A Review of Metaheuristic Algorithms for Job Shop Scheduling

Dharmik Chiragkumar Hajariwala[1], Srishti Sudhir Patil[2] and Sudhir Madhav Patil[3,*]

[1]SYMTech-Project Management, Department of Manufacturing Engineering and Industrial Management,
COEP Technological University (COEP Tech), Chhatrapati Shivajinagar, Pune: 411005, Maharashtra State, India
[2]SYBTech-CSE, Department of Computer Engineering and Technology,
Dr. Vishwanath Karad MIT World Peace University (MIT-WPU), Kothrud, Pune: 411038, Maharashtra State, India
[3]Department of Manufacturing Engineering and Industrial Management,
COEP Technological University (COEP Tech), Chhatrapati Shivajinagar, Pune: 411005, Maharashtra State, India

*Corresponding Email : smp.prod@coeptech.ac.in

Received May 9, 2024, Revised June 11, 2024, Accepted June 17, 2024, Published December 17, 2024

Abstract. *Job shop scheduling (JSS) is a critical problem in the field of operations research and manufacturing, where the goal is to optimize the scheduling of jobs on machines to enhance productivity and efficiency. Combinatorial optimization problems like JSS present significant challenges due to their diverse applications and practical importance. In order to meet this challenge, metaheuristic algorithms have become extremely effective tools. They provide effective solutions that strike a balance between computational cost and solution quality. Given the Nondeterministic Polynomial time (NP)-hard nature of the problem, exact methods are often impractical for large instances, making metaheuristic approaches highly valuable due to their ability to find near-optimal solutions within reasonable computational times. The primary purpose of this review manuscript is to comprehensively analyze and synthesize the current state of research on metaheuristic algorithms applied to JSS. This review categorizes and summarizes contemporary metaheuristic methods such as harmony search, and ant colony optimization, alongside traditional techniques like genetic algorithms, simulated annealing, tabu search, and particle swarm optimization. The fundamental concepts, key components, and typical applications of metaheuristic algorithms are explored. The paper evaluates robustness, scalability, and adaptability of different methods to different problem instances and constraints, and performance metrics, highlighting their strengths and weaknesses. Additionally, this paper reviews recent advancements in hybrid and multi-objective metaheuristic methods aimed at balancing scheduling constraints and improving solution quality and convergence speed. By offering a critical evaluation of the literature, this manuscript aims to identify trends, gaps, and future research directions in the application of metaheuristic algorithms to JSS. The discussion includes an exploration of emerging techniques and their potential impact on the field, as well as the practical implications for industrial applications. The conclusion of the review highlights that while significant advancements have been made, there remain numerous opportunities for innovation and improvement in developing more robust, efficient, and adaptive metaheuristic algorithms. Future research should focus on hybrid approaches, real-time scheduling, and integrating machine learning techniques to further enhance the performance and applicability of these algorithms in complex, real-world JSS problems. This comprehensive review not only serves as a valuable resource for researchers and practitioners but also sets the stage for future innovations in the optimization of complex scheduling problems.*

Keywords:

Job shop scheduling, metaheuristic algorithms, multi-objective, scheduling, scheduling problem

1. Introduction

Effective scheduling is essential to many industrial and manufacturing processes; it affects things like output volume, how resources are used, and total operating expenses. Because of its complexity and the numerous constraints involved, job shop scheduling (JSS) stands out among the many scheduling issues as a particularly difficult task. A set of jobs must be completed in a job shop setting in a certain order on a set of machines while abiding by a number of restrictions, include processing times, resource availability, and precedence relationships [1]. Traditionally, precise optimization techniques like mathematical programming have been used to solve job shop scheduling problem (JSSP). However, for large problem sizes encountered in real-world scenarios, exact methods are frequently computationally impractical due to the combinatorial nature of these problems [2]. Metaheuristic algorithms have become a potential and useful tool for solving intricate optimization problems, such as JSS, in recent years [3]. When it comes to optimization, metaheuristics provide a versatile and adaptive method that can quickly and effectively search through large solution spaces and identify excellent solutions [4].

The goal of this review is to give a thorough overview of metaheuristic algorithms used to solve scheduling issues in job shops. This paper examines the foundations of JSS, present several metaheuristic algorithms frequently employed in this setting, and investigate their uses, difficulties, and potential applications.

A. Brief Overview of Job Shop Scheduling Problem

The JSSP is a well-known optimization problem that arises in manufacturing settings where a range of jobs, each with a unique processing requirement and sequence of operations must be processed on a set of machines. Essentially, the objective is to efficiently assign resources and ascertain the best order in which to complete tasks in order to minimizing makespan (the total amount of time needed to finish all jobs) or other performance metrics while respecting different constraints. The origins of job shop production scheduling can be traced back to the early days of industrialization when manufacturing processes began to diversify, necessitating more sophisticated scheduling techniques to manage the increased complexity [5]. Over the years, JSS has evolved into a prominent area of research within the fields of operations research, industrial engineering, and computer science, owing to its significance in optimizing resource utilization, reducing production lead times, and improving overall efficiency [6]. Traditional methods of JSS, such as manual scheduling or simple rule-based approaches, often proved inadequate in handling the complexities inherent in modern manufacturing environments. This led to the development of algorithmic approaches aimed at automating and optimizing the scheduling process, thereby enabling manufacturers to achieve higher levels of productivity and competitiveness [7]. JSS is more complicated than simpler scheduling problems where each job travels a predetermined path through the production system. In this case, each job might need to be processed on several machines in a particular order, and factors like processing times, machine availability, and precedence relationships must be taken into account [8]. The combinatorial explosion of possible schedules as the number of jobs and machines increases is the fundamental complexity of the JSSP [9]. Consequently, for real-world scenarios with large problem sizes, finding an optimal solution through exhaustive search methods becomes unmanageable. For manufacturing operations to increase productivity, cut lead times, and maximize resource utilization, the JSSP must be solved effectively [10]. Therefore, in order to effectively address this difficult problem, researchers and practitioners have turned to optimization techniques like metaheuristic algorithms.

This paper examines how metaheuristic algorithms present a viable method for dealing with the difficulties involved in JSS and producing excellent results in a reasonable amount of computational time. Moreover, this paper explores a range of metaheuristic algorithms, how they are used in JSS, and their comparative performance analysis.

B. Importance of Efficient Scheduling in Industrial and Manufacturing Processes

Effective scheduling is critical to industrial and manufacturing processes because it affects many aspects of the business, such as overall competitiveness, productivity, and resource utilization. Manufacturers face constant pressure to streamline their production procedures and meet customer deadlines for high-quality products in today's hectic and fiercely competitive business world. By ensuring that resources are used effectively and production targets are met efficiently, efficient scheduling is essential to achieving these goals [11]. ***Figure 1*** shows the various key aspects of efficient scheduling.

Key Aspects of Efficient Scheduling

- Resource utilization: Using machinery, labor, and materials to their fullest potential is made possible by effective scheduling [12]. Throughput and productivity can be increased in manufacturing by avoiding bottlenecks and minimizing idle time [13].
- Lead time minimization: It is critical to reduce lead times in order to better respond to market shifts and satisfy customer demands. By cutting down on waiting times between operations, efficient scheduling speeds up production and makes it possible to fulfill orders more quickly [14].
- Production cost optimization: By lowering overtime costs, cutting setup times, and maximizing inventory levels, efficient scheduling techniques assist in lowering production costs. Manufacturing organizations can reduce costs and increase profitability by optimizing their production processes [15].
- Increased flexibility: Manufacturers need to be able to react fast to shifts in market conditions, resource availability, and demand in the fast-paced business world of today. Greater flexibility is made possible by efficient scheduling, which makes it possible to quickly modify production schedules in response to shifting requirements [16].
- Competitive advantage and customer satisfaction: Efficient scheduling gives manufacturers a big competitive edge because it allows them to deliver products faster, cheaper, and of better quality than their rivals [17]. Manufacturers can become preferred suppliers in the market, increase customer satisfaction, and forge closer bonds with customers by streamlining their production processes and strengthening their scheduling skills. In the end, effective scheduling promotes operational excellence

and adds value for customers, which leads to long-term business success and sustainable growth [18].

Thus, efficient scheduling is essential to optimizing industrial and manufacturing processes, reducing costs, increasing output, and maintaining competitiveness in today's fast-paced business environment. In the following sections of this review, the paper explores the ways in which advanced scheduling techniques, more especially metaheuristic algorithms, facilitate efficient scheduling and continuous improvement in manufacturing and industrial processes.

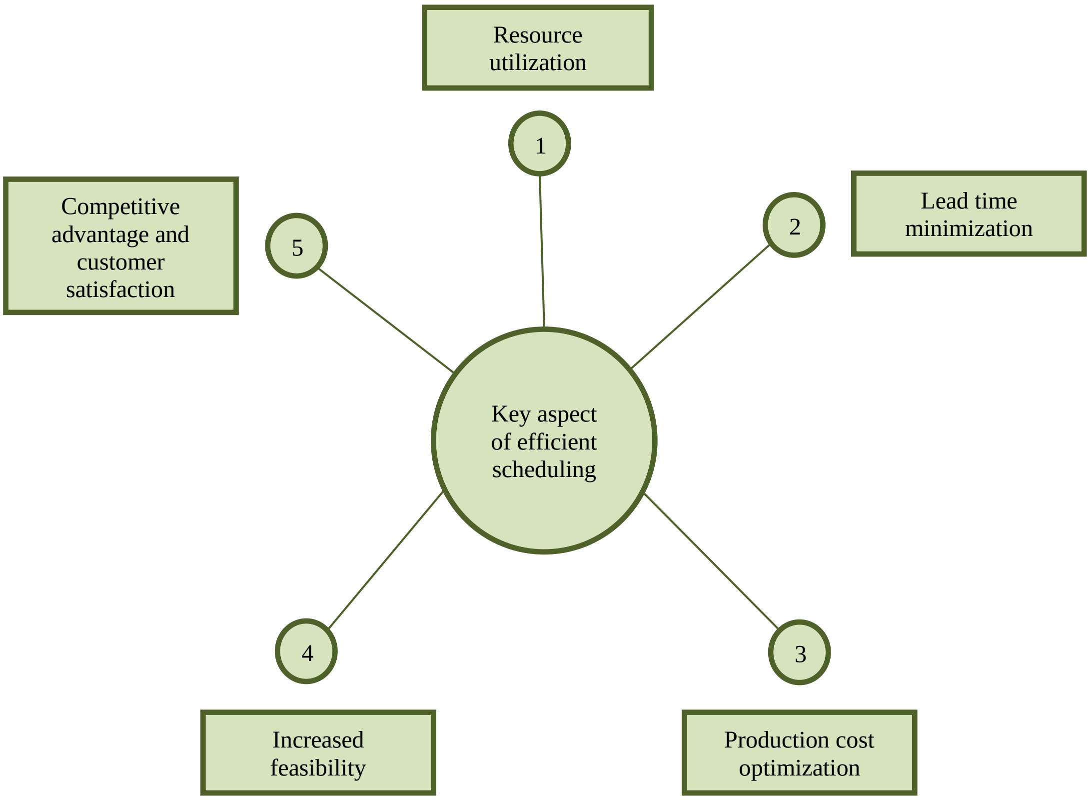

Fig. 1 Key aspects of efficient scheduling

C. Introduction to Metaheuristic Algorithms as a Solution Approach

In the quest of solving complicated optimization issues, large-scale examples with extensive solution spaces and multiple constraints frequently present challenges for conventional exact methods. In particular, metaheuristic algorithms provide a strong substitute for optimization when dealing with issues involving non-linear relationships and high computational complexity [19].

Definition of Metaheuristic Algorithm: Metaheuristic algorithms are iterative approaches to solving problems that search for better solutions by exploring and navigating solution spaces using ideas drawn from mathematics, social behavior, or natural phenomena. Metaheuristics are ideally suited to tackle real-world optimization problems because they put an emphasis on obtaining good solutions in a reasonable amount of time, as opposed to exact methods that ensure optimality but may have scalability issues [20].

Key Characteristics of Metaheuristic Algorithms:

- Iterative Improvement: Through a series of continuous steps, metaheuristic algorithms improve candidate solutions iteratively, gradually bringing the solution closer to optimality [20].
- Exploration and Exploitation: Metaheuristics strike a balance between exploitation (exploiting promising areas to refine the search towards optimal solutions) and exploration (diversifying the search to explore different regions of the solution space) [21].
- Stochastic Components: Many metaheuristic algorithms include unpredictable elements such as randomness, probabilistic making decisions, or simulated annealing (SA) in order to introduce randomness into the search process and escape local optimal conditions [4, 19].
- Adaptability: By changing parameters, operators, or search strategies, metaheuristic algorithms can be made more specialized for particular problem domains. This adaptability is a common feature [4, 20].

Categories of Metaheuristic Algorithms: Metaheuristic algorithms come in a multitude of forms, each with unique search mechanisms, optimization strategies, and underlying principles. ***Figure 2*** presents timeline of various meta-heuristic algorithm for JSS. Typical metaheuristic categories include the following [20]:

- Genetic Algorithms (GA)
- Simulated Annealing (SA)
- Tabu Search (TS)
- Particle Swarm Optimization (PSO)
- Harmony Search (HS)
- Ant Colony Optimization (ACO)
- Artificial Bee Colony (ABC)

Fig. 2 Timeline of Meta-heuristic algorithm for job shop scheduling

Applications of Metaheuristic Algorithms: Applications for metaheuristic algorithms are widely used in many different fields, such as engineering, finance, healthcare, and telecommunications, etc. ***Figure 3*** portrays the various applications of metaheuristic algorithms. When it comes to solving real-world problems where precise methods might not be feasible or practical, they are an invaluable tool due to their efficiency, versatility, and capacity to handle complex optimization problems [4, 44, 45]. Metaheuristic algorithms present a viable method for maximizing production schedules, minimizing makespan, and enhancing resource efficiency in the context of JSS [46-48]. The upcoming sections of this review examine the use of metaheuristic algorithms as a solution approach for JSSP, emphasizing their efficacy, applications, and potential to improve industrial and manufacturing process efficiency.

This paper examines the application of metaheuristic algorithms as a solution approach for JSS issues in the ensuing sections of this review. By delving into the underlying principles, mechanisms, and applications of metaheuristic algorithms in this context, this paper aim to provide insights into their effectiveness and potential for optimizing scheduling processes in industrial and manufacturing environments.

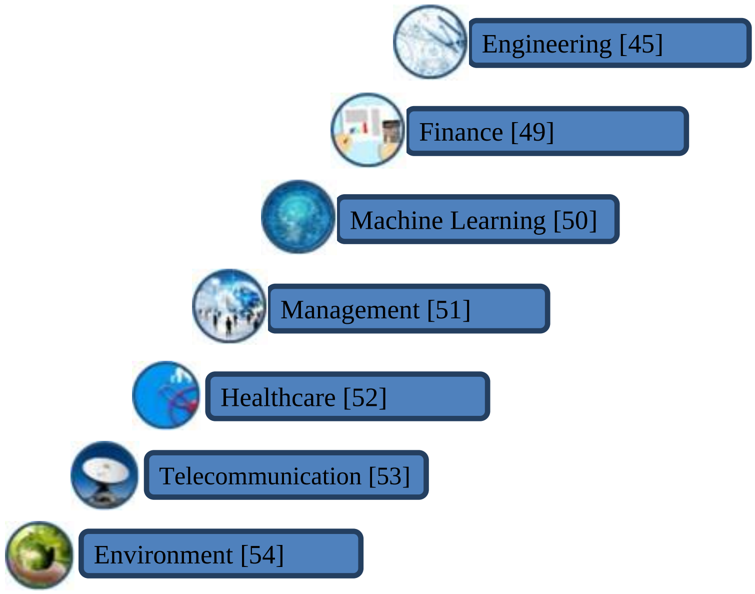

Fig. 3 Application of Meta-heuristic Algorithm

D. Objectives of using Metaheuristic Algorithm

- Minimize Maximum Completion Time (MMCT): Completion time refers to the arrangement of tasks in a way that maximizes the early completion of the longest task. The goal of this strategy is to shorten the total amount of time needed to finish all tasks. To guarantee effectiveness and cut down on delays, it is frequently used in scheduling. Concentrating on the most extended task aids in workload distribution and boosts output [55].

- Minimizing Makespan (MMK): The total amount of time needed to finish every task in a production schedule is referred to as makespan. This objective's algorithms focus on minimizing the makespan by streamlining the machine's sequence of operations to shorten the total production time. These algorithms give top priority to finishing every task as soon as possible, which reduces production time and boosts throughput overall [56].

- Robustness (RBS): This is a system's capacity to withstand unforeseen alterations or disruptions without malfunctioning. It guarantees dependability and stability in a range of circumstances. In engineering and design, a robust system is able to continue operating in the face of uncertainty. For the creation of reliable and long-lasting processes and products, this attribute is crucial [57].

- Convergence Speed (CVS): This refers to how quickly a process or algorithm reaches its outcome or solution. In optimization and iterative methods, faster convergence means fewer steps to achieve the desired result. This is crucial for efficiency, saving time and computational resources [46].

- Accuracy (ACC): This refers to the degree to which a result or measurement aligns with the true or accepted value. High accuracy translate into less error and more precision in the result. For results that are valid and dependable in domains like science, engineering, and data analysis, accuracy is essential [58, 59].

- Minimization of Tardiness (MTD): Tardiness refers to the delay in completing jobs beyond their due dates or deadlines. This objective's algorithms prioritize minimizing tardiness by making sure that tasks are finished on time or ahead of schedule. In order to maximize customer satisfaction, reduce late delivery penalties, and maintain a competitive advantage in the market, these algorithms optimize production schedules to meet predetermined deadlines [60, 61].

- Improve Scheduling Efficiency (ISE): Efficient scheduling ensures that tasks are completed in a timely and organized manner, reducing downtime and bottlenecks. This objective is essential in industries like manufacturing, project management, and logistics. Increasing scheduling efficiency results in higher productivity and lower operating costs. It results in improved performance overall and more efficient workflows [62, 63].

- Sequence Dependent Adjust Time (SDAT): This refers to the variation in the time required to switch from one task to another based on the order in which tasks are performed. This adjustment time depends on the specific sequence of operations. In scheduling and manufacturing, accounting for sequence-dependent adjust time is crucial for accurate planning and minimizing delays. It ensures that transitions between tasks are efficient, reducing overall completion time and improving productivity [64, 65].

- Machine Release Date (MRD): This refers to the specific time when a machine becomes available for use in a production process. This date is crucial for planning and scheduling tasks, ensuring that operations begin as soon as the machine is ready. Knowing the release date helps in coordinating activities and avoiding delays. It ensures that resources are utilized efficiently and production timelines are met [66, 67].
- Production Efficiency (PEF): This refers to the ability to maximize output while minimizing input, such as time, materials, and labour. A high level of production efficiency involves quick and minimal waste product creation. Reaching this goal is essential to cutting expenses and raising profits. Efficient production processes guarantee improved resource utilization and increased productivity [68].
- Work Sequence (WSQ): This describes the precise sequence in which actions or tasks are carried out. Sequencing correctly is essential for productivity and a seamless workflow. It guarantees that every task is finished on schedule, reducing delays and avoiding conflicts. An ideal sequence of operations improves output and efficient use of resources in both manufacturing and project management [69].
- Minimization of Production Time (MPT): This aims to reduce the overall duration required to manufacture goods or complete tasks. This objective focuses on optimizing processes, eliminating bottlenecks, and streamlining operations. By minimizing production time, businesses can increase output, meet deadlines, and respond swiftly to customer demands [70, 71].
- Cost Saving (CSV): This involves identifying and implementing measures to reduce expenses without compromising quality or efficiency. The goal of this objective is to reduce unnecessary expenditures in order to strengthen an organization's financial position. Achieving cost savings involves a variety of tactics, including improving resource usage, negotiating better prices with suppliers, and putting cost-effective technologies into use [72, 73].
- Minimize Set-up Time (MST): Set-up time is a crucial element in manufacturing and production processes, representing the duration required preparing equipment, machinery, or systems for the execution of a specific task or production run. It encompasses activities such as equipment adjustment, tooling, calibration, and material preparation before actual production can commence. Set-up time plays a significant role in overall production efficiency, as it directly affects downtime, throughput, and resource utilization [74, 75].
- Minimize Flow Time (MFT): Flow time is the amount of time needed for a job to proceed from the start of production to the end. In order to minimize the flow time for each job, algorithms that aim to minimize flow time focus on optimizing the job sequences and resource allocations. These algorithms seek to minimize flow time in order to increase customer responsiveness, decrease work-in-progress inventory, and increase production efficiency [76, 77].
- Minimize Relative Error (MRE): This objective aims to reduce discrepancies between planned and actual job completion times, enhancing scheduling accuracy [78].
- Improve Solution Quality (ISQ): This objective focuses on enhancing overall scheduling outcomes by optimizing job sequencing and resource allocation for better performance. This makes the schedule better to achieve higher efficiency and performance [78, 79].
- Minimize Earliness Penalties (MEP): Goal is to decrease costs associated with completing jobs earlier than required, thus reducing financial penalties [80, 81].
- Time Lag Requirement (TLR): This objective focuses on meeting specific time constraints between consecutive job operations to ensure smooth workflow and timely completion. It ensures the correct time intervals between tasks to meet production needs [82, 83].
- Minimize Energy Consumption (MEC): This aims to optimize scheduling to reduce energy usage, contributing to environmental sustainability and cost savings. This decreases the energy used in production to save money and resources [84, 85].
- Minimize Transportation Time (MTT): This objective focuses on minimizing the time required for transporting materials or products between different stages or workstations, improving efficiency and reducing delays. It cuts down the time needed to move materials between workstations [86, 87].

2. Overview of Job Shop Scheduling

JSS is an essential issue in operations management and manufacturing, which involves a requirement to allocate resources efficiently to complete a set of jobs on a set of machines, subject to various constraints [6]. This section will explore the essential ideas and concepts of JSS, including its definition, key characteristics, and categorization. JSS includes the coordination and sequencing of operations or tasks across different machinery to meet production objectives while satisfying various constraints. JSS differs from simpler scheduling problems in that it involves figuring out the best order of operations for every job on every machine, taking into account variables like processing times, machine availability, and precedence relationships [2, 82].

A. Definition of Job Shop Scheduling

This section delves into the definition and key characteristics of the JSS. At its core, the JSSP revolves around the efficient allocation of resources to complete a set of jobs on a set of machines within a specified period.

JSS is a typical optimization scenario came across in manufacturing and operations management, where a number of jobs must be processed through a series of operations on a number of machines. Every job consists of a number of tasks or operations that, under a number of restrictions, must be completed on a particular machine in the correct sequence [5, 88, 89].

Key Characteristics of JSS [90]:

- Machine Dependency: Each task within a job involves a series of operations that must be carried out on specific machines. On the other hand, machines might differ in their capacities, which could limit their availability and use.
- Precedence Constraints: Before they can begin, some operations might need to wait for others to finish. In order to ensure that the schedule is correct, these precedence constraints must be met.
- Complexity: The JSSP involves a large number of potential schedules, making it fundamentally combinatorial in nature. The complexity of the problems grows exponentially with the no. of jobs and machines, making it difficult to find optimal solutions using conventional optimization techniques.
- Objective Function: The main aim of JSS revolves around minimizing the makespan, which refers to the total duration required to complete all tasks. Other goals could be maximizing machine utilization, reducing tardiness, or distributing the workload evenly among the machines.

Importance of JSS: In manufacturing environments, maximizing resource utilization, cutting production lead times, and enhancing overall operational efficiency all depend on effective JSS. The implementation of efficient scheduling strategies can result in notable cost savings and productivity gains as they minimize idle time, minimize setup costs, and maximize machine utilization [1, 5, 6].

B. Description of the Job Shop Constraints

In this subsection, this paper discusses the constraints usually encountered in JSS, including machine availability, processing times, precedence relationships between tasks, and resource constraints. It is essential to understand the constraints, when creating scheduling problems and optimization algorithms.

Constraints in JSS:

- Machine Availability: Due to factors like resource limitations, setup times, and maintenance, each machine has a limited amount of availability. Jobs cannot be processed on machines that are already busy with other tasks [91].
- Processing Times: Every operation has a processing time that corresponds to how long it takes to finish the task on a specific machine. Processing times can differ based on a number of variables, including material properties, job complexity, and machine capabilities [92].
- Precedence Relationships: There are precedence relationships among certain operations, which determine that they must be carried out in a particular order. For instance, the outcome of machining operations on the same or separate machines may be necessary for the assembly process to proceed [93].
- Resource constraints: JSS may be impacted by limitations on labor, materials, and tools. Limited resources may have an influence on scheduling choices and reduce the effectiveness of production as an entire process [94].
- Job priorities: Different jobs may have varying priorities depending on things like deadlines, client demands, and output targets. Setting the right priorities for your work is crucial to meeting deadlines and maximizing output [95].
- Optimization objectives: The goal of JSS is to maximize machine utilization, minimize idle time, minimize makespan (total completion time), and reduce job tardiness (lateness). Alternatives may need to be made during the scheduling process because these goals may conflict with one another [96].

Understanding the job shop constraints is essential for formulating scheduling problems, designing effective algorithms, and making informed scheduling decisions. By considering these factors, researchers and practitioners can develop strategies to improve production efficiency, meet customer demands, and optimize resource utilization in job shop environments.

C. Classification of Job Shop Scheduling Problems

By categorizing JSSP, scholars and professionals are able to understand the details and complexity of various problem scenarios. It makes it easier to choose the best approaches for solving problems and to create specialized scheduling plans that are suited to the demands and features of particular problems [5].

The application of various metaheuristic algorithms to various JSSP, which aim to maximize production schedules, reduce makespan, and enhance resource utilization in industrial and manufacturing settings are discussed in the following sections of this review. The number of machines, the nature of the jobs and operations, and the scheduling environment are some of the factors that can be used to categorize JSS issues. Selecting suitable solution approaches and creating efficient scheduling strategies require an understanding of these classifications.

Classification of JSSPs:

Number of Machines:

- Single-Machine Job Shop: There is just one machine available for processing tasks in a single-machine job shop. Every job requires scheduling of several operations on this one machine. In comparison to other configurations, this scenario is very simple [81].
- Parallel Machine Job Shop: A parallel machine job shop has several identical machines that can process orders at the same time. More scheduling flexibility is possible with this configuration, which may also result in better throughput and resource use [97], [98].
- Hybrid Job Shop: A hybrid job shop incorporates aspects of parallel and single machine configurations. It could be made up of a combination of parallel and single machines, each with different processing capabilities and constraints [98].

Characteristics of Jobs and Operations:

- Deterministic Job Shop: The processing times for operations in a deterministic job shop are fixed and unchanging. Because job completion times and resource requirements can be accurately predicted, this helps to partially simplify the scheduling problem [99].
- Stochastic Job Shop: Processing times for operations are uncertain in a stochastic job shop. Scheduling decisions become more difficult when considering processing times that indicate random fluctuations or follow probabilistic distributions [99].
- Job Shop with Setup Times: Setup times are a factor in some scheduling problems in job shops because they are necessary when switching between various tasks or operations. Setup times add extra costs and delays to production, which can have a negative impact on efficiency [100].

Scheduling Environment:

- Static Job Shop: In a static job shop, scheduling choices are predetermined based on the known constraints and characteristics of the job. Scheduling algorithms are able to optimize schedules in person without requiring real-time adjustments, and the scheduling environment is still comparatively stable [101].
- Dynamic Job Shop: Scheduling decisions in a dynamic job shop are made either dynamically or in real-time in response to shifts in job priorities, the availability of resources, or outside variables. Adaptive algorithms that can quickly adjust to changing conditions and minimize production schedule disruptions are necessary for dynamic JSS [102].

Researchers and practitioners can effectively optimize scheduling processes and customize solution approaches to specific problem instances by having a thorough understanding of the classifications of JSSPs. The application of various metaheuristic algorithms to various JSSP, which aim to maximize production schedules, reduce makespan, and enhance resource utilization in industrial and manufacturing settings are discussed in the following sections of this review.

3. Overview of Metaheuristic Algorithms

In this section, the paper provides an overview of metaheuristic algorithms, exploring their principles, characteristics, and applications in solving optimization problems. Metaheuristic algorithms provide flexible and adaptive methods for optimization. They can quickly and effectively search through large solution spaces to find outstanding solutions. Developing an understanding of the foundations of metaheuristic algorithms is necessary to fully utilize their potential in solving complicated optimization problems, such as JSSPs.

A. Commonly Used Metaheuristic Algorithms

This subsection provides an overview of common metaheuristic algorithms used in JSS. Metaheuristic algorithms are powerful optimization techniques that offer flexible and robust approaches for finding near-optimal solutions to complex scheduling problems. By exploring these common metaheuristic algorithms, this paper aims to elucidate their principles, operational characteristics, and advantages in addressing various scheduling challenges in different environments.

- Genetic Algorithm (GA): Utilizes evolutionary processes to iteratively evolve solutions (schedules). Representing potential schedules as chromosomes, genetic operators like selection, crossover, and mutation are applied to breed better solutions across generations [22, 103].
- Simulated Annealing (SA): Mimics the physical annealing process, starting with an initial solution and exploring neighboring solutions probabilistically. By gradually decreasing the temperature parameter, it allows for escaping local optima and converging towards near-optimal solutions [23, 104].
- Tabu Search (TS): Employs a local search mechanism with a tabu list to avoid revisiting previously explored solutions. Aspiration criteria allow overcoming some restrictions, enabling efficient traversal of the search space and finding superior schedules [24, 105, 106].
- Parallel Genetic Algorithm (PGA): Utilizes parallel computing to speed up GA processes by running multiple instances concurrently, exploring different solution areas simultaneously [25, 103].
- Artificial Immune Algorithm (AIA): Mimics immune system processes to optimize problems, using antibodies to represent solutions and leveraging immune memory and clonal selection for adaptation [26].
- Ant Colony Optimization (ACO): Based on the foraging behavior of ants, where pheromone trails guide the search for solutions. Ants probabilistically select jobs and machines, with pheromone updates leading to the construction of high-quality schedules over time [27, 107].
- Music-Based Harmony Search (MBHS): Inspired by creating harmonious music. Solutions are represented as melodies, updated using principles of harmony and music theory to explore and find near-optimal solutions [28, 108].
- Improved Music Based Harmony Search (IMBHS): Enhances traditional Harmony Search with advanced harmony memory and improvisation techniques, improving convergence speed and solution quality for various optimization problems [28].
- Real Coded Genetic Algorithm (RGA): Uses real-number vectors instead of binary strings to represent solutions, allowing effective optimization of problems with continuous variables [29, 103].
- Bacterial Foraging Optimization (BFO): Simulates bacterial foraging behaviors. Solutions (bacterial colonies) are iteratively improved through chemotaxis, reproduction, and elimination-dispersal, effectively finding optimal solutions for various problems [26, 109].
- Particle Swarm Optimization (PSO): Inspired by social behaviors of animals, particles represent potential solutions. They move through the solution space based on their own experiences and those of the swarm, iteratively improving their positions to find near-optimal schedules [30, 44, 110].
- Artificial Bee Colony (ABC): Models the foraging behavior of honeybees. Employed, onlooker, and scout bees explore the solution space, sharing information and introducing diversity to iteratively improve solutions and converge towards optimal schedules [31, 111].
- Improved Artificial Bee Colony (IABC) Algorithm: The IABC algorithm improves the ABC algorithm. It ensures thorough search space coverage by using a four-layer chromosome encoding structure to represent solutions in an adaptable and natural way. Performance is further optimized by adding a random selection method for onlooker bees and a neighborhood search mechanism for employed bees [32, 112].
- Parallel Artificial Bee Colony (PABC) Algorithm: This algorithm enhances the traditional ABC by mimicking the foraging behavior of bees in a parallelized manner. The colony is divided into several sub colonies, and in order to increase efficiency, parallel operations are carried out within each sub colony with dynamic migration [33, 113, 114].
- Invasive Weed Optimization (IWO): Inspired by the invasive behavior of weeds. Candidate solutions (weed areas) evolve by growth, reproduction, and competition, efficiently exploring the solution space [26, 115].
- Differential Evolution (DE) Algorithm: It improves solutions by combining and mutating existing ones. This helps to reduce the total time needed to complete all jobs. It also enhances the use of resources efficiently [34, 116].
- Improved Differential Evolution (IDE) Algorithm: This algorithm improves traditional DE with four mutation strategies and an exponential crossover method, developing new

candidate solutions repeatedly. By choosing the best solution from each iteration based on fitness, it seeks to maximize the objectives in JSSP [35, 117].

- Cuckoo Search (CS) Algorithm: This algorithm optimizes JSS by mimicking cuckoo breeding strategies. It replaces poor solutions with better ones discovered through random search and Lévy flights. This method minimizes the total job completion time and improves scheduling efficiency [36, 118].

- Discrete Cuckoo Search (DCS) Algorithm: This algorithm uses random generation to create an initial population. It then transforms continuous solutions into discrete permutations and evaluates the fitness of each population. The algorithm accepts new solutions based on an exponentially decreasing probability and improves the population based on random moves (swap, insert, and inverse) [37, 119].

- Bat Algorithm (BA): The Bat Algorithm is used for JSS by simulating bat echolocation to explore solutions. It adjusts job sequences based on loudness and pulse rates, finding optimal schedules. This reduces completion time and improves scheduling efficiency [38, 120].

- Parallel Bat Algorithm (PBA): This algorithm enhances the traditional BA by uses a random-key encoding scheme for mapping continuous positions to discrete job sequences. It uses parallel processing, executing several procedures at once, and combining the outcomes on a regular basis to improve the quality of the solution. Neighborhood operators that enhance local search efficiency and preserve population diversity include swapping, insertion, and inversion [39, 121].

- Coral Reef Optimization (CRO): Mimics the cooperative behaviors of coral reef organisms. Candidate solutions interact through feeding, communication, and reproduction, effectively solving a range of optimization problems [40, 122].

- Parallel Coral Reef Optimization (PCRO): This algorithm establishes multiple sub-populations (coral reefs) and employs local search and crossover operators in parallel to enhance solution diversity and quality. Periodically, it improves convergence towards the ideal job scheduling by merging and exchanging solutions between subpopulations [41, 122].

- Grey Wolf Optimization (GWO): Inspired by the hunting behavior of grey wolves, where positions of solutions are updated based on alpha, beta, and delta wolves. The algorithm balances exploration and exploitation to find optimal solutions [42, 123].

- Discrete Grey Wolf Optimization (DGWO): A variation of GWO for discrete problems. Wolves reposition themselves similarly, evaluating solutions and converging towards optimal results [43, 124].

- Modified Genetic Algorithm (MoGA): A customized version of the classic genetic algorithm designed to address specific optimization problems, evolving solutions iteratively through processes like crossover, mutation, and selection [125].

- Improved Genetic Algorithm (IGA): Enhances the traditional genetic algorithm with sophisticated strategies like elitism and adaptive parameter tuning to improve exploration and exploitation, leading to faster convergence towards optimal or nearly optimal solutions [103, 126].

- Modified Ant Colony Optimization Algorithm (MACO): Customizes the standard ACO algorithm with improved pheromone updating rules, sophisticated heuristic data, or specific exploration tactics to quickly approach optimal or nearly optimal solutions [127].

- Enhanced Particle Swarm Optimization (EPSO): Improves the traditional PSO algorithm with mechanisms like adaptive inertia weights and local search strategies to explore solution spaces and approach optimal or nearly optimal solutions more effectively [128].

Table 1 presents the summary of various metaheuristic algorithms based on main concept, advantages, disadvantages and specific performance characteristics of specified research work from the literature.

Table 1 Summary of Metaheuristic Algorithms

Study/Reference	Metaheuristic Algorithm	Main Concept	Advantages	Disadvantages	Note on Performance
[22], [103]	Genetic Algorithm	Evolutionary algorithm using selection, crossover, and mutation	Good global search capabilities, adaptable	Can be slow, may converge prematurely	Often used for its robustness and ability to escape local optima
[23], [104]	Simulated Annealing	Probabilistic technique inspired by annealing in metallurgy	Simple, avoids local minima effectively	Slow convergence, parameter tuning is critical	Effective for problems with a complex landscape
[24], [105], [106]	Tabu Search	Uses memory structures to avoid cycles and enhance search	Avoids local minima, flexible memory structures	Computationally intensive, parameter tuning required	Performs well with complex constraints
[25], [103]	Parallel Genetic Algorithm	Parallel execution of Genetic Algorithms	Faster execution, maintains diversity	Complexity in implementation, requires parallel computing resources	Suitable for large-scale problems
[26]	Artificial Immune Algorithm	Inspired by the human immune system's adaptive learning	Good global search, adaptable to changes	Can be computationally expensive, complex	Effective in dynamic environments
[27], [107]	Ant colony optimization	Mimics behavior of ants finding paths to food sources	Good for combinatorial problems, positive feedback loop	Slow convergence, requires many iterations	Effective for routing and network problems
[28], [108]	Music Based Harmony Search	Inspired by musical improvisation	Simple, few parameters to tune	Can get stuck in local optima, performance depends on harmony memory	Easy to implement, often used for optimization problems with continuous variables
[28]	Improved Music Based Harmony Search	Enhanced version of Harmony Search with better exploration and exploitation	Improved performance over basic Harmony Search	More complex, requires fine-tuning	Better convergence and solution quality compared to basic version
[29], [103]	Real Coded Genetic Algorithm	Genetic Algorithm using real numbers instead of binary	Handles continuous optimization well	Similar issues as GA (slow convergence, premature convergence)	Suitable for continuous optimization problems
[26], [109]	Bacterial Foraging Optimization	Inspired by the foraging behavior of bacteria	Good global search ability, adaptable	Computationally intensive, slow convergence	Effective for dynamic and noisy environments
[30], [44], [110]	Particle Swarm Optimization	Models social behavior of birds flocking or fish schooling	Simple, few parameters, fast convergence	May get trapped in local optima, sensitive to parameter settings	Popular for its simplicity and efficiency
[31], [111]	Artificial Bee Colony	Mimics the foraging behavior of honey bees	Simple, flexible, good global search ability	Can converge prematurely, requires balancing exploration and exploitation	Effective for a wide range of optimization problems
[32], [112]	Improved Artificial Bee Colony Algorithm	Enhanced version with mechanisms to improve convergence and solution quality	Better performance compared to basic ABC	Increased complexity	Improved results in terms of convergence speed and solution quality
[33], [113], [114]	Parallel Artificial Bee Colony Algorithm	Parallel implementation of ABC for faster execution	Faster, maintains diversity, scalable	Complexity in implementation, parallel computing resources required	Suitable for large-scale and complex problems
[26], [115]	Invasive Weed Optimization	Inspired by the colonizing behavior of weeds	Good for multi-modal optimization	Can be computationally expensive, parameter sensitivity	Effective for dynamic and multi-modal environments
[34], [116]	Differential Evolution Algorithm	Uses differential mutation and crossover for optimization	Robust, simple, good for continuous optimization	May converge slowly, sensitive to parameter settings	Popular for continuous and real-valued optimization problems
[35], [117]	Improved Differential Evolution Algorithm	Enhanced version with better exploration and exploitation strategies	Better performance over basic DE	Increased complexity, requires fine-tuning	Improved convergence speed and solution quality compared to basic DE
[36], [118]	Cuckoo Search Algorithm	Based on brood parasitism of some cuckoo species	Simple, efficient for global optimization	Can converge prematurely, parameter sensitivity	Effective for various optimization problems, often outperforms some traditional algorithms

Study/Reference	Metaheuristic Algorithm	Main Concept	Advantages	Disadvantages	Note on Performance
[37], [119]	Discrete Cuckoo Search Algorithm	Adapted for discrete optimization problems	Effective for discrete problems	Can be complex to implement	Suitable for scheduling and combinatorial optimization problems
[38], [120]	Bat Algorithm	Inspired by the echolocation behavior of bats	Good balance of exploration and exploitation	May require fine-tuning, can converge prematurely	Effective for continuous optimization problems
[39], [121]	Parallel Bat Algorithm	Parallel implementation of Bat Algorithm for faster execution	Faster, maintains diversity	Complexity in implementation, requires parallel computing resources	Suitable for large-scale problems
[40]	Coral Reef Algorithm	Models coral reef formation and reproduction	Good balance of exploration and exploitation	Relatively new, less studied, parameter sensitivity	Promising results, effective for various optimization problems
[41], [122]	Parallel Coral Reef Algorithm	Parallel implementation for faster and scalable optimization	Faster, scalable, maintains diversity	Complexity in implementation, requires parallel computing resources	Suitable for large-scale and complex problems
[42], [123]	Grey Wolf Optimizer	Inspired by the social hierarchy and hunting mechanism of grey wolves	Simple, effective for various types of optimization problems	May converge prematurely, sensitive to parameter settings	Effective for multi-modal optimization problems
[43], [124]	Discrete Grey Wolf Optimizer	Adapted for discrete optimization problems	Effective for discrete problems	Can be complex to implement	Suitable for scheduling and combinatorial optimization problems
[125]	Modified Genetic Algorithm	Genetic Algorithm with modifications to improve performance	Improved performance over basic GA	Increased complexity, still may converge prematurely	Improved convergence speed and solution quality compared to basic GA
[126], [103]	Improved Genetic Algorithm	Enhanced version with better exploration and exploitation strategies	Better performance compared to basic GA	Increased complexity	Better results in terms of convergence speed and solution quality
[127]	Modified Ant Colony Optimization Algorithm	Enhanced version of ACO with better exploration and exploitation strategies	Improved performance over basic ACO	More complex, computationally expensive	Better convergence speed and solution quality compared to basic ACO
[128]	Enhanced Particle Swarm Optimization	Enhanced version with better mechanisms to avoid local optima and improve convergence	Better performance over basic PSO	Increased complexity, parameter sensitivity	Improved convergence speed and solution quality compared to basic PSO

Understanding the working principle of these commonly used metaheuristic algorithms in JSS is crucial for selecting appropriate solution approaches and designing effective scheduling strategies tailored to specific problem instances. The subsequent section (performance evaluation) of this review delves into each metaheuristic algorithms in more detail, analyzing their effectiveness, performance, and practical implications in industrial and manufacturing settings.

B. Advantages of Using Metaheuristics for JSS

This subsection discusses the advantages of employing metaheuristic algorithms as solution approaches for JSSP. Metaheuristic algorithms offer several benefits over traditional exact optimization methods when it comes to addressing the complexities of JSS, including their ability to efficiently explore large solution spaces, find high-quality solutions, and adapt to changing conditions.

- Flexibility and Adaptability: Metaheuristic algorithms are inherently flexible and adaptable, able to handling various problem instances and constraints [4]. By modifying the parameters, operators, and search strategies, they can be made to fit particular JSS scenarios, providing effective and customized solutions.
- Efficient Exploration of Solution Spaces: Scheduling issues in job shops usually involve large solution spaces with multiple possible schedules. Metaheuristic algorithms to effectively explore these solution spaces, finding promising areas and avoiding local optima, use iterative search strategies [9].
- Effective Handling of Complex Constraints: Complex constraints like machine dependencies, precedence relationships, and setup times are frequently present in JSSP [93]. In order to find workable and superior solutions, metaheuristic algorithms can efficiently navigate these constraints by taking into account several objectives and constraints at once [20].
- Scalability to Large Problem Instances: Large-scale JSSPs, which may be computationally demanding for precise optimization techniques,

are ideally suited for metaheuristic algorithms [91]. They provide scalable solutions that do not significantly impair performance even as problem sizes increase [129].

- Robustness to Uncertainty and Variability: Environments used for JSS are frequently unpredictable and variable due to things like unforeseen events, machine breaks down, or changes in processing times [56]. Metaheuristic algorithms demonstrate resilience against these kinds of uncertainties by dynamically adjusting to shifts in the scheduling context while preserving consistent performance [19].

Understanding the advantages of using metaheuristic algorithms for JSS is crucial for selecting appropriate solution approaches and designing effective scheduling strategies. This explores specific metaheuristic algorithms and their advantages in addressing JSSP, analyzing their effectiveness, performance, and practical implications in industrial and manufacturing settings.

4. Performance Evaluation

This section conducts a comprehensive performance evaluation and comparison of the various metaheuristic algorithms discussed in the preceding sections. By systematically assessing the effectiveness, efficiency, and robustness of these optimization methods across different problem instances and evaluation metrics, this aims to provide insights into their relative performance and practical implications for JSS applications.

Evolutionary Algorithm for JSS Optimization:

Yongsuo et al. [130] This study addresses dynamic JSS with extended constraints, using a GA to minimize the maximum completion time. The GA outperforms heuristic algorithms (weighted shortest processing time (WSPT), earliest due date (EDD), first come first serve (FCFS), longest processing time (LPT), critical ratio (CR)) in achieving better results across various scenarios. Siregar et al. [131] In an aluminum industry case, the GA significantly reduced the makespan from 55,970 to 46,637 hours compared to the FCFS method, improving efficiency by 20.13% and demonstrating the GA's effectiveness in production scheduling. Wang et al. [132] This research introduces the search economics for job-shop scheduling problem (SEJSP) algorithm for flexible JSS, which outperforms other algorithms like CRO and GA in minimizing makespan. SEJSP's efficient exploration and parameter settings lead to superior or competitive performance on benchmark instances. Mishra et al. [26] Evaluating various evolutionary algorithms (PSO, AIA, IWO, BFO, MBHS) on 250 benchmark instances, study finds that selective initial populations yield better results. Hybrid methods, particularly Hybrid PSO (HPSO), Hybrid AIA (HAIA), and IMBHS, show enhanced performance, providing effective solutions for single-objective JSSP. Hu et al. [133] This study proposes a differential evolutionary algorithm with uncertainty handling techniques (DEA_UHT) for stochastic reentrant JSSP (SRJSSP). DEA_UHT, utilizing hypothesis tests and optimal budget allocation, outperforms conventional methods, balancing computational efficiency and solution quality amidst uncertainties.

Heuristic algorithm and real world-constraints:

Huynh-Tuong et al. [134] This study addresses a teamwork scheduling problem with job-person constraints, using a Mixed integer linear programming (MILP) model, heuristics, and metaheuristic (Assignment algorithm based on FCFS (ASGN), Assignment algorithm based on SPT (ASPT), Assignment algorithm based on LPT (ALPT)), and SA to minimize makespan. The SA metaheuristic achieves the best solution quality but with longer computation time, while the other heuristics offer faster runtimes with lower solution quality. Future research may explore evolutionary algorithms for larger instances and additional constraints. Liu et al. [135] The study focuses on coordinated scheduling in shared manufacturing environments using a non-cooperative game model. The Nash equilibrium genetic algorithm (NE-GA) is proposed to minimize completion time and makespan while ensuring fair payoff distribution among customers. The NE-GA outperforms heuristic algorithms (FCFS, SPT, LPT) and other metaheuristic optimization techniques (SA, PSO). Future work may consider transportation capacity limits and theoretical convergence analysis. Campo et al. [136] This research addresses the flexible JSSP with real-world constraints using a GA integrated with fuzzy logic to minimize tardiness/earliness penalties. Tested on a fabric finishing production system, the proposed method outperforms traditional heuristics by over 30%, providing efficient and effective solutions. Future research may compare with other metaheuristics and explore additional constraints.

Novel algorithmic framework and optimization:

Zhang et al. [137] This study addresses the flexible JSSP with Lot Streaming (LSFJSP) using a DGWO. The DGWO algorithm shows strong robustness and superior convergence speed compared to GA and PSO, demonstrating competitive performance in computational experiments. Kumar et al. [138] The study compares Shifting Bottleneck (SB), GA, and PSO for solving JSSP. GA performs best for smaller problems, SB is faster but less reliable, and PSO struggles with larger problems due to random initialization. Future research may explore dynamic JSSP scenarios. Wang et al. [139] Introduces an ABC algorithm for the Flexible JSSP, focusing on minimizing maximum completion time. The ABC algorithm demonstrates effective global exploration and local exploitation, showing efficiency in simulations compared to existing methods. Zarrouk et al. [140] Presents a two-level PSO algorithm for the flexible job-shop scheduling problem (FJSP), where the upper level assigns operations to machines and the lower level manages sequencing. The algorithm improves efficiency by reducing evaluated solutions and demonstrates significant performance improvements in convergence and central processing unit (CPU) time. Hajibabaei et al. [141] Investigates the Flexible JSSP with unrelated parallel machines and sequence-dependent setup times (SDST). In study TS algorithm outperforms GA, particularly in instances with varying job numbers and processing times, confirmed by statistical validation. Dabah et al. [142] Focuses on the Blocking JSSP with zero buffer capacity. The study proposes a parallelized TS algorithm, demonstrating significant improvements in solution quality and exploration efficiency over sequential approaches, highlighting the benefits of parallelism.

Comparative study and Algorithm performance:

Alharkan et al. [143] This study introduces TS and GPSO for scheduling jobs on two identical parallel machines with a single server to minimize makespan, an NP-hard problem. TS and Greedy PSO (GPSO) are compared to SA, GA, and Local search (I-L) algorithms, showing strong performance, especially for medium and large instances, with TS performing best overall except in cases with 8 and 200 jobs. Yu et al. [144] Provides a comprehensive overview of JSSP and solutions, emphasizing the importance of efficient scheduling in manufacturing. It discusses three main approaches GA, TS, and SA highlighting their principles, advancements, and applications. The study underscores the need for integrating diverse algorithms and methodologies to enhance scheduling efficiency and precision. Hasani et. al [145] examines scheduling n jobs on two parallel machines with a single server to minimize the makespan, proposing two heuristic algorithms: one to minimize machine idle time and another to minimize gaps between job loadings. Experiments reveal that for small to medium instances (up to 400 jobs), SA and GA methods outperform the heuristics. However, for larger instances (500 jobs and above), the proposed I-L Algorithm proves superior, remaining efficient even for very large instances up to 10,000 jobs. Hasani et al. [146] Focuses on minimizing makespan by scheduling jobs on two parallel machines with a single server, using SA and GA. These methods are tested on instances with up to 1000 jobs, showing that SA and GA outperform previous algorithms, particularly for instances up to 250 jobs. SA tends to perform better for larger instances, although GA occasionally reaches the lower bound more frequently for very large instances. Future research will explore hybrid algorithms and different objective functions.

Parallel Genetic Algorithm (PGA) for JSS optimization:

Defersha et al. [147] This study explores the Flexible JSSP, allowing operations to be assigned to multiple machines, considering SDST, machine release dates, and time lags. A PGA is proposed to solve this model efficiently. Numerical examples show that PGA significantly enhances computational performance, particularly for medium to large problem instances, where traditional sequential GA struggles. Future work will extend this approach to handle multiple objectives and additional constraints. Abdullah et al. [148] Focuses on the NP-hard JSSP and introduces a PGA with adaptive genetic operators and a migration operation to improve results and reduce computation time. Extensive experiments reveal that adaptive operators and parallelism significantly enhance scheduling quality and efficiency. The PGA effectively minimizes job completion times by scheduling tasks across multiple machines and jobs, demonstrating notable improvements in both scheduling quality and computational efficiency.

Meta-heuristic algorithm for scheduling problem and optimization:

Liu, Z. et al. [149] This study focuses on the Flexible JSSP and proposes an enhanced GA with a three-layer coding mechanism to optimize batching strategies and subsequent scheduling. Experimental results show significant improvements in flexible JSSP optimization compared to traditional algorithms, highlighting the algorithm's effectiveness in stabilizing production in batch job shops. Future research may involve exploring more advanced optimization algorithms and addressing uncertainties in dynamic production environments. Kumar, P. et al. [150] This study presents a MoGA approach for solving the JSSP to optimize makespan in manufacturing systems. The algorithm effectively reduces makespan for specific problems sourced from literature, highlighting the significance of considering short processing time and transportation time in JSS optimization. Future research aims to consider additional factors like maintenance time and setup time in machine scheduling. Habbadi, S. et al. [151] Discusses the application of GA in solving the JSSP, focusing on manual implementation to understand the procedure and decision-making involved in finding optimal solutions. The study emphasizes the importance of population initialization and coding in GA processes, shedding light on practical applications of mathematics and programming for complex scheduling problems. Abdullah, N. et al. [152] Introduces a modified PSO algorithm to efficiently solve the JSSP. By overcoming idle particle positions, the enhanced PSO algorithm demonstrates improved performance in finding optimal schedules across different problem instances, effectively tackling the complexities of the JSSP. Ali, B. et al. [153] Focuses on the Dynamic JSSP and proposes a GA approach to minimize makespan while considering setup times and precedence constraints. The experiments demonstrate the superior performance of the GA-based approach in dynamic scheduling scenarios, highlighting the competitiveness and effectiveness of GA in minimizing makespan. Shen, Z. et al. [154] Addresses the JSSP in a brewery production setting, implementing GA, SA, and ACO algorithms for optimization. Results show significant improvements in production time for all three algorithms compared to the unoptimized case, with GA performing the best, highlighting the effectiveness of metaheuristic methods in optimizing complex production processes. Janes, G. et al. [155] Utilizes a GA to efficiently find schedules for various real-world scenarios of the JSSP, demonstrating satisfactory results and potential cost savings by using fewer machines. The study emphasizes the algorithm's readiness for industrial application and provides insights for system performance evaluation. Salido et al. [156] Addresses the JSSP with Machine Speed Scaling (JSMS), proposing a GA to efficiently solve this NP-hard problem by optimizing both makespan and energy consumption objectives. The GA presents a promising approach for addressing large-scale scheduling problems with energy-aware optimization requirements. Chaouch et al. [127] Explores bio-inspired algorithms, including Ant System (AS), Ant Colony System (ACS), and MACO, to tackle the Distributed JSSP. MACO emerges as superior, consistently outperforming AS and ACS across different instances, demonstrating promising results in addressing the challenging Distributed JSSP. Teekeng et al. [128] Introduces EPSO, a novel algorithm based on PSO, designed to solve the Flexible JSSP. EPSO demonstrates superior performance compared to existing optimization methods, consistently achieving solutions equal to or better than lower bounds of benchmarks, paving the way for future research in complex flexible JSSP scenarios.

Multi-objective optimization and advanced algorithmic technique:

Zhang et al. [157] Introduces an IGA for the Flexible JSSP, aiming to minimize makespan, total setup time, and total transportation time while considering constraints such as processing time, setup time, and transportation time. Experimental studies validate the effectiveness of the proposed approach across various datasets, demonstrating superior performance compared to existing algorithms in terms of finding non-dominated solutions and optimizing objective values. The IGA consistently produces high-quality solutions across different problem sizes and complexities, confirming its efficacy in addressing FJSP with multiple time constraints. Future research may explore the relationship between initial and final solutions and enhance genetic operators using individual fitness information. Sel et al. [158] Addresses the Dynamic JSSP by introducing a Simulation Model (SM) that incorporates machine failures and changing due dates. Three scheduling rules (SRs) are integrated into the SM, and a SA based simulation-optimization approach is proposed to find optimal schedules in the dynamic system. Results indicate that SA closely approximates the performance of the shortest SPT rule with reasonable computational burden, outperforming other scheduling rules. The study suggests possible extensions, including generalizing the SM to handle other types of disturbances and incorporating different heuristics for larger-scale applications. Zhang et al. [159] Proposes a multi-population genetic algorithm for the multi-objective scheduling of flexible JSP, aiming to reduce the longest makespan of workpieces, the load on each machine, and the total machine load simultaneously. By considering factors such as shortest processing time and balanced machine utilization, the method efficiently allocates machines and simplifies the scheduling process. The algorithm demonstrates superior performance compared to conventional algorithms in terms of population quality, initial solution quality, and convergence rate, offering an effective solution to the multi-objective scheduling challenges in flexible job-shop environments.

Dynamic scheduling and rescheduling strategy:

Zhang et al. [160] introduce the Improved Heuristic Kalman Algorithm (IHKA) for Dynamic JSSP, utilizing a cellular neighbor network to efficiently find optimal schedules. Compared to other methods like HKA and Genetic Algorithm-Mixed (GAM), IHKA demonstrates superior performance, quickly generating practical solutions for real-world factories. Singh et al. [161] tackle the Flexible JSSP using Quantum-Behaved Particle Swarm Optimization (QPSO), which integrates GA mutation and chaotic numbers for enhanced global search capabilities, showing promising results in minimizing makespan across various datasets. Wang et al. [162] address Dynamic JSSP in manufacturing, developing a mixed-integer programming model and enhanced PSO with modified decoding schemes and population initialization strategies to efficiently handle rescheduling tasks, demonstrating superior performance compared to existing methods across diverse instances.

Integration of Meta-heuristic and machine learning for scheduling problem:

Lin et. al [163] The Learning-based Cuckoo Search (LCS) algorithm, integrated with machine learning techniques like auto encoders and factorization machines (FM), efficiently tackles the flexible JSSP. By compressing solution spaces and dynamically adjusting parameters through reinforcement learning (RL), LCS achieves faster convergence and higher-quality solutions compared to conventional methods. Computational experiments demonstrate LCS's superiority over heuristic rules, metaheuristics, and even IBM CPLEX Interactive Optimizer, particularly in large-scale scenarios. This amalgamation of machine learning and metaheuristics positions LCS as a potent tool for resolving complex Flexible JSSPs, with future improvements focusing on enhancing optimization through advanced machine learning methods. Zebari et. al [164] In addressing the Multi-Objective Flexible JSSP, this research proposes a hybrid approach integrating the Hybrid BA and SA. Given the complexities involved in optimizing conflicting objectives like makespan reduction and production cost minimization, conventional approaches often struggle. The Hybrid BA and SA approach outperforms individual algorithms and other state-of-the-art techniques by leveraging the exploration capabilities of BA and the exploitation strategies of SA. Through comprehensive experimental evaluation, the hybrid approach demonstrates superior performance in exploring the solution space, balancing exploitation and exploration, and generating diverse, high-quality solutions across various scenarios.

Table 2 presents the summary of comparative analysis of various metaheuristic algorithms based category of algorithm, objective(s), and problem addressed. It also highlights the key findings of each study reviewed.

Table 2 Summary of the Comparative analysis of metaheuristic algorithm

Types	Study	Objective(s)	Problem Addressed	Methodology	Key findings
Evolutionary Algorithm for JSS optimization	[130]	MMCT	Dynamic JSSP with extended process constraints.	GA compared with WSPT, EDD, FCFS, LPT, and CR algorithms.	GA outperforms heuristic algorithms in minimizing maximum completion time.
	[131]	MMK	Scheduling in the aluminum industry	GA, and FCFS	GA reduces makespan by 20.13% and shows better efficiency
	[132]	MMK	FJSSP	SEJSP Algorithm, CRO, and GA.	SEJSP achieves competitive or superior performance
	[26]	MMK	JSSP	PSO, AIA, IWO, BFO, MBHS	Hybrid methods show improved performance
	[133]	ISE, ISQ	SRJSSP	DEA with uncertainty handling techniques (hypothesis test technique (HTT) and optimal computing budget allocation technique (OCBAT))	DEA_UHT is robust and effective, outperforming conventional methods
Heuristic algorithm and real world-constraints	[134]	MMK	Teamwork scheduling with job-person constraints	MILP model, heuristics (ASGN, ASPT, ALPT, SA)	SA achieves best solution quality; ASGN, ASPT, ALPT offer faster runtime
	[135]	MMCT, MMK	Coordinated scheduling of parallel machine production and transportation	Nash equilibrium genetic algorithm (NE-GA)	NE-GA minimizes makespan effectively
	[136]	MTD, MEP	FJSSP with SDST, due windows, uncertainties	GA with fuzzy logic	Reduces tardiness/earliness penalties by over 30%
Novel algorithmic framework and optimization	[137]	RBS, CVS	LSFJSP	DGWO algorithm	DGWO shows strong robustness, superior convergence speed and accuracy
	[138]	RBS, CVS, ACC	JSSP	SB, GA, and PSO	GA performs best for smaller problems, SB is faster but less reliable
	[139]	MMCT	Flexible Job-Shop Scheduling Problem (FJSP)	ABC algorithm	ABC algorithm shows efficiency and effectiveness
	[140]	WSQ	FJSP with two-level PSO	Two-level PSO algorithm	Significant improvements in convergence and CPU time
	[141]	MST	FJJSP with unrelated parallel machines and SDST	MILP, TS algorithm	TS outperforms GA, applicable in diverse industries
	[142]	ISQ, IST	Blocking JSSP (BJSS) with zero buffer capacity	Parallelized TS algorithm	Parallel TS methods enhance solution quality

Types	Study	Objective(s)	Problem Addressed	Methodology	Key findings
Comparative study and Algorithm performance	[143]	MMK	Scheduling on two identical parallel machines with a single server	TS, GPSO	TS and GPSO perform well for medium- and large-scale instances
	[144]	ISE	JSSP	GA, TS, SA	advancements and practical applications of each method
	[145]	MMK	Scheduling on two parallel machines with a single server.	Heuristic algorithms minimizing makespan with two parallel machines and a single server.	Proposed heuristic algorithms (Min-idle, Min-load gap) perform well for larger instances (500 jobs and above), outperforming other methods.
	[146]	MMK	Scheduling on two parallel machines with a single server	SA and GA	Composite neighborhood approach outperforms previous algorithms
Parallel Genetic Algorithm (PGA) for JSS optimization	[147]	SDAT, TLR, MRD	Flexible JSSP with multiple constraints	PGA	PGA improves computational performance, especially for medium-sized problems
	[148]	MPT	JSSP, NP-hard nature	PGA with adaptive operators and migration	Adaptive operators and parallelism improve results and reduce computation time
Meta-heuristic algorithm for scheduling problem and optimization	[149]	ISE	FJSSP	Enhanced GA with three-layer coding and operation overlapping strategy	Significant improvements in computational efficiency and solution quality
	[150]	MMK	JSSP	Modified GA	Reductions in makespan for specific problems
	[151]	MMK	JSSP optimization using GA	Manually implemented GA	Emphasizes importance of population initialization and coding
	[152]	ISE	JSSP with PSO algorithm	Modified PSO	Outperforms standard PSO in finding optimal schedules
	[153]	MMK	DJSSP with setup times and precedence constraints	GA Approach	GA shows superior performance in minimizing makespan
	[154]	PEF	JSSP in brewery production	GA, SA, and ACO	GA saves 35% in production time; SA and ACO also improve production time
	[155]	CSV, MMK	JSSP with modified GA operations	GA	Achieves reduced makespan in short time for real-world scenarios.
	[156]	MMK, MEC	JSSP with Machine Speed Scaling (JSMS)	GA	GA optimizes makespan and energy consumption compare to the IBM ILOG CPLEX CP Optimizer.
	[127]	ACC	Distributed JSSP (DJSP)	Ant System (AS), Ant Colony System (ACS), MACO	MACO minimizes makespan effectively
	[128]	ISQ	Flexible JSSP	Enhanced PSO (EPSO), AIA	EPSO outperforms AIA and Demir's model

Types	Study	Objective(s)	Problem Addressed	Methodology	Key findings
Multi-objective optimization and advanced algorithmic technique	[157]	MMK, MST, MTT	FJSP with constraints	Improved Genetic Algorithm (IGA)	Experimental studies validate the effectiveness of the proposed approach across various datasets, demonstrating superior performance compared to existing algorithms.
	[158]	ISE	Dynamic JSSP (DJSP)	Simulation model with SA, EDD, SPT, FCFS	SPT yields the best performance; SA closely approximates SPT with reasonable computational burden, outperforming EDD and FIFO.
	[159]	MMK, MFT	Multi-objective FJSP	Multi-population Genetic Algorithm	Efficiently allocates machines, reduces total machine load
Dynamic scheduling and rescheduling strategy	[160]	ISE	Dynamic JSSP (DJSSP)	Improved Heuristic Kalman Algorithm (IHKA) utilizing a cellular neighbor network	IHKA outperforms HKA and GAM, and Quickly generates practical solutions for real-world factories
	[161]	MMK	Flexible Job-Shop Scheduling Problem (FJJSP)	Quantum-Behaved Particle Swarm Optimization (QPSO) integrating genetic algorithm mutation and chaotic numbers	QPSO shows promising results in minimizing makespan across various datasets
	[162]	WSQ	DJSSP	Mixed-integer programming model with enhanced Particle Swarm Optimization (PSO)	Enhanced PSO shows superior performance in handling rescheduling tasks compared to existing methods
Integration of Meta-heuristic and machine learning for scheduling problem	[163]	CVS, ISQ	FJSSP	Learning-based Cuckoo Search (LCS) integrating machine-learning techniques	LCS achieves better solutions in less CPU time than CPLEX, heuristic rules, and metaheuristics.
	[164]	MMK, CSV	Multi-Objective Flexible Job-Shop Scheduling Problem (MOFJSSP)	Hybrid Bat Algorithm (BA) and SA	Hybrid BA and SA improve convergence rates and solution quality. Better balance between exploitation and exploration compared to other techniques.

This rigorous comparative analysis provides valuable insights into the performance and applicability of metaheuristic algorithms for JSS. This analysis contribute to the understanding of algorithm behavior and guide practitioners and researchers in choosing appropriate optimization methods for their scheduling tasks.

5. Challenges and Future Directions

This section discusses challenges faced by current metaheuristic algorithms and emerging trends in the field of JSS optimization. Identify the key challenges and potential avenues for future research and development, this aims to stimulate further advancements in algorithm design, problem modeling, and practical implementation. This section provide insights into the unresolved issues and opportunities for innovation in the domain of JSS, guiding future research efforts and industry practices.

A. Challenges in Applying Metaheuristic Algorithms to Job Shop Scheduling

This sub section provide an overview of common metaheuristic algorithms used in JSS. Metaheuristic algorithms are powerful optimization techniques that offer flexible and robust approach to find optimal solutions to complex scheduling problems. By exploring these common metaheuristic algorithms, this paper aims to elucidate their principles, operational characteristics, and advantages in addressing various scheduling challenges in different environments. Despite their effectiveness, applying metaheuristic algorithms to JSS encounters several challenges. Understanding and addressing these challenges is crucial for further improving the performance and applicability of optimization techniques in manufacturing environments.

Problem Complexity:

- Description: JSSPs pose inherent complexity owing to the combinatorial nature of optimization tasks, the existence of multiple conflicting objectives, and the presence of diverse constraints.
- Impact: The complexity of JSS poses challenges for metaheuristic algorithms, requiring them to efficiently explore the vast solution space and balance competing objectives while respecting constraints.

High-Dimensional Search Space:

- Description: JSSPs typically entail navigating a high-dimensional search space, characterized by a multitude of decision variables representing machine assignments, operation sequences, and job schedules.
- Impact: Navigating high-dimensional search spaces presents challenges for metaheuristic algorithms in terms of exploration, exploitation, and convergence speed, as well as memory and computational resource requirements.

Dynamic and Uncertain Environments:

- Description: Manufacturing environments are dynamic and subject to uncertainties such as machine breakdowns, unexpected job arrivals, and changes in demand or resource availability.
- Impact: Metaheuristic algorithms must adapt to dynamic and uncertain environments, requiring robustness, flexibility, and adaptability to handle changing conditions and maintain high-quality solutions.

Scalability and Efficiency:

- Description: As manufacturing systems grow in complexity and scale, the scalability and efficiency of metaheuristic algorithms become increasingly important.
- Impact: Ensuring the scalability and efficiency of algorithms is essential for handling large-scale scheduling instances with many jobs, machines, and production constraints, as well as meeting real-time or near-real-time scheduling requirements.

Incorporation of Domain Knowledge:

- Description: Domain-specific knowledge, such as production rules, machine capabilities, and scheduling preferences, can significantly influence the effectiveness of scheduling solutions.
- Impact: Integrating domain knowledge into metaheuristic algorithms poses challenges in terms of knowledge representation, incorporation methods, and balancing the use of explicit knowledge with algorithmic exploration and exploitation.

Multi-Objective Optimization:

- Description: JSS frequently entails managing multiple conflicting objectives, including minimization makespan, reducing tardiness, and optimizing resource utilization.
- Impact: Metaheuristic algorithms must effectively handle multi-objective optimization, balancing trade-offs between competing objectives and generating Pareto-optimal solutions that represent meaningful compromises.

These challenges requires interdisciplinary research efforts combining expertise in optimization, operations research, computer science, and manufacturing engineering. By overcome to these challenges, researchers and practitioners can further advance the capabilities and practical applicability of metaheuristic algorithms for JSS, leading to more efficient and agile manufacturing systems.

B. Potential Research Directions and Areas for Improvement

Identify the potential research directions and areas for improvement is essential for advancing the field of JSS optimization using metaheuristic algorithms. By focus on the key challenges and emerging trends, researchers can address current limitations and drive innovation in algorithm design, problem modeling, and practical implementation.

Development of Hybrid and Advanced Techniques:

- Explore novel combinations of metaheuristic algorithms, machine learning techniques, and optimization paradigms to develop hybrid and advanced approaches for JSS. Investigate the integration of reinforcement learning, deep learning, and other AI-based methods with metaheuristic algorithms to enhance solution quality, scalability, and adaptability.

Multi-Objective and Robust Optimization:

- Develop metaheuristic algorithms capable of efficiently solving multi-objective JSSPs, balancing conflicting objectives and generating Pareto-optimal solutions. Explore robust optimization techniques that can handle uncertainties and dynamic changes in manufacturing environments, ensuring the resilience and reliability of scheduling solutions.

Scalability and Efficiency Improvements:

- Develop scalable metaheuristic algorithms capable of efficiently handling large-scale JSS instances with thousands of jobs, machines, and production constraints. Investigate parallel and distributed optimization techniques, metaheuristic ensembles, and adaptive search strategies for improving algorithm efficiency and reducing computational overhead.

Dynamic and Real-Time Scheduling:

- Address the challenges of dynamic and real-time JSS by developing metaheuristic algorithms that can adaptively respond to changing production conditions and resource constraints. Explore online optimization techniques, predictive modeling approaches, and reactive scheduling strategies for optimizing scheduling decisions in real-time manufacturing environments.

Explainable AI and Decision Support Systems:

- Develop metaheuristic algorithms with enhanced interpretability and explainability, enabling stakeholders to understand and trust scheduling solutions generated by optimization techniques. Investigate visualization tools, decision support systems, and interactive interfaces for facilitating human-computer collaboration in the JSS process.

These research directions and areas for improvement helps the researchers can advance the state-of-the-art in metaheuristic optimization for JSS, leading to more efficient, agile, and resilient manufacturing systems. These efforts contribute to the development of innovative solutions that address the evolving challenges and requirements of modern manufacturing environments.

C. Application Guidelines for Using Metaheuristic Algorithms in Future Work

For future research and practice, these following guideline offers a thorough method for applying metaheuristic algorithms to JSSPs in an efficient manner. To handle the complexity and limitations of job shop environments, researchers and practitioners can design, implement, and assess metaheuristic algorithms by following these steps.

- Problem Definition and Modelling:

  Identify Objectives: The scheduling problem's objectives (such as minimizing makespan, reducing tardiness, and maximizing resource utilization) should be clearly defined.

  Define Constraints: Determine every relevant limitations, such as processing times, resource constraints, precedence relationships, and machine availability.

  Formulate the Problem: Create a computational or mathematical model of the JSS issue, making sure that it accurately represent the actual situation.

- Selection of Metaheuristic Algorithm:

  Algorithm Choice: Based on the characteristics of the problem and the goals of the research, select an appropriate metaheuristic algorithm. Popular options include TS, PSO, ACO, GA, and SA.

  Hybrid Approaches: To improve performance, think about combining the chosen metaheuristic with additional optimization strategies (such as machine learning, exact methods, or local search).

- Algorithm Design and Implementation:

  Parameter Tuning: Determine the metaheuristic algorithm's important parameters (population size, mutation rate, cooling schedule, etc.) and optimize those using parameter-tuning strategies (grid search, response surface methodology, etc.).

  Algorithm Customization: Modify the algorithm to include constraints and knowledge unique to the problem. This could entail creating neighborhood structures, encoding schemes, or specialized operators.

  Adaptive Mechanisms: In order to balance exploration and exploitation, implement adaptive mechanisms that dynamically adjust algorithm parameters based on the search progress.

- Computational Experiments:

  Benchmarking: To assess the algorithm's performance in comparison to current methods, test it on common benchmark instances of JSSPs.

  Scalability Testing: Assess the algorithm’s scalability by applying it to problems of varying sizes and complexities to ensure it can handle large-scale instances.

- Real-World Application:

  Data Collection: Gather real-world information from workshop settings, such as schedule restrictions, machine specifications, and job specifications.

Customization for Real-World Use: Modify the algorithm to accommodate particular real-world limitations and specifications, such as material handling considerations, operator availability, and machine maintenance schedules.

Simulation and Validation: Prior to implementing the algorithm in real-world settings, validate its performance in a controlled environment using simulation models.

Researchers and practitioners effectively use the metaheuristic algorithms to address the challenges of JSS, improve production efficiency, and stimulate innovation in manufacturing processes by adhering to this application guideline.

D. Trends in Metaheuristic-Based Approaches for Job Shop Scheduling

This sub-section explores emerging trends and advancements in metaheuristic-based approaches for JSS optimization. These trends reflect recent developments and innovations that have the potential to shape the future of scheduling techniques in manufacturing and production environments.

Hybridization with Machine Learning Techniques:

- Description: Integration of metaheuristic algorithms with machine learning techniques such as reinforcement learning, neural networks, and deep learning.
- Advantages: Enhances the learning and adaptation capabilities of metaheuristic algorithms, enabling them to leverage historical data, learn from experience, and improve solution quality over time.

Metaheuristic-Driven Optimization Platforms:

- Description: Development of optimization platforms and frameworks that provide a unified environment for implementing, testing, and deploying metaheuristic-based scheduling solutions.
- Advantages: Facilitates rapid prototyping, experimentation, and deployment of metaheuristic algorithms for JSS tasks, fostering collaboration and knowledge sharing among researchers and practitioners.

Multi-Objective and Pareto-Based Optimization:

- Description: Focus on multi-objective optimization techniques that generate Pareto-optimal solutions, balancing conflicting objectives such as makespan minimization, tardiness reduction, and resource utilization optimization.
- Advantages: Enables decision-makers to explore trade-offs between competing objectives and make informed decisions based on the Pareto front, leading to more flexible and adaptable scheduling solutions.

Explainable AI and Transparent Optimization:

- Description: Emphasis on explainable AI techniques and transparent optimization methods that provide insights into the decision-making process and rationale behind scheduling solutions generated by metaheuristic algorithms.
- Advantages: Enhances trust, understanding, and acceptance of scheduling solutions among stakeholders, enabling effective communication, collaboration, and decision-making in manufacturing environments.

Real-Time and Adaptive Scheduling Strategies:

- Description: Creating real-time and adaptive scheduling strategies involves creating approaches that dynamically adapt scheduling decisions to accommodate evolving production conditions, resource availability, and fluctuations in demand.
- Advantages: Enables agile, responsive, and adaptive scheduling in dynamic manufacturing environments, improving production efficiency, flexibility, and resilience.

Embracing these emerging trends and advancements, researchers and practitioners leverage the full potential of metaheuristic-based approaches for addressing the evolving challenges and requirements of JSS in modern manufacturing environments. These trends represent exciting opportunities for innovation and improvement in scheduling techniques, paving the way for more efficient, flexible, and intelligent manufacturing systems.

6. Conclusion

This review paper provides valuable insights into the application of metaheuristic algorithms for addressing the complex problem of JSS. Through a comprehensive analysis of various metaheuristic approaches such as GA, SA, PSO, ACO, and others, the paper highlights their working principle, strength and suitability for different JSS scenarios. The analysis pertaining to efficiency, adaptability, computational complexity and robustness of various metaheuristic algorithms like GA, PSO, ACO, and SA in JSS is discussed in performance evaluation. GA, PSO, ACO, and SA each have unique strengths. GA and ACO are known for their high solution quality and adaptability. PSO converges quickly but is sensitive to parameter settings. SA is effective in avoiding local optima but can be computationally intensive. Together, these algorithms are versatile for various JSS challenges. In JSS, GA have moderate computational complexity and are suitable for large problems. PSO has relatively low computational overhead. ACO is computationally intensive due to pheromone updates. SA is straightforward but can be computationally demanding. Key findings suggest that no single metaheuristic consistently outperforms others across all problem instances. Instead, the performance of each algorithm is heavily dependent on problem-specific characteristics and parameter tuning. Emerging trends in this field include the integration of advanced techniques like machine learning and real time strategies, as well as the exploration of hybrid metaheuristic approaches for improved performance. Potential research directions encompass the development of novel metaheuristic algorithms inspired by hybrid and advanced techniques, as well as the investigation of multi-objective JSSPs to balance conflicting objectives. Challenges in this domain revolve around the scalability, uncertain environment, and problem complexity of metaheuristic algorithms. Additionally, the lack of standardized benchmark datasets and performance evaluation metrics poses a challenge to algorithm comparison and validation. Addressing these challenges will require interdisciplinary collaboration and innovative methodologies to advance the efficiency and effectiveness of JSS optimization techniques. Thus, this review underscores the importance of continued research and development in metaheuristic algorithms to enhance scheduling practices and optimize manufacturing operations.

Conflicts of Interest

The authors have no conflicts of interest to declare.

Authors' Contribution Statement

DCH: Conceptualization, data acquisition, data collection, data curation, data analysis, writing original draft, and interpretation of results. **SSP:** Data analysis, design, data representation, interpretation of results, writing original draft, and reading proof. **SMP:** Conceptualization, study conception, writing original draft, interpretation of results, supervision, review, reading proof, and the revision of the whole article.

References

[1] M. Geurtsen, J. B. H. C. Didden, J. Adan, Z. Atan, and I. Adan, "Production, maintenance and resource scheduling: a review," European Journal of Operational Research, vol. 305, no. 2, pp. 501–529, Mar. 2023, doi: 10.1016/j.ejor.2022.03.045.

[2] L. Xue, S. Zhao, A. Mahmoudi, and M. R. Feylizadeh, "Flexible job-shop scheduling problem with parallel batch machines based on an enhanced multi-population genetic algorithm," Complex & Intelligent Systems, vol. 10, no. 3, pp. 4083–4101, Jun. 2024, doi: 10.1007/s40747-024-01374-7.

[3] T. Dokeroglu, T. Kucukyilmaz, and E.-G. Talbi, "Hyper-heuristics: a survey and taxonomy," Computers & Industrial Engineering, vol. 187, p. 109815, Jan. 2024, doi: 10.1016/j.cie.2023.109815.

[4] V. Tomar, M. Bansal, and P. Singh, "Metaheuristic algorithms for optimization: a brief review," Multidisciplinary digital publishing institute, Mar. 2024, p. 238. doi: 10.3390/engproc2023059238.

[5] G. Da Col and E. C. Teppan, "Industrial-size job shop scheduling with constraint programming," Operations Research Perspectives, vol. 9, p. 100249, 2022, doi: 10.1016/j.orp.2022.1002490.

[6] R. A. Liaqait, S. Hamid, S. S. Warsi, and A. Khalid, "A critical analysis of job shop scheduling in context of industry 4.0," Sustainability, vol. 13, no. 14, p. 7684, Jul. 2021, doi: 10.3390/su13147684.

[7] J. Zhang, G. Ding, Y. Zou, S. Qin, and J. Fu, "Review of job shop scheduling research and its new perspectives under Industry 4.0," Journal of Intelligent Manufacturing, vol. 30, no. 4, pp. 1809–1830, Apr. 2019, doi: 10.1007/s10845-017-1350-2.

[8] X. Wu, X. Yan, D. Guan, and M. Wei, "A deep reinforcement learning model for dynamic job-shop scheduling problem with uncertain processing time," Engineering Applications of Artificial Intelligence, vol. 131, p. 107790, May 2024, doi: 10.1016/j.engappai.2023.107790.

[9] D. Vivekanandan, S. Wirth, P. Karlbauer, and N. Klarmann, "A reinforcement learning approach for scheduling problems with improved generalization through order swapping," Journal of computer science and artificial intelligence, Mar 2023, doi: 10.48550/ARXIV.2302.13941.

[10] G. Infantes, S. Roussel, P. Pereira, A. Jacquet, and E. Benazera, "Learning to Solve Job Shop Scheduling under Uncertainty," Journal of computer science and artificial intelligence, Mar 2024, doi: 10.48550/ARXIV.2404.01308.

[11] Z. Zhou, L. Xu, X. Ling, and B. Zhang, "Digital-twin-based job shop multi-objective scheduling model and strategy," International Journal of Computer Integrated Manufacturing, vol. 37, no. 1–2, pp. 87–107, Feb. 2024, doi: 10.1080/0951192X.2023.2204475.

[12] P. Penchev, P. Vitliemov, and I. Georgiev, "Optimization model for production scheduling taking into account preventive maintenance in an uncertainty-based production system," Heliyon, vol. 9, no. 7, pp. e17485, Jul. 2023, doi: 10.1016/j.heliyon.2023.e17485.

[13] M. M. Teshome, T. Y. Meles, and C.-L. Yang, "Productivity improvement through assembly line balancing by using simulation modeling in case of Abay garment industry Gondar," Heliyon, vol. 10, no. 1, pp. e23585, Jan. 2024, doi: 10.1016/j.heliyon.2023.e23585.

[14] S. Khorasani, E. Körpeoğlu, and V. V. Krishnan, "Dynamic Development Contests," Operations Research, vol. 72, no. 1, pp. 43–59, Jan. 2024, doi: 10.1287/opre.2021.0420.

[15] M. Aghelinejad, Y. Ouazene, and A. Yalaoui, "Production scheduling optimisation with machine state and time-dependent energy costs," International Journal of Production Research, vol. 56, no. 16, pp. 5558–5575, Aug. 2018, doi: 10.1080/00207543.2017.1414969.

[16] Y. M. Omar, M. Minoufekr, and P. Plapper, 'Business analytics in manufacturing: current trends, challenges and pathway to market leadership', Operations Research Perspectives, vol. 6, pp. 100127, Oct. 2019, doi: 10.1016/j.orp.2019.100127.

[17] Z.-Y. Wang and C. Lu, "An integrated job shop scheduling and assembly sequence planning approach for discrete manufacturing," Journal of Manufacturing Systems, vol. 61, pp. 27–44, Oct. 2021, doi: 10.1016/j.jmsy.2021.08.003.

[18] P. J. Lederer and L. Li, "Pricing, production, scheduling, and delivery-time competition," Operations Research, vol. 45, no. 3, pp. 407–420, 1997, Accessed: May 06, 2024. [Online]. Available: https://www.jstor.org/stable/172018.

[19] A. M. Nassef, M. A. Abdelkareem, H. M. Maghrabie, and A. Baroutaji, "Review of metaheuristic optimization algorithms for power systems problems," Sustainability, vol. 15, no. 12, pp. 9434, Jun. 2023, doi: 10.3390/su15129434.

[20] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," Complex & Intelligent Systems, vol. 56, no. 11, pp. 13187–13257, Nov. 2023, doi: 10.1007/s10462-023-10470-y.

[21] R. P. Badoni et al., "An exploration and exploitation-based metaheuristic approach for university course timetabling problems," Axioms, vol. 12, no. 8, pp. 720, Jul. 2023, doi: 10.3390/axioms12080720.

[22] L. Davis, "Job shop scheduling with genetic algorithm," In Proceedings of the First International Conference on Genetic Algorithms and their Applications, J. J. Grefenstette, Ed., New York, Psychology Press, 1985, pp. 136-140. doi: 10.4324/9781315799674.

[23] R. W. Eglese, "Simulated annealing: a tool for operational research," European Journal of Operational Research, vol. 46, no. 3, pp. 271–281, Jun. 1990, doi: 10.1016/0377-2217(90)90001-R.

[24] M. Widmer, "Job Shop Scheduling with Tooling Constraints: a Tabu Search Approach," Journal of the Operational Research Society, vol. 42, no. 1, pp. 75–82, Jan. 1991, doi: 10.1057/jors.1991.9.

[25] S.-C. Lin, E. D. Goodman, and W. F. Punch, "Investigating parallel genetic algorithms on job shop scheduling problems," in Evolutionary Programming VI, vol. 1213, P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 383–393. doi: 10.1007/BFb0014827.

[26] S. K. Mishra and C. S. P. Rao, "Performance comparison of some evolutionary algorithms on job shop scheduling problems," *IOP Conference Series: Materials Science and Engineering*, vol. 149, pp. 012041, Sep. 2016, doi: 10.1088/1757-899X/149/1/012041.

[27] M. den Besten, T. Stützle, and M. Dorigo, "Ant Colony Optimization for the Total Weighted Tardiness Problem," in Parallel Problem Solving from Nature PPSN VI, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds., Berlin, Heidelberg: Springer, 2000, pp. 611–620. doi: 10.1007/3-540-45356-3_60.

[28] H. Madivada and R. C.S.P, "Improved Music Based Harmony Search (IMBHS) for Solving Job Shop Scheduling Problems (JSSPS)," International journal of Programming Languages and applications, vol. 3, no. 3, pp. 1–15, Jul. 2013, doi: 10.5121/ijpla.2013.3301.

[29] S. Matsui, I. Watanabe, and K. Tokoro, "Real-Coded Parameter-Free Genetic Algorithm for Job-Shop Scheduling Problems," in Parallel Problem Solving from Nature — PPSN VII, vol. 2439, J. J. M. Guervós, P. Adamidis, H.-G. Beyer, H.-P. Schwefel, and J.-L. Fernández-Villacañas, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 800–810. doi: 10.1007/3-540-45712-7_77.

[30] H. W. Ge, Y. C. Liang, Y. Zhou, and X. C. Guo, "A particle swarm optimization-based algorithm for job-shop scheduling problems," international journal of computational methods, vol. 02, no. 03, pp. 419–430, Sep. 2005, doi: 10.1142/S0219876205000569.

[31] Chong, C. S., Low, M. Y. H., Sivakumar, A. I., & Gay, K. L. "A bee colony optimization algorithm to job shop scheduling," In IEEE Proceedings of the 2006 winter simulation conference, pp. 1954-1961, Dec. 2006, doi: https://doi.org/10.1109/WSC.2006.322980.

[32] B. Z. Yao, C. Y. Yang, J. J. Hu, G. D. Yin, and B. Yu, "An improved artificial bee colony algorithm for job shop problem," Applied mechanics and materials, vol. 26–28, pp. 657–660, Jun. 2010, doi: 10.4028/www.scientific.net/AMM.26-28.657.

[33] L. Asadzadeh, "A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy," Computers & Industrial Engineering, vol. 102, pp. 359–367, Dec. 2016, doi: 10.1016/j.cie.2016.06.025.

[34] W. Wisittipanich and V. Kachitvichyanukul, "Differential evolution algorithm for job shop scheduling problem," Industrial Engineering and Management Systems, vol. 10, no. 3, pp. 203–208, Sep. 2011, doi: 10.7232/iems.2011.10.3.203.

[35] L. Tang, Y. Zhao, and J. Liu, "An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production," IEEE transactions on evolutionary computation, vol. 18, no. 2, pp. 209–225, Apr. 2014, doi: 10.1109/TEVC.2013.2250977.

[36] S. Hanoun, S. Nahavandi, D. Creighton, and H. Kull, "Solving a multiobjective job shop scheduling problem using Pareto Archived Cuckoo Search," In Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation, Kracow, Poland, IEEE, Sep. 2012, pp. 1–8. doi: https://doi.org/10.1109/ETFA.2012.6489617.

[37] A. Ouaarab, B. Ahiod, X.-S. Yang, and M. Abbad, "Discrete cuckoo search algorithm for job shop scheduling problem," in 2014 IEEE International Symposium on Intelligent Control, Juan Les Pins, France, Oct. 2014, pp. 1872–1876. doi: 10.1109/ISIC.2014.6967636.

[38] M. K. Marichelvam and T. Prabaharan, "A bat algorithm for realistic hybrid flowshop scheduling problems to minimize makespan and mean flow time," ICTACT Journal of Soft Computing, vol. 03, no. 01, pp. 428–433, Oct. 2012, doi: 10.21917/ijsc.2012.0066.

[39] T.-K. Dao, T.-S. Pan, T.-T. Nguyen, and J.-S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems," Journal of Intelligent Manufacturing, vol. 29, no. 2, pp. 451–462, Feb. 2018, doi: 10.1007/s10845-015-1121-x.

[40] Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil-López, S., & Portilla-Figueras, A., "The coral reefs optimization algorithm: an efficient meta-heuristic for solving hard optimization problems," In Proceedings of the 15th international conference on applied stochastic models and data analysis, pp. 751-758, Jun. 2013.

[41] Chun-Wei Tsai, Heng-Ci Chang, Kai-Cheng Hu, and Ming-Chao Chiang, "Parallel coral reef algorithm for solving JSP on Spark," in 2016 IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary, Oct. 2016, pp. 001872–001877. doi: 10.1109/SMC.2016.7844511.

[42] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[43] C. Lu, S. Xiao, X. Li, and L. Gao, "An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production," Advances in Engineering Software, vol. 99, pp. 161–176, Sep. 2016, doi: 10.1016/j.advengsoft.2016.06.004.

[44] A. G. Gad, "Particle swarm optimization algorithm and its applications: a systematic review," Archives of Computational Methods in Engineering, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.

[45] J.-S. Pan, P. Hu, V. Snášel, and S.-C. Chu, "A survey on binary metaheuristic algorithms and their engineering applications," Journal of Intelligent Manufacturing, vol. 56, no. 7, pp. 6101–6167, Jul. 2023, doi: 10.1007/s10462-022-10328-9.

[46] H. Zhang, B. Buchmeister, X. Li, and R. Ojstersek, 'An Efficient Metaheuristic Algorithm for Job Shop Scheduling in a Dynamic Environment', Mathematics, vol. 11, no. 10, pp. 2336, May 2023, doi: 10.3390/math11102336.

[47] Z. Liang, P. Zhong, M. Liu, C. Zhang, and Z. Zhang, 'A computational efficient optimization of flow shop scheduling problems', Scientific reports, vol. 12, no. 1, p. 845, Jan. 2022, doi: 10.1038/s41598-022-04887-8.

[48] B. Jeong, J.-H. Han, and J.-Y. Lee, 'Metaheuristics for a flow shop scheduling problem with urgent jobs and limited waiting times', Algorithms, vol. 14, no. 11, pp. 323, Nov. 2021, doi: 10.3390/a14110323.

[49] Khaje Zadeh, S., Shahverdiani, S., Daneshvar, A., & Madanchi Zaj, M., "Predicting the optimal stock portfolio approach of meta-heuristic algorithm and Markov decision process," Journal of decisions and operations research, vol. 5, no. 4, pp. 426-445, 2021, doi: https://doi.org/10.22105/dmor.2020.239616.1183.

[50] T. Q. Ngo, L. Q. Nguyen, and V. Q. Tran, "Novel hybrid machine learning models including support vector machine with meta-heuristic algorithms in predicting unconfined compressive strength of organic soils stabilised with cement and lime," International Journal of Pavement Engineering, vol. 24, no. 2, pp. 2136374, Jan. 2023, doi: 10.1080/10298436.2022.2136374.

[51] S. Chen and H. Xu, “Meta-heuristic algorithm-based human resource information management system design and development for industrial revolution 5.0,” Soft Computing, vol. 27, no. 7, pp. 4093–4105, Apr. 2023, doi: 10.1007/s00500-021-06650-z.
[52] N. Manavizadeh, H. Farrokhi-Asl, and P. Beiraghdar, “Using a metaheuristic algorithm for solving a home health care routing and scheduling problem,” Journal of project management, pp. 27–40, 2020, doi: 10.5267/j.jpm.2019.8.001.
[53] P. Sharma and U. Modani, “Trusted cluster-based communication for wireless sensor network using meta-heuristic algorithms,” Journal of tech science press, vol. 45, no. 2, pp. 1935–1951, 2022, doi: 10.32604/csse.2023.031509.
[54] H. Khajavi and A. Rastgoo, “Predicting the carbon dioxide emission caused by road transport using a Random Forest (RF) model combined by Meta-Heuristic Algorithms,” Sustainable Cities and Society, vol. 93, pp. 104503, Jun. 2023, doi: 10.1016/j.scs.2023.104503.
[55] W. Li, H. Dai, and D. Zhang, “The relationship between maximum completion time and total completion time in flowshop production”, Procedia Manufacturing, vol. 1, pp. 146–156, 2015, doi: 10.1016/j.promfg.2015.09.077.
[56] K. Li, Q. Deng, L. Zhang, Q. Fan, G. Gong, and S. Ding, “An effective MCTS-based algorithm for minimizing makespan in dynamic flexible job shop scheduling problem”, Computers & Industrial Engineering, vol. 155, pp. 107211, May 2021, doi: 10.1016/j.cie.2021.107211.
[57] M. H. Ali, A. Saif, and A. Ghasemi, “Robust job shop scheduling with condition-based maintenance and random breakdowns”, International federation of automatic contrl-PapersOnLine, vol. 55, no. 10, pp. 1225–1230, 2022, doi: 10.1016/j.ifacol.2022.09.557.
[58] C. Cebi, E. Atac, and O. K. Sahingoz, “Job shop scheduling problem and solution algorithms: a review”, in 2020 11th International Conference on Computing, Communication and Networking Technologies, Kharagpur, India, Jul. 2020, pp. 1–7. doi: 10.1109/ICCCNT49239.2020.9225581.
[59] K. Zhou, C. Tan, Y. Zhao, J. Yu, Z. Zhang, and Y. Wu, “Research on solving flexible job shop scheduling problem based on improved gwo algorithm ss-gwo”, Neural Process Lett, vol. 56, no. 1, p. 26, Feb. 2024, doi: 10.1007/s11063-024-11488-1.
[60] M. Yazdani, A. Aleti, S. M. Khalili, and F. Jolai, “Optimizing the sum of maximum earliness and tardiness of the job shop scheduling proble"’, Computers & Industrial Engineering, vol. 107, pp. 12–24, May 2017, doi: 10.1016/j.cie.2017.02.019.
[61] N. Tyagi, R. P. Tripathi, and A. B. Chandramouli, “Single machine scheduling model with total tardiness problem”, Indian Journal of science and technology, vol. 9, no. 37, Oct. 2016, doi: 10.17485/ijst/2016/v9i37/97527.
[62] L. Song, Y. Li, and J. Xu, “Dynamic job-shop scheduling based on transformer and deep reinforcement learning”, Processes, vol. 11, no. 12, pp. 3434, Dec. 2023, doi: 10.3390/pr11123434.
[63] A. Delgoshaei, M. K. A. B. M. Ariffin, and Z. B. Leman, “An effective 4–phased framework for scheduling job-shop manufacturing systems using weighted nsga-ii’, Mathematics, vol. 10, no. 23, Pp. 4607, Dec. 2022, doi: 10.3390/math10234607.
[64] F. Abderrabi et al., “Flexible job shop scheduling problem with sequence dependent setup time and job splitting: hospital catering case study”, Applied Sciences, vol. 11, no. 4, pp. 1504, Feb. 2021, doi: 10.3390/app11041504.
[65] A. Azzouz, A. Chaabani, M. Ennigrou, and L. B. Said, “Handling sequence-dependent setup time flexible job shop problem with learning and deterioration considerations using evolutionary bi-level optimization”, Applied Artificial Intelligence, vol. 34, no. 6, pp. 433–455, May 2020, doi: 10.1080/08839514.2020.1723871.
[66] F. M. Defersha and D. Rooyani, “An efficient two-stage genetic algorithm for a flexible job-shop scheduling problem with sequence dependent attached/detached setup, machine release date and lag-time”, Computers & Industrial Engineering, vol. 147, pp. 106605, Sep. 2020, doi: 10.1016/j.cie.2020.106605.
[67] S. Dauzère-Pérès, J. Ding, L. Shen, and K. Tamssaouet, “The flexible job shop scheduling problem: a review”, European Journal of Operational Research, vol. 314, no. 2, pp. 409–432, Apr. 2024, doi: 10.1016/j.ejor.2023.05.017.
[68] Z. Li and Y. Chen, “Minimizing the makespan and carbon emissions in the green flexible job shop scheduling problem with learning effects”, Scientific Reports, vol. 13, no. 1, pp. 6369, Apr. 2023, doi: 10.1038/s41598-023-33615-z.
[69] J. C. Serrano-Ruiz, J. Mula, and R. Poler, “Job shop smart manufacturing scheduling by deep reinforcement learning”, Journal of Industrial Information Integration, vol. 38, pp. 100582, Mar. 2024, doi: 10.1016/j.jii.2024.100582.
[70] L. Wang, Y. Zhao, and X. Yin, “Precast production scheduling in off-site construction: mainstream contents and optimization perspective”, Journal of Cleaner Production, vol. 405, pp. 137054, Jun. 2023, doi: 10.1016/j.jclepro.2023.137054.
[71] S. Kawaguchi and Y. Fukuyama, “Improved parallel reactive hybrid particle swarm optimization using improved neighborhood schedule generation method for the integrated framework of optimal production scheduling and operational planning of an energy plant in a factory”, Electronics & Communication in Japan, vol. 103, no. 7, pp. 37–48, Jul. 2020, doi: 10.1002/ecj.12237.
[72] C. Jun Tan, S. Hanoun, and C. Peng Lim, “A multi-objective evolutionary algorithm-based decision support system: A case study on job-shop scheduling in manufacturing”, in 2015 Annual IEEE Systems Conference Proceedings, Vancouver, Canada, Apr. 2015, pp. 170–174. doi: 10.1109/SYSCON.2015.7116747.
[73] C. J. Tan et al., “Application of an evolutionary algorithm-based ensemble model to job-shop scheduling”, J Intell Manuf, vol. 30, no. 2, pp. 879–890, Feb. 2019, doi: 10.1007/s10845-016-1291-1.
[74] D. Lee, D. Lee, and K. Kim, “Self-growth learning-based machine scheduler to minimize setup time and tardiness in OLED display semiconductor manufacturing”, Applied Soft Computing, vol. 145, pp. 110600, Sep. 2023, doi: 10.1016/j.asoc.2023.110600.
[75] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, “A survey of scheduling problems with setup times or costs”, European Journal of Operational Research, vol. 187, no. 3, pp. 985–1032, Jun. 2008, doi: 10.1016/j.ejor.2006.06.060.
[76] I. A. Chaudhry and P. R. Drake, “Minimizing flow-time variance in a single-machine system using genetic algorithms”, International Journal of advanced manufacturing technology, vol. 39, no. 3–4, pp. 355–366, Oct. 2008, doi: 10.1007/s00170-007-1221-7.
[77] K. Gowrishankar, C. Rajendran, and G. Srinivasan, “Flow shop scheduling algorithms for minimizing the completion time variance and the sum of squares of completion time deviations from a common due date”, European Journal of Operational Research, vol. 132, no. 3, pp. 643–665, Aug. 2001, doi: 10.1016/S0377-2217(00)00170-3.
[78] A. Syarif, A. Pamungkas, R. Kumar, and Mitsuo Gen, “Performance Evaluation of Various Heuristic Algorithms to Solve Job Shop Scheduling Problem (JSSP)”, International Journal of Intelligent Engineering & Systems, vol. 14, no. 2, pp. 334–343, Apr. 2021, doi: 10.22266/ijies2021.0430.30.
[79] A. Delgoshaei et al., “A supervised method for scheduling multi-objective job shop systems in the presence of market uncertainties”, Operation Research, vol. 55, pp. S1165–S1193, 2021, doi: 10.1051/ro/2020082.
[80] G. Wan and B. P.-C. Yen, “Single machine scheduling to minimize total weighted earliness subject to minimal number of tardy jobs”, European Journal of Operational Research, vol. 195, no. 1, pp. 89–97, May 2009, doi: 10.1016/j.ejor.2008.01.029.
[81] V. Poongothai, P. Godhandaraman, and A. A. Jenifer, “Single machine scheduling problem for minimizing total tardiness of a weighted jobs in a batch delivery system, stochastic rework and reprocessing times”, The 11th national conference on mathematical techniques and applications, Chennai, India, Jan. 2019, p. 020132. doi: 10.1063/1.5112317.
[82] V. Boyer, J. Vallikavungal, X. Cantú Rodríguez, and M. A. Salazar-Aguilar, “The generalized flexible job shop scheduling problem”, Computers & Industrial Engineering, vol. 160, pp. 107542, Oct. 2021, doi: 10.1016/j.cie.2021.107542.

[83] X. Zhang and S. Van De Velde, "On-line two-machine job shop scheduling with time lags", Information Processing Letters, vol. 110, no. 12–13, pp. 510–513, Jun. 2010, doi: 10.1016/j.ipl.2010.04.002.

[84] Y. He, F. Liu, H. Cao, and C. Li, "A bi-objective model for job-shop scheduling problem to minimize both energy consumption and makespan", Journal of Central South University of Technology, vol. 12, no. 2, pp. 167–171, Oct. 2005, doi: 10.1007/s11771-005-0033-x.

[85] Z. Wei, W. Liao, and L. Zhang, "Hybrid energy-efficient scheduling measures for flexible job-shop problem with variable machining speeds", Expert Systems with Applications, vol. 197, pp. 116785, Jul. 2022, doi: 10.1016/j.eswa.2022.116785.

[86] C. Peng, Z. Li, H. Zhong, X. Li, A. Lin, and Y. Liao, "Research on Multi-Objective Scheduling Algorithm of Job Shop Considering Limited Storage and Transportation Capacity", IEEE IEEE internet of things journal, vol. 11, pp. 94252–94280, 2023, doi: 10.1109/ACCESS.2023.3285710.

[87] H. Hu, J. He, X. He, W. Yang, J. Nie, and B. Ran, "Emergency material scheduling optimization model and algorithms: A review", Journal of Traffic and Transportation Engineering, vol. 6, no. 5, pp. 441–454, Oct. 2019, doi: 10.1016/j.jtte.2019.07.001.

[88] P. Schworm, X. Wu, M. Glatt, and J. C. Aurich, "Solving flexible job shop scheduling problems in manufacturing with Quantum Annealing", Production engineering , vol. 17, no. 1, pp. 105–115, Feb. 2023, doi: 10.1007/s11740-022-01145-8.

[89] A. S. Jain and S. Meeran, "Deterministic job-shop scheduling: Past, present and future", European Journal of Operational Research, vol. 113, no. 2, pp. 390–434, Mar. 1999, doi: 10.1016/S0377-2217(98)00113-1.

[90] M. M. Ahmadian, M. Khatami, A. Salehipour, and T. C. E. Cheng, "Four decades of research on the open-shop scheduling problem to minimize the makespan," European Journal of Operational Research, vol. 295, no. 2, pp. 399–426, Dec. 2021, doi: 10.1016/j.ejor.2021.03.026.

[91] K. Tamssaouet, S. Dauzère-Pérès, and C. Yugma, "Metaheuristics for the job-shop scheduling problem with machine availability constraints", Computers & Industrial Engineering, vol. 125, pp. 1–8, Nov. 2018, doi: 10.1016/j.cie.2018.08.008.

[92] A. Agnetis, J.-C. Billaut, S. Gawiejnowicz, D. Pacciarelli, and A. Soukhal, "Scheduling problems with variable job processing times", in Multiagent Scheduling, Springer, Berlin Heidelberg, 2014, pp. 217–260. doi: 10.1007/978-3-642-41880-8_6.

[93] Z. Zhu and X. Zhou, "Flexible job-shop scheduling problem with job precedence constraints and interval grey processing time," Computers & Industrial Engineering, vol. 149, pp. 106781, Nov. 2020, doi: 10.1016/j.cie.2020.106781.

[94] S. Zhang, H. Du, S. Borucki, S. Jin, T. Hou, and Z. Li, "Dual Resource Constrained Flexible Job Shop Scheduling Based on Improved Quantum Genetic Algorithm," Machines, vol. 9, no. 6, pp. 108, May 2021, doi: 10.3390/machines9060108.

[95] G. Mauroy, Y. Wardi, and J. M. Proth, "Perturbation analysis of job shop scheduling with multiple job priorities," in Proceedings of the 1998 IEEE International Conference on Control Applications, Trieste, Italy, 1998, pp. 930–931. doi: 10.1109/CCA.1998.721594.

[96] P. Schworm, X. Wu, M. Klar, M. Glatt, and J. C. Aurich, "Multi-objective quantum annealing approach for solving flexible job shop scheduling in manufacturing", Journal of Manufacturing Systems, vol. 72, pp. 142–153, Feb. 2024, doi: 10.1016/j.jmsy.2023.11.015.

[97] Ding Zhu, Wu Ping, Zhang Libo, Wang Feng, and Zhang Xuefeng, "A multi-agent ant colony optimization algorithm for earliness/tardiness scheduling with different due window on non-uniform parallel machines," in International Technology and Innovation Conference 2006, Hangzhou, China: IEEE, pp. 67–71, 2006, doi: 10.1049/cp:20060730.

[98] F. Dugardin, H. Chehade, L. Amodeo, F. Yalaoui, and C. Prins, "Hybrid job shop and parallel machine scheduling problems: minimization of total tardiness criterion," in Multiprocessor Scheduling, Theory and Applications, E. Levner, Ed., I-Tech Education and Publishing, 2007. doi: 10.5772/5227.

[99] R. L. C. Souza, A. Ghasemi, A. Saif, and A. Gharaei, "Robust job-shop scheduling under deterministic and stochastic unavailability constraints due to preventive and corrective maintenance," Computers & Industrial Engineering, vol. 168, pp. 108130, Jun. 2022, doi: 10.1016/j.cie.2022.108130.

[100] P. Sharma and A. Jain, "A review on job shop scheduling with setup times," Proceedings of the institution of mechanical engineers, part b: journal of engineering manufacture, vol. 230, no. 3, pp. 517–533, Mar. 2016, doi: 10.1177/0954405414560617.

[101] M. Vázquez and D. Whitley, "A comparison of genetic algorithms for the static job shop scheduling problem", in Parallel Problem Solving from Nature PPSN VI, vol. 1917, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 303–312. doi: 10.1007/3-540-45356-3_30.

[102] Mohan, K. Lanka, and A. N. Rao, "A Review of Dynamic Job Shop Scheduling Techniques," Procedia Manufacturing, vol. 30, pp. 34–39, 2019, doi: 10.1016/j.promfg.2019.02.006.

[103] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," Multimedia Tools & Application, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[104] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated Annealing: From Basics to Applications," in Handbook of Metaheuristics, vol. 272, pp. 1-35 M. Gendreau and J.-Y. Potvin, Eds., Cham: Springer International Publishing, 2019, doi: 10.1007/978-3-319-91086-4_1.

[105] Z. Lu, A. Martínez-Gavara, J.-K. Hao, and X. Lai, "Solution-based tabu search for the capacitated dispersion problem," Expert Systems with Applications, vol. 223, p. 119-156, Aug. 2023, doi: 10.1016/j.eswa.2023.119856.

[106] R. Martí, A. Martínez-Gavara, and F. Glover, "Tabu Search," in Discrete Diversity and Dispersion Maximization: A Tutorial on Metaheuristic Optimization, R. Martí and A. Martínez-Gavara, Eds., Cham: Springer International Publishing, 2023, pp. 137–149. doi: 10.1007/978-3-031-38310-6_7.

[107] A. Soofastaei, "Introductory Chapter: Ant Colony Optimization," in the Application of Ant Colony Optimization, A. Soofastaei, Ed., IntechOpen, 2022. doi: 10.5772/intechopen.103801.

[108] J. Yi, C. Lu, G. Li, J. Yi, C. Lu, and G. Li, 'A literature review on latest developments of Harmony Search and its applications to intelligent manufacturing', Multimedia tools & application, vol. 16, no. 4, pp. 2086–2117, 2019, doi: 10.3934/mbe.2019102.

[109] C. Guo, H. Tang, B. Niu, and C. Boon Patrick Lee, 'A survey of bacterial foraging optimization', Neurocomputing, vol. 452, pp. 728–746, Sep. 2021, doi: 10.1016/j.neucom.2020.06.142.

[110] C. Wang, S. Zhang, T. Ma, Y. Xiao, M. Z. Chen, and L. Wang, "Swarm intelligence: a survey of model classification and applications," Chinese Journal of Aeronautics, pp. S1000936124000931, Mar. 2024, doi: 10.1016/j.cja.2024.03.019.

[111] A. Kumar, D. Kumar, and S. K. Jarial, "A review on artificial bee colony algorithms and their applications to data clustering," Cybernetics and Information Technologies, vol. 17, no. 3, pp. 3–28, Sep. 2017, doi: 10.1515/cait-2017-0027.

[112] X. Li, J. Lu, C. Yang, and J. Wang, "Research of flexible assembly job-shop batch–scheduling problem based on improved artificial bee colony," Frontiers in bioengineering & biotechnology, vol. 10, pp. 909548, Aug. 2022, doi: 10.3389/fbioe.2022.909548.

[113] A. Priya, S. Mandal, and Yogesh, "Parallel artificial bee colony algorithm for solving advance industrial productivity problems," in Advances in Library and Information Science, B. Holland and K. Sinha, Eds., IGI Global, 2024, pp. 21–41. doi: 10.4018/979-8-3693-0807-3.ch002.

[114] L. Asadzadeh, "A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy", Computers & Industrial Engineering, vol. 102, pp. 359–367, Dec. 2016, doi: 10.1016/j.cie.2016.06.025.

[115] S. K. Mishra, P. S. C. Bose, and C. S. P. Rao, "An invasive weed optimization approach for job shop scheduling problems,"

International journal of advanced manufacturing & technology, vol. 91, no. 9–12, pp. 4233–4241, Aug. 2017, doi: 10.1007/s00170-017-0091-x.

[116] P. Shilpa and G. Balaraju, "Job shop scheduling using differential evolution algorithm", International journal of mechanical and production engineering research and development, vol. 8, no. 3, pp. 327–338, 2018, doi: 10.24247/ijmperdjun201837.

[117] P. Sriboonchandr, N. Kriengkorakot, and P. Kriengkorakot, "Improved differential evolution algorithm for flexible job shop scheduling problems", Mathematical computation application, vol. 24, no. 3, p. 80, Sep. 2019, doi: 10.3390/mca24030080.

[118] S. Singh and K. P. Singh, "Cuckoo search optimization for job shop scheduling problem," in Proceedings of Fourth International Conference on Soft Computing for Problem Solving, vol. 335, K. N. Das, K. Deep, M. Pant, J. C. Bansal, and A. Nagar, Eds., New Delhi: Springer India, 2015, pp. 99–111. doi: 10.1007/978-81-322-2217-0_9.

[119] A. Ouaarab, B. Ahiod, and X.-S. Yang, "Discrete cuckoo search applied to job shop scheduling problem," in Recent Advances in Swarm Intelligence and Evolutionary Computation, vol. 585, X.-S. Yang, Ed., Cham: Springer International Publishing, 2015, pp. 121–137. doi: 10.1007/978-3-319-13826-8_7.

[120] S. Batubara, D. K. Sari, and D. A. Wicaksono, "Design of job scheduling using bat algorithm to minimize makespan in hybrid flowshop," In Proceedings of the International Conference on Industrial Engineering and Operations Management Dubai, UAE, March 10-12, 2020, pp. 2606-2617. Available at https://www.ieomsociety.org/ieom2020/papers/618.pdf.

[121] T.-K. Dao, T.-S. Pan, T.-T. Nguyen, and J.-S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems," Journal of intelligent manufacturing, vol. 29, no. 2, pp. 451–462, Feb. 2018, doi: 10.1007/s10845-015-1121-x.

[122] Chun-Wei Tsai, Heng-Ci Chang, Kai-Cheng Hu, and Ming-Chao Chiang, "Parallel coral reef algorithm for solving JSP on Spark," in 2016 IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary, Oct. 2016, pp. 001872–001877. doi: 10.1109/SMC.2016.7844511.

[123] T. Jiang and C. Zhang, "Application of grey wolf optimization for solving combinatorial problems: job shop and flexible job shop scheduling cases", IEEE Access, vol. 6, pp. 26231–26240, 2018, doi: 10.1109/ACCESS.2018.2833552.

[124] C. Zhang, K. Wang, Q. Ma, X. Li, and L. Gao, "A discrete grey wolf optimizer for solving flexible job shop scheduling problem with lot-streaming," in 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China: IEEE, May 2021, pp. 969–974. doi: 10.1109/CSCWD49262.2021.9437886.

[125] M. S. Viana, O. Morandin Junior, and R. C. Contreras, "A modified genetic algorithm with local search strategies and multi-crossover operator for job shop scheduling problem," Sensors, vol. 20, no. 18, p. 5440, Sep. 2020, doi: 10.3390/s20185440.

[126] Y. Song, F. Wang, and X. Chen, "An improved genetic algorithm for numerical function optimization," Applied Intelligence, vol. 49, no. 5, pp. 1880–1902, May 2019, doi: 10.1007/s10489-018-1370-4.

[127] I. Chaouch, O. B. Driss, and K. Ghedira, "A modified ant colony optimization algorithm for the distributed job shop scheduling problem," Procedia Computer Science, vol. 112, pp. 296–305, 2017, doi: 10.1016/j.procs.2017.08.267.

[128] W. Teekeng, A. Thammano, P. Unkaw, and J. Kiatwuthiamorn, "A new algorithm for flexible job-shop scheduling problem based on particle swarm optimization," Artificial Life & Robotics, vol. 21, no. 1, pp. 18–23, Mar. 2016, doi: 10.1007/s10015-015-0259-0.

[129] B. Denkena, F. Schinkel, J. Pirnay, and S. Wilmsmeier, "Quantum algorithms for process parallel flexible job shop scheduling," CIRP Journal of Manufacturing Science and Technology, vol. 33, pp. 100–114, May 2021, doi: 10.1016/j.cirpj.2021.03.006.

[130] Z. Yongsuo, Z. Caimeng, H. Yinsheng, Q. Pengrui, and S. Wangjiao, "Research on JOB SHOP Scheduling Problem," Journal of Physics.: Conference Series, vol. 1607, no. 1, p. 012050, Aug. 2020, doi: 10.1088/1742-6596/1607/1/012050.

[131] Siregar, I. Rizkya, R. M. Sari, and K. Syahputri, "Minimization of makespan using fcfs method and genetic algorithm method comparison in aluminum industry," in 2019 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM), Medan, Indonesia: IEEE, pp. 9–12, sept. 2019, doi: 10.1109/ELTICOM47379.2019.8943914.

[132] S.-J. Wang, C.-W. Tsai, and M.-C. Chiang, "A high performance search algorithm for job-shop scheduling problem," Procedia Computer Science, vol. 141, pp. 119–126, 2018, doi: 10.1016/j.procs.2018.10.157.

[133] R. Hu, X. Wu, B. Qian, J. Mao, and H. Jin, "Differential evolution algorithm combined with uncertainty handling techniques for stochastic reentrant job shop scheduling problem," Complexity, vol. 2022, pp. 1–11, Jan. 2022, doi: 10.1155/2022/9924163.

[134] T. H. Son, H. X. Long, N. Huynh-Tuong, and T. Van Lang, "An approach for the teamwork scheduling problem with job-person constraint," in 2022 9th NAFOSTED Conference on Information and Computer Science, Ho Chi Minh City, Vietnam, pp. 64–68, Oct. 2022 doi: 10.1109/NICS56915.2022.10013369.

[135] P. Liu, K. Xu, H. Gong, "Non-cooperative game of coordinated scheduling of parallel machine production and Transportation in shared manufacturing," Computers, materials & continua, Vol. 76, no.1, pp. 239-258, 2023, doi: 10.32604/cmc.2023.038232.

[136] A. Campo, J. A. Cano, R. Gómez-Montoya, E. Rodríguez-Velásquez, and P. Cortés, "Flexible job shop scheduling problem with fuzzy times and due-windows: minimizing weighted tardiness and earliness using genetic algorithms," Algorithms, vol. 15, no. 10, pp. 334, Sep. 2022, doi: 10.3390/a15100334.

[137] H. Zhang, B. Buchmeister, X. Li, and R. Ojstersek, "Advanced metaheuristic method for decision-making in a dynamic job shop scheduling environment," Mathematics, vol. 9, no. 8, pp. 909, Apr. 2021, doi: 10.3390/math9080909.

[138] N. Kumar and A. Mishra, "Comparative study of different heuristics algorithms in solving classical job shop scheduling problem," Materials Today: Proceedings, vol. 22, pp. 1796–1802, 2020, doi: 10.1016/j.matpr.2020.03.013.

[139] L. Wang, G. Zhou, Y. Xu, S. Wang, and M. Liu, "An effective artificial bee colony algorithm for the flexible job-shop scheduling problem," International journal of advanced manufacturing & technology, vol. 60, no. 1–4, pp. 303–315, Apr. 2012, doi: 10.1007/s00170-011-3610-1.

[140] R. Zarrouk, I. E. Bennour, and A. Jemai, "A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem," Swarm Intelligence, vol. 13, no. 2, pp. 145–168, Jun. 2019, doi: 10.1007/s11721-019-00167-w.

[141] Hajibabaei and J. Behnamian, "Flexible job-shop scheduling problem with unrelated parallel machines and resources-dependent processing times: a tabu search algorithm," International Journal of Management Science and Engineering Management, vol. 16, no. 4, pp. 242–253, Oct. 2021, doi: 10.1080/17509653.2021.1941368.

[142] A. Dabah, A. Bendjoudi, A. AitZai, and N. N. Taboudjemat, "Efficient parallel tabu search for the blocking job shop scheduling problem," Soft Computing, vol. 23, no. 24, pp. 13283–13295, Dec. 2019, doi: 10.1007/s00500-019-03871-1.

[143] Alharkan, M. Saleh, M. A. Ghaleb, H. Kaid, A. Farhan, and A. Almarfadi, "Tabu search and particle swarm optimization algorithms for two identical parallel machines scheduling problem with a single server," Journal of King Saud University - Engineering Sciences, vol. 32, no. 5, pp. 330–338, Jul. 2020, doi: 10.1016/j.jksues.2019.03.006.

[144] Y. Yu, "A research review on job shop scheduling problem," E3S Web Conf., vol. 253, p. 02024, 2021, doi: 10.1051/e3sconf/202125302024.

[145] Hasani, S. A. Kravchenko, and F. Werner, "Minimizing the makespan for the two-machine scheduling problem with a single server: Two algorithms for very large instances," Engineering Optimization, vol. 48, no. 1, pp. 173–183, Jan. 2016, doi: 10.1080/0305215X.2015.1005083.

[146] Hasani, S. A. Kravchenko, and F. Werner, "Simulated annealing and genetic algorithms for the two-machine scheduling problem

with a single server," International Journal of Production Research, vol. 52, no. 13, pp. 3778–3792, Jul. 2014, doi: 10.1080/00207543.2013.874607.

[147] F. M. Defersha and M. Chen, "A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups," International journal of advanced manufacturing & technology, vol. 49, no. 1–4, pp. 263–279, Jul. 2010, doi: 10.1007/s00170-009-2388-x.

[148] W. N. Abdullah and S. A. Alagha, "A parallel adaptive genetic algorithm for job shop scheduling problem," Journal of physics: Conference Series, vol. 1879, no. 2, p. 022078, May 2021, doi: 10.1088/1742-6596/1879/2/022078.

[149] Z. Liu et al., "An improved genetic algorithm with an overlapping strategy for solving a combination of order batching and flexible job shop scheduling problem," Engineering Applications of Artificial Intelligence, vol. 127, p. 107321, Jan. 2024, doi: 10.1016/j.engappai.2023.107321.

[150] Kumar, P., Ghangas, G., Sharma, A., & Dhull, S., "Minimising the makespan of job shop scheduling problem by using genetic algorithm (GA)," International journal of production engineering, vol. 6, pp. 27-39, 2020.

[151] S. Habbadi, B. Herrou, and S. Sekkat, "Job shop scheduling problem using genetic algorithms," in Proceedings of the International Conference on Industrial Engineering and Operations Management, Rome, Europe: IEOM Society International, Jul. 2022, pp. 3050–3062. doi: 10.46254/EU05.20220592.

[152] Abdullah, W. N., "Solving job-shop scheduling problem using a developed particle swarm optimization algorithm," International Journal of Science and Research, vol. 7 no.1, pp. 1845-1848, 2018. doi: 10.21275/ART20179752.

[153] Ben Ali, A. J. Telmoudi, and S. Gattoufi, "Stochastic cases of the dynamic job shop problem based on the genetic algorithm to minimize," in 2017 4th International Conference on Control, Decision and Information Technologies, Barcelona, pp. 0760–0765, Apr. 2017, doi: 10.1109/CoDIT.2017.8102686.

[154] Z. Shen and L. Smalov, "Comparative performance of genetic algorithm, simulated annealing and ant colony optimisation in solving the job-shop scheduling problem," in 2018 26th International Conference on Systems Engineering, Sydney, Australia, pp. 1–5, Dec. 2018, doi: 10.1109/ICSENG.2018.8638185.

[155] Janes, G., Perinic, M., & Jurkovic, Z., "Applying improved genetic algorithm for solving job shop scheduling problems," Technical gazatte, vol. 24, no. 4, pp. 1243-1247, Aug. 2017, doi: 10.17559/TV-20150527133957.

[156] A. Salido, J. Escamilla, A. Giret, and F. Barber, "A genetic algorithm for energy-efficiency in job-shop scheduling," International journal of advanced manufacturing & technology, vol. 85, no. 5–8, pp. 1303–1314, Jul. 2016, doi: 10.1007/s00170-015-7987-0.

[157] Zhang, Y. Hu, J. Sun, and W. Zhang, "An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints," Swarm and Evolutionary Computation, vol. 54, pp. 100664, May 2020, doi: 10.1016/j.swevo.2020.100664.

[158] Ç. Sel and A. Hamzadayı, "A simulated annealing approach based simulation-optimisation to the dynamic job-shop scheduling problem," Pamukkale journal of engineering & science, vol. 24, no. 4, pp. 665–674, 2018, doi: 10.5505/pajes.2017.47108.

[159] W. Zhang, J. B. Wen, Y. C. Zhu, and Y. Hu, "Multi-objective scheduling simulation of flexible job-shop based on multi-population genetic algorithm," International journal of simulation model, vol. 16, no. 2, pp. 313–321, Jun. 2017, doi: 10.2507/IJSIMM16(2)CO6.

[160] Zhang, B. Buchmeister, X. Li, and R. Ojstersek, "Advanced metaheuristic method for decision-making in a dynamic job shop scheduling environment," Mathematics, vol. 9, no. 8, pp. 909, Apr. 2021, doi: 10.3390/math9080909.

[161] R. Singh and S. S. Mahapatra, "A quantum behaved particle swarm optimization for flexible job shop scheduling," Computers & Industrial Engineering, vol. 93, pp. 36–44, Mar. 2016, doi: 10.1016/j.cie.2015.12.004.

[162] Z. Wang, J. Zhang, and S. Yang, "An improved particle swarm optimization algorithm for dynamic job shop scheduling problems with random job arrivals," Swarm and Evolutionary Computation, vol. 51, pp. 100594, Dec. 2019, doi: 10.1016/j.swevo.2019.100594.

[163] C. Lin, Z. Cao, and M. Zhou, "Learning-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility," IEEE Transactions on Cybernetics, vol. 53, no. 10, pp. 6663–6675, Oct. 2023, doi: 10.1109/TCYB.2022.3210228.

[164] A. Y. Zebari, S. M. Almufti, and C. M. Abdulrahman, "Bat algorithm (BA): review, applications and modifications," International Journal of Scientific World, vol. 8, no. 1, pp. 1–7, Jan. 2020, doi: 10.14419/ijsw.v8i1.30120.