (GEP), Variable Neighborhood Search (VNS) and local sequence-based Mixed Integer Linear Programming (MILP). Experiments are performed with medium-sized instances. It is shown that better solutions have been found with a significant reduction in computation time.

Thi et al. [82] propose a hybrid algorithm to handle scheduling problems considering machine breakdowns, aiming to reduce makespan and machine idle time in cases of random failure.

The study [83] introduces a customized multi-objective evolutionary algorithm NSGA-III for solving flexible job shop scheduling problems, showing excellent performance with reduced computational effort. It combines smart initialization approaches to generate good first population, various crossover operators and different local search strategies.

Paper [84] addresses a real-world scheduling problem in the printing industry, proposing populational metaheuristic approaches combined with a local search strategy. The algorithm developed is suitable for solving large-sized instances.

Borissova and Mustakerov [85] propose a parallel algorithm for FJSSP with complex constraints, where the processing of part of details is independent, and the others have a "fixed processing order". The original problem is decomposed in linear programming tasks, which are solved in parallel. The parallel organization of the computations allows solving larger FJSSPs. The algorithm is tested on a real-world problem.

In the last years Big Data connected with scheduling attract the attention of many researchers. A survey of the scheduling mechanisms in the Big Data cases is presented in [86].

5. Advances in FJSSP optimization in the last decade

5.1. Hybrid methods

Such methods are combining exact methods with metaheuristics or combining multiple metaheuristics to improve solution quality and efficiency. In the past decade, several studies have integrated Mixed-Integer Linear Programming for initial solution generation with Genetic Algorithms for refinement to address the "Flexible Job Shop Scheduling Problem". Notable examples include:

The chapter [87] discusses a hybrid genetic algorithm and simulated annealing approach tested on benchmark problems, yielding better performance in minimizing the makespan. An acceptance criterion is incorporated into the crossover operator. 162 benchmark problems have been solved demonstrating the better performance of this hybridization compared to other metaheuristic algorithms.

The paper [88] proposes an efficient scheduling method for the FJSSP using a Genetic Algorithm (GA) that incorporates heuristic rules. The scheduler's goal is to minimize mean tardiness. Two types of decision-making are required: machine selection and job selection. Heuristics combining five job selection and five machine selection rules are studied. The test experiments show that the combination of specific machine and job selection rules provides the best performance under different shop conditions when incorporated into the GA. Other result is that applying GA only for either machine selection or job selection can generate good schedules.

A paper based on the Gray Wolf Optimization (GWO) algorithm in combination with the Spiral Search (SS) mechanism of Whale Optimization algorithm is proposed in [10]. The authors apply a convergence strategy, which increases the convergence speed and improves the accuracy of the algorithm. The effectiveness of the proposed SS-GWO model is proven on 22 flexible job shop scheduling benchmark problems. A comparison is made with five other modern algorithms, and it is shown that the performance of SS-GWO is better.

The study [89] addresses the FJSSP with job-splitting. The number of sub-lots each job should be split into, as well as the size of each sub-lot, are determined. A Mixed-Integer Programming (MIP) model is proposed, where the objective is the makespan minimization. The size of sub-lots and their number are predefined (or bounded). For large-sized problems the mathematical model cannot find feasible solutions. For this reason, a Hybrid Genetic Algorithm (HGA) is proposed, including a Local Search Algorithm (LSA) to determine the size of sub-lots and to improve the GA efficiency. The performance of the novel HGA is compared with the classical GA, showing its effectiveness for large-size FJSSP examples.

5.2. Artificial intelligence and machine learning

Artificial Intelligence (AI) techniques predict bottlenecks and dynamically adjust scheduling parameters. Reinforcement learning has been explored for adaptive scheduling strategies.

A survey on JSSP with application of AI is presented in [90].

In the past decade, several studies have applied Machine Learning (ML) techniques to address the Flexible Job Shop Scheduling Problem (FJSSP), Some notable instances are:

A novel Deep Reinforcement Learning (DRL) method for solving FJSSP, particularly for large instances, is introduced in [91]. The approach utilizes heterogeneous graph neural networks to create a more informative representation of the problem, enhancing decision-making capabilities. Additionally, the study proposes generating a diverse set of scheduling policies and combining DRL with dispatching rules to constrain the action space. The test results indicate that this method outperforms traditional dispatching rules and other state-of-the-art DRL methods.

The paper [92] investigates the application of Deep Reinforcement Learning (DRL) to solve the FJSSP, considering the maximum completion time as an objective function. The authors propose a DRL algorithm to optimize scheduling decisions, demonstrating its effectiveness in handling the complexities of FJSSP.

The authors of [93] present a novel end-to-end learning framework that combines self-attention models for deep feature extraction with DRL for scalable decision-making. They propose a Dual-Attention Network (DAN) to capture complex relationships between operations and machines, achieving high-quality scheduling decisions. Experimental results show that this approach outperforms traditional priority dispatching rules and state-of-the-art DRL methods.

The study [94] integrates Constraint Programming (CP) within a deep learning framework to solve FJSSP dynamically. By training a deep learning model using optimal solutions generated by CP, the authors enhance the model's performance without extensive exploration typical in DRL. The hybrid approach demonstrates superior performance over state-of-the-art DRL methods and a widely used CP solver.

Machine learning models are investigated in [95] to predict capacity consumption in a flexible job-shop environment. The authors propose several ML models, including linear regression variants, decision trees, and artificial neural networks, to estimate makespan. Numerical experiments demonstrate that these models outperform traditional exact approaches and dispatching rules, especially when computation time is limited.

These studies highlight the growing application of AI and ML techniques in enhancing the efficiency and effectiveness of flexible job shop scheduling over the past decade.

5.3. Multi-objective optimization

It is characteristic of this type of methods and algorithms that they have an increased focus on balancing trade-offs among conflicting objectives, such as minimizing makespan while maximizing machine utilization. During the last ten years numerous studies have addressed the Flexible Job Shop Scheduling Problem using multi-objective optimization methods. Some of them are listed below, as follows:

A teaching-and-learning-based hybrid genetic (GA) – Particle Swarm Optimization (PSO) algorithm to address the Multi-Objective FJSSP (MOFJSP) is proposed in [96]. The novel algorithm combines PSO with GA and has the advantages of both approaches. The learning capability of the GA is improved by means of an external memory library storing elite individuals. The proposed algorithm is tested on benchmark instances, and the results show its effectiveness in solving MOFJSP.

A "multi-objective FJSSP with transportation constraints" is solved in [97] by means a proposed improved Multi-Objective Wolf Pack Algorithm (MOWPA). According to the green manufacturing principles, the formulated model minimizes the total energy consumption and the maximum completion time. The developed algorithm is compared with other algorithms on standard examples. The test results show its "superior performance in solving FJSSP".

A novel method of integrating simulated modelling and Multi-Criteria Decision Making (MCDM) methods is proposed in [98]. The authors formulated a Discrete Event Simulation (DES) Model for defining the "job priorities". They attack large-size problems with multiple criteria. A partial FJSSP is modelled, and the multiple criteria include evaluation of Makespan, Flow Time, and Tardiness-based measures considering static and dynamic job arrivals. The solutions are generated using best-performing Composite Dispatching Rules (CDR) in combination with several Priority Dispatching Rules (PDR).

5.4. Cloud and parallel computing

These techniques are leveraging distributed systems and cloud platforms to solve larger instances faster by parallelizing computation. In the past decade, several studies have addressed the FJSSP by leveraging cloud and parallel computing technologies. Several examples are cited below:

The authors of [99] propose a novel "cloud-based bacterial foraging optimization algorithm" to solve multi-objective FJSSP. The developed algorithm includes minimizing the makespan and machine workload and utilizes cloud computing to improve its computational efficiency. Experimental results demonstrate the effectiveness of this algorithm in solving complex flexible job shop scheduling problems.

The research [100] explores the application of quantum computing to FJSSP by proposing a Quadratic Unconstrained Binary Optimization (QUBO) model to minimize the makespan. The model is solved using a Coherent Ising Machine (CIM), and numerical experiments demonstrate that quantum computing holds significant potential for solving FJSSPs more efficiently than traditional computational methods.

The study [101] addresses the FJSSP by incorporating parallel batch processing machines to minimize the maximum completion time. The authors propose a solution that combines variable neighborhood search with multi-population genetic algorithms, conducting neighborhood searches on elite populations to reduce the likelihood of local optima. The approach was evaluated using real production scenarios, demonstrating its effectiveness in practical applications.

The paper [102] proposes a dual island genetic algorithm consisting of a parallel cellular model and a parallel pseudo model to address large-scale flexible flow shop scheduling problems. The two-level parallelization is highly consistent with the underlying architecture and is well-suited for parallelizing inside or between GPUs and multi-core CPUs. Computational results show that the proposed method obtains competitive results and reduces execution time.

The study [103] proposes a dynamic energy-efficient flexible flow shop scheduling model using peak power value, considering new arrival jobs. A priority-based hybrid parallel genetic algorithm with a predictive reactive complete rescheduling approach is developed. Designed to be highly consistent with NVIDIA's CUDA software model, the approach achieves better performance than traditional static approaches and significantly reduces time requirements.

The study [104] considers FJSSP with parallel machines, minimizing the makespan. Efficient algorithms are proposed, including parallel computing techniques to improve the computational speed and the solution quality. The results show that the "proposed algorithms are effective for large-scale scheduling problems".

The above-listed works highlight the integration of cloud and parallel computing technologies in developing efficient algorithms for FJSSP, contributing to advancements in solving complex scheduling problems.

5.5. Dynamic flexible job shop scheduling

Such algorithms are developed to deal with real-time uncertainties like machine breakdowns, job arrivals, and processing times.

The research work [105] presents "a methodology for both static and dynamic scheduling". A method is proposed, using a "hybrid algorithm" to optimize the "static FJSSP" and the "Dynamic FJSSP (DFJSSP)". Simulated annealing as a local search procedure and a genetic algorithm as "a global optimization technique" are combined in this algorithm. Within this framework, a "Variable Neighborhood Search (VNS)" is also included "for efficient neighborhood search". The algorithm is tested on 40 benchmark test instances. In the DFJSP framework, dynamic events such as "single job arrival, single machine breakdown, multiple job arrivals, and multiple machine breakdowns" are considered. By applying a rescheduling strategy, the novel hybrid algorithm receives a "significant improvement" in the solution quality. Disadvantages: (i) Combining two computationally intensive methods (SA and GA) leads to higher runtime, especially for large-scale problems; (ii) The hybrid model introduces additional parameters (e.g., GA crossover/mutation rates, SA temperature schedule), which require careful tuning for optimal performance.

The authors of [106] present an evolutionary multi-task optimization framework combined with genetic programming to tackle the dynamic FJSSP. This approach addresses the complexities arising from dynamic and uncertain manufacturing environments where new tasks are continually introduced.

The research [107] proposes a hierarchical and distributed architecture to address DFJSSP, utilizing a Double Deep Q-network algorithm to train scheduling agents. The approach captures the relationship between production information and scheduling decisions, facilitating real-time control in agile and flexible production environments.

A Markov Decision Process (MDP) model for the flexible job shop scheduling problem with a single Autonomous Mobile Robot (AMR) is defined in [108]. The authors suggest a heuristic algorithm applying dynamic programming to solve problem instances with multiple AMRs. The proposed algorithm is validated by comparison with various algorithms.

DFSSP with setup time and random job arrival is considered in [109]. An improved gene expression programming algorithm, including a dynamic scheduling framework, is proposed to construct scheduling rules. The authors test 24 groups of instances with different scales. The results show that the improved gene expression programming is better than the standard gene expression programming.

Two new approaches are proposed for extracting composite priority rules for FJSSP in [110]. The first approach uses a multigenic system, adding modified and operational features of the scheduling environment to the terminal set. The second approach uses priority rules as automatically defined functions. They are combined with the cellular system. This gene expression programming generates better solutions than the first approach, but both approaches with extracted rules yield better results than the rules of other known methods.

These studies illustrate the importance of the Dynamic approach for effectively solving FJSSPs.

6. Conclusion

Over the last decade, the field of Flexible Job Shop Scheduling Problems has seen significant advancements in both exact methods and approximate algorithms, reflecting a balance between precision and scalability. The major trends are outlined as follows:

1. The exact methods can be characterized by increasing mathematical model complexity. Researchers are developing advanced mathematical models incorporating real-world constraints like sequence-dependent setup times, machine breakdowns, and multi-objective optimization. Mixed Integer Linear Programming (MILP) and Constraint Programming (CP) continue to dominate, but their applicability is often limited to small or medium-sized problem instances due to computational complexity. Another important trend is the hybridization. Exact methods are being increasingly combined with heuristic/metaheuristic techniques to handle larger problem instances. For example, branch-and-bound algorithms are sometimes paired with heuristics for faster convergence. Some exact methods are being adapted to optimize multiple objectives simultaneously (e.g., minimizing makespan, tardiness, and energy consumption). Techniques like Pareto-based MILP are gaining traction. The adoption of parallel computing and advanced solvers has extended the feasibility of exact methods for larger-scale problems.

2. In the field of research on approximate algorithms for FJSSP, the metaheuristics dominate. Metaheuristics like "Genetic Algorithms (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO)" have been extensively studied. Researchers focus on improving convergence speed and solution quality through problem-specific adaptations. Combining multiple metaheuristics (e.g., GA-PSO, SA-ACO) or integrating metaheuristics with local search and dispatching rules has become common to enhance performance. Hybridization leverages the exploration capability of one method and the exploitation capability of another. Hyper-heuristic frameworks, which combine rule-based systems and metaheuristics, have emerged as a promising area for solving FJSSP with dynamic and uncertain conditions. Recent trends show the incorporation of machine learning techniques (e.g., reinforcement learning) to improve heuristic performance and adapt strategies based on problem characteristics. Approximate algorithms are favored for their scalability, making them suitable for dynamic and real-time scheduling applications.

Directions for future research:

- **Automation.** Combining machine learning with metaheuristics to automate the selection of algorithmic parameters and improve adaptability to diverse FJSSP instances.

- **New objectives.** With increased emphasis on sustainable manufacturing, algorithms are incorporating energy efficiency and environmental impact as additional objectives.
- **Dynamic scheduling.** Algorithms are evolving to accommodate real-time data, stochastic disruptions, and adaptive scheduling in Industry 4.0 environments.

In summary, while exact methods maintain relevance for small-scale problems and theoretical insights, approximate algorithms, especially hybrid and machine learning-augmented approaches, dominate practical applications due to their flexibility, scalability, and ability to address real-world complexity.

***Acknowledgments*:** The authors express their gratitude to the Bulgarian National Science Fund, supporting this work by the project "Mathematical models, methods and algorithms for solving hard optimization problems to achieve high security in communications and better economic sustainability", Grant No KP-06-N52/7.

References

1. B o r i s s o v a, D. Optimal Scheduling for Dependent Details Processing Using MS Excel Solver. – Cybernetics and Information Technologies, Vol. **8**, 2008, No 2, pp. 102-111.
2. V a s s i l e v a, M. Operative Planning of the Production Program in a Textile Enterprise with the Help of MKO-1 Software System. – Cybernetics and Information Technologies, Vol. **6**, 2006, No 1, pp. 58-68.
3. V a s s i l e v a, M., B. S t a y k o v. Operative Planning of the Production Program with the Help of MKO-2 Software System. – Cybernetics and Information Technologies, Vol. **9**, 2009, No 4, pp. 110-120.
4. K i r i l o v, L., V. G u l i a s h k i, K. G e n o v a. Multicriteria Decision Making in Manufacturing Scheduling (in Bulgarian). Prof. Dr. Ivan Mustakerov, Ed. Education Ltd., 2016. 281 p. ISBN 978-954-552-074-7.
5. H a j a r i w a l a, D. C., S. S. P a t i l, S. M. P a t i l. A Review of Metaheuristic Algorithms for Job Shop Scheduling. – Engineering Access, Vol. **11,** 2024, No 1, pp. 65-91.
6. B r u c k e r, P. Scheduling Algorithms. 3rd Edition. Springer, 2007. ISBN 978-3-540-73053-4.
7. G u l i a s h k i, V., L. K i r i l o v, A. N u z i. Optimization Models and Strategy Approaches Dealing with Economic Crises, Natural Disasters, and Pandemics – an Overview. – Cybernetics and Information Technologies, Vol. **23**, 2023, No 4, pp. 3-25.
8. G e n o v a, K., L. K i r i l o v, V. G u l i a s h k i. A Survey of Solving Approaches for Multiple Objective Flexible Job Shop Scheduling Problems. – Cybernetics and Information Technologies, Vol. **15**, 2015, No 2, pp. 3-22.
9. S a n t o s, H. G., T. A. M. T o f f o l o. Mixed Integer Linear Programming with Python. 2020. 73 p.
https://readthedocs.org/projects/python-mip/downloads/pdf/latest/
10. Z h o u, K., C. T a n, Y. Z h a o, J. Y u, Z. Z h a n g. Research on Solving Flexible Job Shop Scheduling Problem Based on Improved GWO Algorithm SS-GWO. – Neural Processing Letters, Vol. **56**, 2024, No 26. DOI: 10.1007/s11063-024-11488-1.
11. X i e, J., L. G a o, K. P e n g, X. L i, L. H a o r a n. Review on Flexible Job Shop Scheduling. – IET Collaborative Intelligent Manufacturing, Vol. **1**, 2019, No 2. DOI: 10.1049/iet-cim.2018.0009**.**
12. Y u, Y. A Research Review on Job Shop Scheduling Problem. – E3S Web of Conferences, Vol. **253**, 2021, No 1, 02024. DOI:10.1051/e3sconf/202125302024.
13. D a u z è r e-P é r è s, S., J. D i n g, L. S h e n, K. T a m s s a o u e t. The Flexible Job Shop Scheduling Problem: A Review. – European Journal of Operational Research, Vol. **314**, 2024, No 2, pp. 409-432. ISSN 0377-2217. DOI: 10.1016/j.ejor.2023.05.017.
14. D r i s s, E., R. M e l l o u l i, W. H a c h i c h a. Mixed Integer Programming for Job Shop Scheduling Problem with Separable Sequence-Dependent Setup Times. – American Journal of Mathematical and Computational Sciences, Vol. **3**, 2018, pp. 31-36.
https://www.researchgate.net/publication/322901435

15. H a m, A., M.-J. P a r k, K. M. K i m. Energy-Aware Flexible Job Shop Scheduling Using Mixed Integer Programming and Constraint Programming. – Mathematical Problems in Engineering, Vol. **2021**, 2021, Article ID 8035806. 12 p. DOI: 10.1155/2021/8035806.
16. F e k i h, A., H. H a d d a, I. K a c e m, A. B. H a d j-A l o u a n e. Mixed-Integer Programming and Constraint Programming Models for the Flexible Job Shop Scheduling Problem. – In: T. Masrour, I. El Hassani, N. Barka, Eds. Artificial Intelligence and Industrial Applications, A2IA 2023, Lecture Notes in Networks and Systems. Vol. **771**. Cham, Springer, 2023. DOI: 10.1007/978-3-031-43524-9_8.
17. Z i a e e, M. A Mixed Integer Linear Programming Model for Flexible Job Shop Scheduling Problem, World Academy of Science, Engineering and Technology. – International Journal of Mathematical and Computational Science, Vol. **12**, 2018, No 6, pp. 91-95.
18. G ü n l ü k, O., U. B i l g e. A New MILP Formulation for the Flexible Job-Shop Scheduling Problem. – European Journal of Operational Research, Vol. **237**, 2014, No 3, pp. 926-939. DOI: 10.1016/j.ejor.2014.02.039.
19. L i, Y., R. J i a o. A Mixed-Integer Linear Programming Model for the Flexible Job-Shop Scheduling Problem with Sequence-Dependent Setup Times. – Journal of Manufacturing Systems, Vol. **41**, 2016, pp. 1-15. DOI: 10.1016/j.jmsy.2016.07.002.
20. V i d a l, T., A. R. A l v e s. A Mixed-Integer Linear Programming Model for the Flexible Job-Shop Scheduling Problem with Tool Assignment. – Computers & Operations Research, Vol. **106**, 2019, pp. 82-94. DOI: 10.1016/j.cor.2018.03.006.
21. M e n g, L., C. Z h a n g, Y. R e n, B. C. L. Z h a n g. Mixed-Integer Linear Programming and Constraint Programming Formulations for Solving Distributed Flexible Job Shop Scheduling Problem. – Computers & Industrial Engineering, Vol. **142**, 2020, 106347. ISSN 0360-8352. DOI: 10.1016/j.cie.2020.106347.
22. V a r o n e, S., D. S c h i n d l, C. B e f f a. Flexible Job Shop Scheduling Problem with Sequence-Dependent Transportation Constraints and Setup Times. – In: Proc. of 16th Conference on Computer Science and Intelligence Systems ACSIS (FEDCSIS'21), Vol. **26**, 2021, pp. 97-102. ISSN: 2300-5963. DOI: 10.15439/2021F14.
23. M e n g, L., B. Z h a n g, K. G a o, P. D u a n. An MILP Model for Energy-Conscious Flexible Job Shop Problem with Transportation and Sequence-Dependent Setup Times. – Sustainability, Vol. **15**, 2023, No 1, 776. DOI: 10.3390/su15010776.
24. P e r r o u x, T., T. A r b a o u i, L. M e r g h e m-B o u l a h i a. The Flexible Job Shop Scheduling Problem with Setups and Operator Skills: An Application in the Textile Industry. – IFAC-PapersOnLine, Vol. **58**, 2024, Issue 19, pp. 1090-1095. ISSN 2405-8963. DOI: 10.1016/j.ifacol.2024.09.125.
25. K r e s s, D., D. M ü l l e r. Mathematical Models for a Flexible Job Shop Scheduling Problem with Machine Operator Constraints. – IFAC-PapersOnLine, Vol. **52**, 2019, Issue 13, pp. 94-99. ISSN 2405-8963. DOI: 10.1016/j.ifacol.2019.11.144.
26. B r u c k e r, P., S. S c h u l z, J. S. N e u f e l d, U. B u s c h e r. A Multi-Objective Iterated Local Search Algorithm for Comprehensive Energy-Aware Hybrid Flow Shop Scheduling. – Journal of Cleaner Production, Vol. **224**, 2019, pp. 421-434. ISSN 0959-6526. DOI: 10.1016/j.jclepro.2019.03.155.
27. A r a u j o, K., E. B i r g i n, D. R o n c o n i. A Constraint Programming Model for the Flexible Job Shop Scheduling Problem with Sequencing Flexibility and Position-Based Learning Effects. – Simpósio Brasileiro de Pesquisa Operacional (SBPO'22), Rio de Janeiro, Brazil, 2022, pp. 2965-1476.
https://proceedings.science/sbpo/sbpo-2022/trabalhos/a-constraint-programming-model-for-the-flexible-job-shop-scheduling-problem-with?lang=pt-br
https://repositorio.usp.br/item/003172659
28. A r a u j o, K., E. B i r g i n, D. R o n c o n i. Models, Constructive Heuristics, and Benchmark Instances for the Flexible Job Shop Scheduling Problem with Sequencing Flexibility and Position-Based Learning Effect. – arXiv:2403.16766 [math.OC], 2022. DOI: 10.48550/arXiv.2403.16766.
29. S a b e r, R. G., P. L e y m a n, E.-H. A g h e z z a f. Modeling the Integrated Flexible Job-Shop and Operator Scheduling in Flexible Manufacturing Systems. – IFAC-PapersOnLine, Vol. **58**, 2024, Issue 19, pp. 343-348. ISSN 2405-8963. DOI: 10.1016/j.ifacol.2024.09.235.

30. A l m g r e n, T., M. P a t r i k s s o n, A. S t r ö m b e r g, K. T h ö r n b l a d. A Competitive Iterative Procedure Using a Time-Indexed Model for Solving Flexible Job Shop Scheduling Problems. – Optimization Online, 2015.
https://optimization-online.org/2013/08/3991/
31. C z e r n i a c h o w s k a, K., R. W i c h n i a r e k, K. Ż y w i c k i. Constraint Programming for Flexible Flow Shop Scheduling Problem with Repeated Jobs and Repeated Operations. – Advances in Science and Technology Research Journal, Vol. **17**, 2023, No 3, pp. 280-293. ISSN 2299-8624. DOI: 10.12913/22998624/166588.
32. L i, Z., Y. C h e n. Minimizing the Makespan and Carbon Emissions in the Green Flexible Job Shop Scheduling Problem with Learning Effects. – Scientific Reports, Vol. **13**, 2023, 6369. DOI: 10.1038/s41598-023-33615-z.
33. M e h d i z a d e h, E., F. S o l e i m a n i n i a. A Vibration-Damping Optimization Algorithm to Solve Flexible Job Shop Scheduling Problems with Reverse Flows. – International Journal of Research in Industrial Engineering, Vol. **12**, 2023, No 4, pp. 431-449. DOI: 10.22105/riej.2023.383451.1363.
34. G r a n, S. S., I. I s m a i l, T. A. A j o l, A. F. A. I b r a h i m. Mixed Integer Programming Model for Flexible Job-Shop Scheduling Problem (FJSP) to Minimize Makespan and Total Machining Time. – In: Proc. of International Conference on Computer, Communications, and Control Technology (I4CT'15), Kuching, Malaysia, 2015, pp. 413-417. DOI: 10.1109/I4CT.2015.7219609.
35. B i r g i n, E. G., P. F e o f i l o f f, C. G. F e r n a n d e s et al. A MILP Model for an Extended Version of the Flexible Job Shop Problem. – Optim. Lett., Vol. **8**, 2014, No 4, pp. 1417-1431.
36. H a n s m a n n, R. S., T. R i e g e r, U. T. Z i m m e r m a n n. Flexible Job Shop Scheduling with Blockages. – Math. Methods Oper. Res., Vol. **79**, 2014, No 2, pp. 135-161.
37. M e n g, L., C. Z h a n g, X. S h a o, Y. R e n. MILP Models for Energy-Aware Flexible Job Shop Scheduling Problem. – Journal of Cleaner Production, Vol. **210**, 2019, pp. 710-723. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2018.11.021.
38. M e n g, L., C. Z h a n g, B. Z h a n g, K. G a o, Y. R e n, H. S a n g. MILP Modeling and Optimization of Multi-Objective Flexible Job Shop Scheduling Problem with Controllable Processing Times. – Swarm and Evolutionary Computation, Vol. **82**, 2023, 101374. ISSN: 2210-6502. DOI: 10.1016/j.swevo.2023.101374.
39. G e d i k, R., C. R a i n w a t e r, H. N a c h t m a n n, E. A. P o h l. Analysis of a Parallel Machine Scheduling Problem with Sequence Dependent Setup Times and Job Availability Intervals. – European Journal of Operational Research, Vol. **251**, 2016, Issue 2, pp. 640-650. ISSN 0377-2217. DOI: 10.1016/j.ejor.2015.11.020.
40. H a m, M., E. C a k i c i. Flexible Job Shop Scheduling Problem with Parallel Batch Processing Machines: MIP and CP Approaches. – Computers & Industrial Engineering, Vol. **102**, 2016, pp. 160-165. ISSN 0360-8352. DOI: 10.1016/j.cie.2016.11.001.
41. F e k i h, A., H. H a d d a, I. K a c e m, A. B. H a d j-A l o u a n e. Mixed-Integer Programming and Constraint Programming Models for the Flexible Job Shop Scheduling Problem. – In: T. Masrour, I. El Hassani, N. Barka, Eds. Artificial Intelligence and Industrial Applications. A2IA 2023. – Lecture Notes in Networks and Systems. Vol. **771**. Cham, Springer, 2023. DOI: 10.1007/978-3-031-43524-9_8.
42. K u, W.-Y., J. C. B e c k. Mixed Integer Programming Models for Job Shop Scheduling: A Computational Analysis. – Computers & Operations Research, Vol. **73**, 2016, pp. 165-173. ISSN 0305-0548. DOI: 10.1016/j.cor.2016.04.006.
43. M ü l l e r, D., M. G. M ü l l e r, D. K r e s s, E. P e s c h. An Algorithm Selection Approach for the Flexible Job Shop Scheduling Problem: Choosing Constraint Programming Solvers through Machine Learning. – European Journal of Operational Research, Vol. **302**, 2022, Issue 3, pp. 874-891. ISSN 0377-2217. DOI: 10.1016/j.ejor.2022.01.034.
44. A s c h a u e r, A., F. R o e t z e r, A. S t e i n b o e c k, A. K u g i. Scheduling of a Flexible Job Shop with Multiple Constraints. – IFAC-PapersOnLine, Vol. **51**, 2018, Issue 11, pp. 1293-1298. ISSN 2405-8963. DOI: 10.1016/j.ifacol.2018.08.354.
https://www. sciencedirect.com/science/article/pii/S2405896318314782

45. S o t o, C., B. D o r r o n s o r o, H. F r a i r e, L. C r u z-R e y e s, C. G o m e z-S a n t i l l a n, N. R a n g e l. Solving the Multi-Objective Flexible Job Shop Scheduling Problem with a Novel Parallel Branch and Bound Algorithm. – Swarm and Evolutionary Computation, Vol. **53**, 2020, 100632. ISSN 2210-6502. DOI: 10.1016/j.swevo.2019.100632.
46. C h e n, X., J. Z h a n g. A Dynamic Programming Approach to Solving Flexible Job Shop Scheduling Problems. – Journal of Manufacturing Science and Engineering, Vol. **137**, 2015, No 3, 031007. DOI: 10.1115/1.4029334.
47. L i u, B., C. L e e. Dynamic Programming for Flexible Job Shop Scheduling with Machine Failure Considerations. – Computers & Operations Research, Vol. **92**, 2018, pp. 26-37. DOI: 10.1016/j.cor.2017.08.009.
48. G u p t a, M., P. J a i n. Dynamic Programming Approach for Multi-Objective Flexible Job Shop Scheduling. – Journal of Manufacturing Processes, Vol. **44**, 2019, pp. 198-211. DOI: 10.1016/j.jmapro.2019.06.030.
49. Y a n g, H., Q. P a n. An Enhanced Dynamic Programming Approach for Flexible Job Shop Scheduling Problem with Parallel Machines. – Computers & Industrial Engineering, Vol. **147**, 2020, 106635. DOI: 10.1016/j.cie.2020.106635.
50. M a l i k a, B., H. K a l l a. A Fault Tolerant Scheduling Heuristics for Distributed Real Time Embedded Systems. – Cybernetics and Information Technologies, Vol. **18**, 2018, No 3, pp. 48-61.
51. A z z o u z, A., M. E n i g r o u, L. B e n S a i d. A Hybrid Heuristic Algorithm for Flexible Job-Shop Scheduling Problem with Setup Times. – International Journal of Production Management and Engineering, Vol. **5**, 2017, No 1, pp. 53-61. DOI: 10.4995/ijpme.2017.6618.
52. Z i a e e, M. A Heuristic Algorithm for Solving Flexible Job Shop Scheduling Problem. – The International Journal of Advanced Manufacturing Technology, Vol. **71**, 2014, pp. 519-528. DOI: 10.1007/s00170-013-5510-z.
53. C a m p o, E. A., J. A. C a n o, R. G ó m e z-M o n t o y a, E. R o d r í g u e z-V e l á s q u e z, P. C o r t é s. Flexible Job Shop Scheduling Problem with Fuzzy Times and Due-Windows: Minimizing Weighted Tardiness and Earliness Using Genetic Algorithms. – Algorithms, Vol. **15**, 2022, No 10, 334. DOI: 10.3390/a15100334.
54. Z h a n g, Y., Y. L i, X. L i. Dynamic Scheduling for Flexible Job Shop Based on MachineRank. – Scientific Reports, Vol. **14**, 2024, 29741. DOI: 10.1038/s41598-024-79593-8.
55. L i u, C.-L., T.-H. H u a n g, S. D. Y o u. Scheduling for the Flexible Job-Shop Problem with a Dynamic Environment Using Deep Reinforcement Learning. – Information, Vol. **15**, 2024, No 2, 82. DOI: 10.3390/info15020082.
56. I n f a n t e s, G. Learning to Solve Job Shop Scheduling under Uncertainty. – arXiv:2404.01308v1, 2024. **https://arxiv.org/html/2404.01308v1?utm_source=chatgpt.com**
57. Y u a n, Y., H. X u, M. Y a n g. A Modified Iterated Greedy Algorithm for Flexible Job Shop Scheduling Problem. – Chinese Journal of Mechanical Engineering, 2019. DOI: 10.1186/s10033-019-0337-7.
58. R e z a e i p a n a h, A., F. S a r h a n g n i a, M. A b d o l l a h i. A Meta-Heuristic Approach Based on the Genetic and Greedy Algorithms to Solving Flexible Job-Shop Scheduling Problem. – Computer Science, 2021. DOI: 10.7494/csci.2021.22.4.4130.
59. A m j a d, M. K., S. I. B u t t, R. K o u s a r, R. A h m a d, M. H. A g h a, Z. F a p i n g, N. A n j u m, U. A s g h e r. Recent Research Trends in Genetic Algorithm Based Flexible Job Shop Scheduling Problems. Wiley, Online Library, 2018. DOI: 10.1155/2018/9270802.
60. J i a n g, M., H. Y u, J. C h e n. Improved Self-Learning Genetic Algorithm for Solving Flexible Job Shop Scheduling. – Mathematics, Vol. **11**, 2023, No 22, 4700. DOI: 10.3390/math11224700.
61. L u o, X., Q. Q i a n, Y. F. F u. Improved Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem. – Procedia Computer Science, Vol. **166**, 2020, pp. 480-485. DOI: 10.1016/j.procs.2020.02.061.
62. Y a z d a n i, E., P. F a t t a h i, M. J o l a i. A Priority-Based Genetic Algorithm for a Flexible Job Shop Scheduling Problem. – Journal of Industrial and Management Optimization, 2016. DOI: 10.3934/jimo.2016.12.1391.

63. Cruz-Chávez, M. A., M. G. Martínez-Rangel, M. H. Cruz-Rosales. Accelerated Simulated Annealing Algorithm Applied to the Flexible Job Shop Scheduling Problem. – International Transactions in Operational Research, Vol. **24**, 2017, No 5, pp. 1119-1137. DOI: 10.1111/itor.12195.
64. Zheng, Q.-Q., Y. Zhang, H.-W. Tian, L.-J. He. An Effective Hybrid Meta-Heuristic for Flexible Flow Shop Scheduling with Limited Buffers and Step-Deteriorating Jobs. – Engineering Applications of Artificial Intelligence, Vol. **106**, 2021, 104503. ISSN 0952-1976. DOI: 10.1016/j.engappai.2021.104503.
65. Lim, K. C. W., L.-P. Wong, J. F. Chin. Simulated-Annealing-Based Hyper-Heuristic for Flexible Job-Shop Scheduling. – Engineering Optimization, Vol. **55**, 2023, No 10, pp. 1635-1651. DOI: 10.1080/0305215X.2022.2106477.
66. Teekeng, W., A. Thammano, P. Unkaw, J. Kiatwuthiamorn. A New Algorithm for Flexible Job-Shop Scheduling Problem Based on Particle Swarm Optimization. – Artificial Life and Robotics, Vol. **21**, 2016, pp. 18-23. DOI: 10.1007/s10015-015-0259-0.
67. Nouiri, M., A. Bekrar, A. C. Ammari. An Effective and Distributed Particle Swarm Optimization Algorithm for Flexible Job-Shop Scheduling Problem. – Journal of Intelligent Manufacturing, 2018. DOI: 10.1007/s10845-015-1039-3.
68. Liu, Z., J. Wang, C. Zhang, H. Chu, G. Ding, L. Zhang. A Hybrid Genetic-Particle Swarm Algorithm Based on Multilevel Neighbourhood Structure for Flexible Job Shop Scheduling Problem. – Computers and Operations Research, Vol. **135**, 2021, Issue C. DOI: 10.1016/j.cor.2021.105431.
69. Wang, L., J. Cai, M. Li, Z. Liu. Flexible Job Shop Scheduling Problem Using an Improved Ant Colony Optimization. – Scientific Programming, 2017. DOI: 10.1155/2017/9016303.
70. Kalinowski, K., D. Krenczyk, I. Paprocka, W. M. Kempa, C. Grabowik. Ant Colony Optimisation for Scheduling of Flexible Job Shop with Multi-Resources Requirements. – MATEC Web of Conferences, 2017. DOI: 10.1051/matecconf/201711206018.
71. Torres-Tapia, W., J. R. Montoya-Torres, J. Ruiz-Meza. A Hybrid Algorithm Based on Ant Colony System for Flexible Job Shop. – In: J. C. Figueroa-García, C. Franco, Y. Díaz-Gutierrez, G. Hernández-Pérez, Eds. Applied Computer Sciences in Engineering. WEA 2022. Part of the Book Series: Communications in Computer and Information Science. Vol. **1685**. Cham, Springer, 2022. DOI: 10.1007/978-3-031-20611-5_17.
72. Yan, S., G. Zhang, J. Sun, W. Zhang. An Improved Ant Colony Optimization for Solving the Flexible Job Shop Scheduling Problem with Multiple Time Constraints. – Mathematical Biosciences and Engineering, Vol. **20**, 2023, No 4, pp. 7519-7547. DOI: 10.3934/mbe.2023325.
73. Da Silva, A. R. Solving the Job Shop Scheduling Problem with Ant Colony Optimization. – arXiv:2209.05284**,** [cs.MA], 2022. DOI: 10.48550/arXiv.2209.05284.
74. Wang, Y., Q. Zhu. A Hybrid Genetic Algorithm for Flexible Job Shop Scheduling Problem with Sequence-Dependent Setup Times and Job Lag Times, July 2021. – IEEE Access, Vol. **PP(99)**, 2021, pp. 1-1. DOI: 10.1109/ACCESS.2021.3096007.
75. Li, X., L. Gao. An Effective Hybrid Genetic Algorithm and Tabu Search for Flexible Job Shop Scheduling Problem. – International Journal of Production Economics, Vol. **174**, 2016, pp. 93-110. ISSN 0925-5273. DOI: 10.1016/j.ijpe.2016.01.016.
76. Marzouki, B., O. B. Driss, L. Ghédira. Decentralized Tabu Searches in Multi Agent System for Distributed and Flexible Job Shop Scheduling Problem. – In: Proc. of 14th International Conference on Computer Systems and Applications IEEE/ACS (AICCSA'17), Hammamet, Tunisia, 2017, pp. 1019-1026. DOI: 10.1109/AICCSA.2017.133.
77. Juvin, C., L. Houssin, P. Lopez. Hybrid Methods to Solve the Two-Stage Robust Flexible Job-Shop Scheduling Problem with Budgeted Uncertainty. – In: Proc. of 12th International Conference on Operations Research and Enterprise Systems (ICORES'23), 2023, pp. 135-142. ISBN: 978-989-758-627-9. ISSN: 2184-4372. DOI: 10.5220/0011619700003396.
78. Shen, L., S. Dauzère-Pérès, J. S. Neufeld. Solving the Flexible Job Shop Scheduling Problem with Sequence-Dependent Setup Times. – European Journal of Operational Research, Vol. **265**, 2018, Issue 2, pp. 503-516. ISSN 0377-2217. DOI: 10.1016/j.ejor.2017.08.021.

79. Z e n g, B., L. Z h a o. Solving Two-Stage Robust Optimization Problems by a Column-and-Constraint Generation Method. – Operations Research Letters, Vol. **45**, 2013, No 5, pp. 457-461. DOI: 10.1016/j.orl.2013.05.003.
80. M e n g, L., W. C h e n g, B. Z h a n g, W. Z o u, P. D u a n. A Novel Hybrid Algorithm of Genetic Algorithm, Variable Neighborhood Search and Constraint Programming for Distributed Flexible Job Shop Scheduling Problem. – International Journal of Industrial Engineering Computations, Vol. **15**, 2024, Issue 3, pp. 813-832. DOI: 10.5267/j.ijiec.2024.3.001.
81. L i, D., T. Z h e n g, J. L i, A. T e y m o u r i f a r. A Hybrid Framework Integrating Machine-Learning and Mathematical Programming Approaches for Sustainable Scheduling of Flexible Job-Shop Problems. – Chemical Engineering Transactions, Vol. **103**, 2023, pp. 385-390. DOI:10.3303/CET23103065.
82. T h i, L. M., T. T. M a i A n h, N. V a n H o p. An Improved Hybrid Metaheuristics and Rule-Based Approach for Flexible Job-Shop Scheduling Subject to Machine Breakdowns. – Engineering Optimization, Vol. **55**, 2022, No 9, pp. 1535-1555. DOI: 10.1080/0305215X.2022.2098283.
83. W a n g, Y., B. v a n S t e i n, M. T. M. E m m e r i c h, T. B ä c k. A Tailored NSGA-III Instantiation for Flexible Job Shop Scheduling. – arXiv:2004.06564, 2020. DOI: 10.48550/arXiv.2004.06564.
84. L u n a r d i, W. T., E. G. B i r g i n, D. P. R o n c o n i, H. V o o s. Metaheuristics for the Online Printing Shop Scheduling Problem. – arXiv:2006.12344, 2020. DOI: 10.48550/arXiv.2006.12344.
85. B o r i s s o v a, D., I. M u s t a k e r o v. A Parallel Algorithm for Optimal Job Shop Scheduling of Semi-Constrained Details Processing on Multiple Machines. – In: Proc. of 18th Int. Conf. on Circuits, Systems, Communications and Computers (CSCC'14), Advanced Information Science and Applications – Vol. **I**, 17-21 July 2014, Santorini Island, Greece, pp. 145-150. ISBN: 978-1-61804-236-1.
86. S e n t h i l k u m a r, M, P. I l a n g o. A Survey on Job Scheduling in Big Data. – Cybernetics and Information Technologies, Vol. **16**, 2016, No 3, pp. 35-51.
87. Z a i d i, M., A. A m i r a t, B. J a r b o u i, A. Y a h y a o u i. A Hybrid Meta-Heuristic to Solve Flexible Job Shop Scheduling Problem. – In: I. Alharbi, C. E. Ben Ncir, B. Alyoubi, H. Ben-Romdhane, Eds. Advances in Computational Logistics and Supply Chain Analytics. Unsupervised and Semi-Supervised Learning. Springer, Cham. 2024. DOI: 10.1007/978-3-031-50036-7_4.
88. K a w e e w i t b u n d i t, P., T. E g u c h i. Flexible Job Shop Scheduling Using Genetic Algorithm and Heuristic Rules. – Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol. **10**, 2016, Issue 1. DOI: 10.1299/jamdsm.2016jamdsm0010.
89. T u t u m l u, B., T. S a r a ç. A MIP Model and a Hybrid Genetic Algorithm for Flexible Job-Shop Scheduling Problem with Job-Splitting. – Computers and Operations Research, Vol. **155**, 2023, 106222. ISSN 0305-0548. DOI: 10.1016/j.cor.2023.106222.
90. Ç a l i ş, B., S. B u l k a n. A Research Survey: Review of AI Solution Strategies of Job Shop Scheduling Problem. – Journal of Intelligent Manufacturing, Vol. **26**, 2015, No 5, pp. 961-973.
91. E c h e v e r r i a, I., M. M u r u a, R. S a n t a n a. Solving the Flexible Job-Shop Scheduling Problem Through an Enhanced Deep Reinforcement Learning Approach. 2023. DOI: 10.48550/arXiv:2310.15706.
92. F e n g, Y., L. Z h a n g, Z. Y a n g, Y. G u o, D. Y a n g. Flexible Job Shop Scheduling Based on Deep Reinforcement Learning. – In: Proc. of 5th Asian Conference on Artificial Intelligence Technology (ACAIT'21), Haikou, China, 2021, pp. 660-666. DOI: 10.1109/ACAIT53529.2021.9731322.
93. W a n g, R., G. W a n g, J. S u n, F. D e n g, J. C h e n. Flexible Job Shop Scheduling via Dual Attention Network Based Reinforcement Learning. – arXiv:2305.05119 [cs.LG], 2023. DOI: 10.48550/arXiv.2305.05119.
94. E c h e v e r r i a, I., M. M u r u a, R. S a n t a n a. Leveraging Constraint Programming in a Deep Learning Approach for Dynamically Solving the Flexible Job-Shop Scheduling Problem. – arXiv:2403.09249 [cs.AI], 2024. DOI: 10.48550/arXiv.2403.09249.

95. T r e m b l e t, D., S. T h e v e n i n, A. D o l g u i. Makespan Estimation in a Flexible Job-Shop Scheduling Environment Using Machine Learning. – International Journal of Production Research, 2024. DOI: 10.1080/00207543.2023.2245918.
96. H u a n g, X., Z. G u a n, L. Y a n g. An Effective Hybrid Algorithm for Multi-Objective Flexible Job-Shop Scheduling Problem. – Advances in Mechanical Engineering, Vol. **10**, 2018, No 9. DOI: 10.1177/1687814018801442.
97. L i, J., H. L i, P. H e, L. X u, K. H e, S. L i u. Flexible Job Shop Scheduling Optimization for Green Manufacturing Based on Improved Multi-Objective Wolf Pack Algorithm. – Applied Sciences, Vol. **13**, 2023, No 14, 8535. DOI: 10.3390/app13148535.
98. T h e n a r a s u, M., K. R a m e s h k u m a r, M. d i M a s c o l o, S. P. A n b u u d a y a s a n k a r. Multi-Criteria Scheduling of Realistic Flexible Job Shop: A Novel Approach for Integrating Simulation Modelling and Multi-Criteria Decision Making. – International Journal of Production Research, 2023, pp. 1-23. DOI: 10.1080/00207543.2023.2238084, hal-04176568. **https://hal.science/hal-04176568/document**
99. T a o, N., J. H u a. A Cloud-Based Improved Method for Multi-Objective Flexible Job-Shop Scheduling Problem. – Journal of Intelligent & Fuzzy Systems, Vol. **35**, 2018, No 1, pp. 823-829. DOI: 10.3233/JIFS-171391.
100. F u, K., J. L i u, M. C h e n. Solving Flexible Job-Shop Scheduling Problems Based on Quantum Computing. – Entropy, Vol. **27**, 2025, No 2, 189. DOI: 10.3390/e27020189.
101. X u e, L., S. Z h a o, A. M a h m o u d i et al. Flexible Job-Shop Scheduling Problem with Parallel Batch Machines Based on an Enhanced Multi-Population Genetic Algorithm. – Complex Intelligent Systems, Vol. **10**, 2024, pp. 4083-4101. DOI: 10.1007/s40747-024-01374-7.
102. L u o, J., D. E l B a z. A Dual Heterogeneous Island Genetic Algorithm for Solving Large Size Flexible Flow Shop Scheduling Problems on Hybrid Multi-Core CPU and GPU Platforms. – arXiv:1903.10722 [cs.DC], 2019. DOI: 10.48550/arXiv.1903.10722.
103. L u o, J., S. F u j i m u r a, D. E l B a z. GPU Based Parallel Genetic Algorithm for Solving an Energy Efficient Dynamic Flexible Flow Shop Scheduling Problem. – arXiv:1903.10741 [cs.DC], 2019. DOI: 10.48550/arXiv.1903.10741.
104. K u b i a k, W., Y. F e n g, G. L i, S. P. S e t h i, C. S r i s k a n d a r a j a h. Efficient Algorithms for Flexible Job Shop Scheduling with Parallel Machines. – Naval Research Logistics, Vol. **67**, 2020, No 4, pp. 272-288. DOI: 10.1002/nav.21901.
105. F u l a d i, S. K., C.-S. K i m. Dynamic Events in the Flexible Job-Shop Scheduling Problem: Rescheduling with a Hybrid Metaheuristic Algorithm. – Algorithms, Vol. **17**, 2024, No 4, 142. DOI: 10.3390/a17040142.
106. C h e n, X., J. L i, Z. W a n g, Q. C h e n, K. G a o, Q. P a n. Optimizing Dynamic Flexible Job Shop Scheduling Using an Evolutionary Multi-Task Optimization Framework and Genetic Programming. – In: IEEE Transactions on Evolutionary Computation, 2023. DOI: 10.1109/TEVC.2025.3543770.
107. L i u, R., R. P i p l a n i, C. T o r o. Deep Reinforcement Learning for Dynamic Scheduling of a Flexible Job Shop. 2022. **https://ideas.repec.org/a/taf/tprsxx/v60y2022i13p4049-4069.html**
108. K i m, M.-S., et al. A Dynamic Programming-Based Heuristic Algorithm for a Flexible Job Shop Scheduling Problem of a Matrix System in Automotive Industry. – In: Proc. of 18th IEEE International Conference on Automation Science and Engineering (CASE’22), Mexico City, Mexico, 2022, pp. 777-782. DOI: 10.1109/CASE49997.2022.9926440.
109. Z h a n g, C., Y. Z h o u, K. P e n g, X. L i, K. L i a n, S. Z h a n g. Dynamic Flexible Job Shop Scheduling Method Based on Improved Gene Expression Programming. – Measurement and Control, Vol. **54**, 2021, No 7-8, pp. 1136-1146. DOI: 10.1177/0020294020946352.
110. O z t u r k, G., O. B a h a d i r, A. T e y m o u r i f a r. Extracting Priority Rules for Dynamic Multi-Objective Flexible Job Shop Scheduling Problems Using Gene Expression Programming. – International Journal of Production Research, Vol. **57**, 2019, No 10, pp. 3121-3137.

Received: 28.01.2025, Revised version: 12.02.2025, Accepted: 07.03.2025