Volume 8 Issue 1 - January 2024 - pp. 45-63 https://doi.org/10.32734/jocai.v8.i1-17138

Contents lists available online at TALENTA Publisher

DATA SCIENCE: JOURNAL OF COMPUTING AND APPLIED INFORMATICS (JoCAI)

Journal homepage: https://jocai.usu.ac.id

A Review on Metaheuristic Approaches for Job-Shop Scheduling Problems

Majid Abdolrazzagh-Nezhad[1*] and Salwani Abdullah[2]

[1]*Department of Computer Science, Faculty of Computer & Industrial Engineering, Birjand University of Technology, Birjand, Iran*
[2]*Centre for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia*

email : [1]abdolrazzagh@birjandut.ac.ir , [2]salwani@ukm.edu.my

ARTICLE INFO

Article history:
Received 13 November 2023
Revised 29 Desember 2023
Accepted 31 2023
Published online 31 July 2023

Keywords:
Heuristics
Job-shop
Metaheuristics
Scheduling
Review

Corresponding Author:
abdolrazzagh@buqaen.ac.ir

ABSTRACT

Over the past several decades, interest in metaheuristic approaches to address job-shop scheduling problems (JSSPs) has increased due to the ability of these approaches to generate solutions which are better than those generated from heuristics alone. This article provides a significant attention on reviewing state-of-the-art metaheuristic approaches that have been developed to solve JSSPs. These approaches are analysed with respect to three steps: (i) preprocessing, (ii) initialization procedures and (iii) improvement algorithms. Through this review, the paper highlights the gaps in the literature and potential avenues for further research.

IEEE style in citing this article:
M. Abdolrazzagh-Nezhad and S. Abdullah " A Review on Metaheuristic Approaches for Job-Shop Scheduling Problems," DATA SCIENCE: JOURNAL OF COMPUTING AND APPLIED INFORMATICS (JoCAI), vol. 8, no. 1, pp. 45-63. 2024.

1. Introduction

The ability to make timely, effective decisions is one of the most important issues faced in manufacturing; slow or poor decisions increase production costs. Thus, good and timely decisions based on optimal scheduling of the production process with regards to limited available resources constitute a key factor in the efficient control of production. The job-shop scheduling problem (JSSP) is an important decision problem facing those involved in the fields of industry, economics and management. This problem is a class of combinational optimization problem known as the NP-hard problem [1]. From the mid-fifties onwards, many researchers have been interested in expanding the theoretical models of the JSSP and have introduced algorithms to solve them. Among these, some have tried to review, categorize and analyse the various methodologies applied to the JSSP. [2] is

e- ISSN: 2580-829X | p- ISSN: 2580-6769

the fundamental work that first reviewed the proposed methodologies for the JSSP. Later, [3] revisited the JSSP as one of the production scheduling problems. Nearly two decades later, exact methods and hybrid techniques, as iterated local search algorithms, were reviewed by [4]. They developed guidelines on the features that should be included to make a suitable job-shop scheduling system. Thereafter, several other researchers reviewed and analysed available methodologies for the JSSP [5]-[9]. Following on from the work of such authors, this paper aims to provide a review of the literature on JSSPs, paying particular attention to metaheuristic approaches due to the ability of these approaches to generate solutions for the JSSP which are better than those generated from heuristics alone.

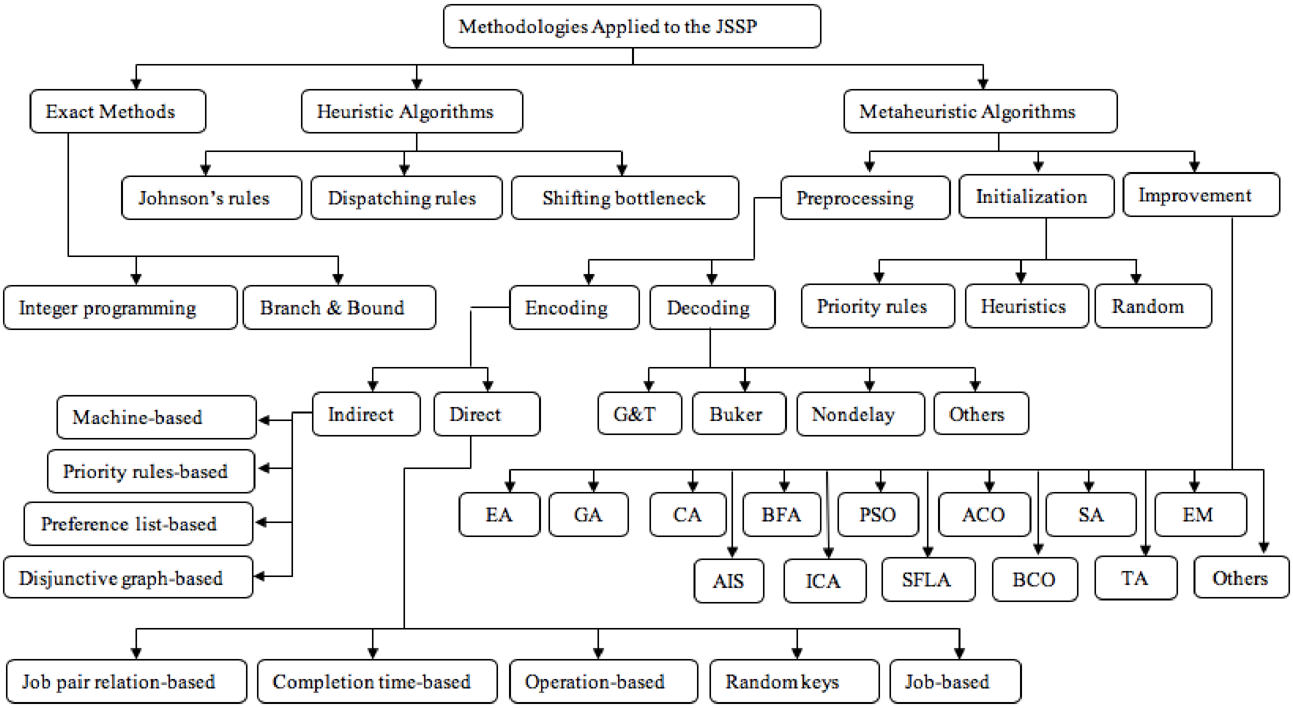
The remain of the paper is organized as follows: Section 2 presents a discussion of the approaches applied to JSSPs and is followed by section 3 which consists of a summary highlighting the effective features that could be considered in future works.

2. Methodologies Applied To The JSSP

Research on JSSP solutions began in the mid-fifties with the proposing of some exact methods. Thereafter, researchers focused on heuristic and metaheuristic algorithms. The purpose of this section is to provide an overview of these exact methods (section 2.1) and the heuristic algorithms (section 2.2) and also to review the latest metaheuristic trends (section 2.3) for solving the crisp JSSP, where there has been a significant tendency towards pre-processing, initialization and improvement procedures. To achieve this purpose, 135 research studies on JSSP were investigated. These studies were the most cited in the literature and were all published in renowned academic journal hence they are considered a reliable statistical population upon which to base an informed assessment of state-of-the-art metaheuristics. The methodologies applied to the JSSP which are discussed below are presented in Figure 1.

2.1 Exact Methods

Exact algorithms are guaranteed to find every small-size instance of a JSSP in bounded time. Unfortunately, for the real large-scale application of JSSPs, which are NP-hard, no algorithms exist that can solve them in polynomial time. Therefore, exact algorithms need exponential computation time in most cases, which leads to an impractical computational burden for these problems. The family of exact methods is extremely large, but the most common exact methods for crisp JSSPs are branch and bound algorithms and mixed integer programming. The most effective exact methods to solve the crisp JSSP are branch and bound algorithms [10]-[13]. They have been designed based on exploring specific information about the critical path of the problem. Although these algorithms guarantee the finding of an optimal solution for a JSSP, they are very slow in optimizing the problem. Their main disadvantage is the lack of effective lower bounds to cut the branches of the tree as early as possible and reduce the computational time required. Some researchers have tried to improve the lower bounds [13]-[16]. It should also be mentioned here that the combination of branch and bound techniques with metaheuristic strategies to develop advanced hybrid algorithms is an emerging area of study in this respect [17],[18].



Note: EA: evolutionary algorithm; GA: genetic algorithm; MA: memetic algorithm; CA: cultural algorithm; AIS: artificial immune system; BFA: bacterial foraging algorithm; ICA: imperialist competitive algorithm; PSO: particle swarm optimization; SFLA: shuffled frog-leaping algorithm; BCO: bee colony optimization; ACO: ant colony optimization; TS: tabu search; SA: simulated annealing; EM: electromagnetic-like mechanism.

FIGURE 1. Methodologies Applied to the JSSP

With regards to the formulation of integer programming [19], one of the earliest works that proposed an exact method using this approach was that of [20]. A very efficient method was proposed by [23], which has remained the best performing exact method for several years.

2.2 Heuristic Algorithm

As mentioned above, the implementation of exact methods to solve JSSPs often leads to unreasonable computational times. This disadvantage led most researchers to consider approximation methods for JSSPs. These methods are able to get near-optimum solutions in reasonable computational times, although approximation methods cannot guarantee the finding of optimal solutions. Approximation algorithms are classified into heuristic [24] and metaheuristic [25] algorithms. The term 'heuristic' derives from the Greek verb *heuriskein* which means 'to find' and the suffix *meta* which means 'beyond an upper level'. A heuristic to solve the JSSP was first proposed by [26] who generalized Johnson's rule [27] for flow-shop scheduling problems and applied it to the two-machine JSSP. Thereafter, a large number of heuristic methods w ere proposed based on dispatching rules [28]-[38]. A dispatching rule is a rule that prioritizes those unscheduled jobs with the highest priority to be processed on a machine. The G&T algorithm [32] is one of the most important and earliest executions of priority rules, and is considered a constructive algorithm. The most well-known dispatching priority rules that have been applied to the JSSP are: Shortest Processing Time first (**SPT**), Shortest Remaining Time first (**SRT**), Longest Processing Time first (**LPT**), Earliest Completion Time first (**ECT**), Weighted Shortest Processing Time first (**WSPT**), Earliest Release Date first (**ERD**) and Earliest Due Date first (**Edd).** Although dispatching rules cannot perform effectively in the case of complex JSSPs and have limited use in solving practical JSSPs, these rules have three advantages: they are very simple to implement, they offer fast computation time and they have the ability to find a reasonable, good solution in a relatively short time for simple JSSPs.

The shifting bottleneck heuristic is one of the most powerful heuristics that has been developed to solve the JSSP, and was originally proposed by [39] and later improved [40]-[43]. This algorithm includes optimizing a one-machine scheduling subproblem for each machine (work-centre). This heuristic is also widely used nowadays in efficient hybrid algorithm implementations [44], [45].

The effectiveness of this heuristic relies on its ability to efficiently obtain good schedules for individual work-centres and to model the interactions between work-centres accurately. The shifting bottleneck heuristic has some advantages over exact methods and dispatching rules. First, it decomposes the shop into a number of disjointed work-centres, which allows each work-centre to be scheduled using the most suitable procedure. Second, it uses global information, in contrast to dispatching rules which uses local information. However, it uses the makespan scheduling objective, which is not practical, and it does not consider sequence-dependent setups [46].

2.3 Metaheuristic Algorithms

The idea of metaheuristic algorithms was first proposed by [25]. Thereafter, several definitions were presented for these algorithms including that of Osman and Laport [47], which states: "*A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space and learning strategies are used to structure information in order to find efficiently near-optimal solutions*" [49]. It could be said that metaheuristics are an intelligent way to explore the solution space and exploit the optimal solution. Meta-Heuristic algorithms have played a significant role in solving FJSSPs. Most recent literature review shows that Meta-Heuristic algorithms receive major attention than other techniques in the fuzzy scheduling environment with more than 70% of the existing studies. To implement one metaheuristic for a JSSP it is necessary to implement three steps: **preprocessing**, **initialization** and **improvement**.

Preprocessing consists of two tasks: representing or encoding the solution space and designing a decoder algorithm to generate the active schedule corresponding to the encoded point. Next, an initial population or initial single point has to be generated in initialization. Finally, the initial solutions are improved by intelligent algorithms based on exploration and exploitation techniques applied to the solution space of the JSSP.

2.3.1 Preprocessing

One of the key issues that needs to be addressed when applying metaheuristics successfully to the JSSP is how to encode a schedule for the search space, i.e., the suitable selection of an **encoding scheme** is crucial in enhancing the search effectiveness of any metaheuristic. To this end, over the last 25 years, nine representations for the JSSP have been proposed. The definitions and properties of the representation schemes of the JSSP are described in [48]. Representations such as job-based, operation-based, job pair relation-based, completion time-based and random keys belong to this category. In the indirect approach, a sequence of dispatching rules for a job assignment, but not a schedule, is encoded into a search space and metaheuristic algorithms are used to evolve those points to find a better sequence of dispatching rules. A schedule is then constructed with that sequence of dispatching rules. Preference list-based, priority rule-based, disjunctive graph-based and machine-based representations belong to this category. The percentage breakdown of the representations applied to the JSSP is illustrated in Figure 2.

All the encoding schemes that have been proposed to solve the JSSP are able to generate an active schedule by using a decoder. [48] classify the degree of complexity of the decoder into four levels: **Level 0**: no decoder; **Level 1**: simple mapping relation; **Level 2**: simple heuristic; and **Level 3**: complex heuristic.

Operation-based representation [49] encodes a schedule into a sequence of operations, although all possible permutations of the sequence cannot define a feasible schedule due to the existence of precedence constraints. [50] introduced an alternative; they denoted all operations for a job with the same character and then interpreted it based on the order of appearance in the sequence. The application of this representation has been found to perform the best when compared to other operation-based representations [51]-[72]. Also, the decoder of this representation consists of two main steps; first it translates the encoded point to a list of ordered operations, second, it generates the schedule by using a one-pass heuristic based on the list. The G&T algorithm [32] is a well-known constructive procedure which has been used as a decoder for this representation [51], [53], [54], [56], [60], [66], [73]. Some researchers have extended the G&T algorithm into a modified decoder [63], [65]. Other decoders for this representation have also been proposed [71], [74]-[76].

Job-based representation [77]-[79] includes a list of *n* jobs (the sequence of jobs). So, to construct a schedule according to this encoding scheme all operations of the first job in the list are scheduled first and then the operations of the second job in the list are considered and so on until all the operations of the rest of the jobs are scheduled

Preference list-based representation [80]-[98] divide an encoded point into *m* subpoints so that each one belongs to one machine. Each subpoint consists of a string of operations which has to be processed on the related machine. Subpoints do not explain the operation sequence on the machine. They are preference lists, so that in effect each machine has its own preference list. Although [66] proposed a heuristic to decode this representation into a non-delay schedule, a G&T algorithm has been the regular decoder used by various researchers. [81, [91]-[93], [95]-[97].

Job pair relation-based representation. Contains a binary matrix which determines the precedence relation of a pair of jobs in corresponding machines [70], [99]-[101].

Priority rule-based representation [102]-[105] includes a sequence of dispatching rules for job assignment and its decoder should construct a schedule by using a priority dispatching heuristic. To decode this representation, [105] proposed a simple schedule builder that sorts operations based on their relevant dispatching rules. Note that algorithms based on the G&T algorithm [30] can be considered to be the common basis of all priority rule-based heuristics.

Disjunctive graph-based representation [44], [105] -[124] is viewed as a kind of job pair relation-based representation that uses a visual style. The disjunctive graph $G=(N, A, E)$ is defined as follows: N includes nodes representing all operations, A consists of arcs connecting consecutive operations of the same jobs and E contains disjunctive arcs connecting operations to be processed by the same machine [62]. A decoder for this representation should translate the orientations of all the disjunctive arcs into sequences of operations on the same machines and then construct the related schedule.

Completion time-based representation [124], [125] encodes a sorted list of completion times of operations and does not have a decoder. Although this representation is not suitable for most metaheuristics, some researchers [104],[124], [126]-[135] have applied this scheme to their algorithm by designing a special operator to avoid illegal schedules.

Machine-based representation [104], [130]-[135] consists of a sequence of machines, and its decoder works by applying a shifting bottleneck heuristic to the sequence. Also, parallel job and parallel machine representations of the chromosome have been proposed for genetic encodings of the flexible JSSP [131].

Random key representation [101], [121]-[131] encodes a solution of the JSSP with a random number. Each encoded point consists of two parts: an integer part in set *{1,2,…,m}* and a fraction part generated randomly from *(0,1)*. The integer part is interpreted as the machine assignment for that job and the sorting of the fractional parts provides the job sequence on each machine. Some researchers [127]-[131], have considered a sequence of all operations, i.e., an operation-based representation, as an integer part of their random key representation.

2.3.2 Initialization

Initialization algorithms are typically the fastest approximation methods, yet they often return solutions of inferior quality when compared to improvement algorithms. In the literature, initialization of the JSSP has been carried out using various methods such as random methods, priority rules and heuristic algorithms. The high quality of the initial population speeds up metaheuristic algorithms [54], [60] although producing an initial population has often attracted less attention than other steps of the metaheuristic algorithm, and random initializations have been the common procedure. The percentage breakdown of the initialization procedures applied to the JSSP in the literature reviewed in this paper is illustrated in Figure 2.

Priority dispatching rules [27], [33] can be ranked second in terms of being an effective initialization technique according to the number of their applications in the literature [102], [39], [104], [55], [130], [91], [116], [66], [132], [133], [67], [98], [72].

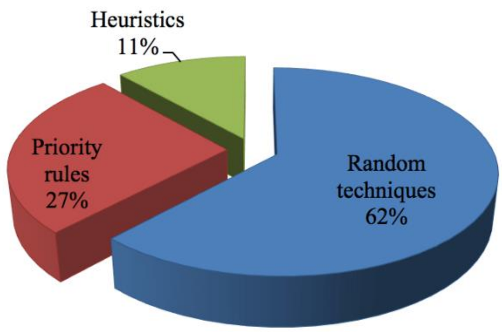


FIGURE 2. Percentage breakdown of initialization methods applied to the JSSP in the statistical population reviewed in this paper

The main advantages of priority rules are that they are easy to execute, have low requirements in terms of computational power and have less time complexity. However, each priority rule applied in initialization is able to generate just one solution. So to overcome this limitation, some researchers , [128], [130], [83], [131], [85]) have combined a set of priority rules and random selections to produce the initial population in their algorithms. Another application of priority rules to generate initial active schedules is the G&T algorithm [30]. Several researchers [128], [46], [125], [135] have applied priority rules as a means to select operations in the procedure of the G&T algorithm

Some **heuristic initialization procedures** have also been proposed for the JSSP [121], [122], [44], [108],[56], [134], [137], [97], [71]. Due to these heuristics often having complex structures and a slightly longer computational time in comparison to random techniques and priority rules they have attracted less interest. However, the significant advantage of heuristic initialization procedures is their ability to generate an initial population that is close to the optimal solution. This advantage speeds up the improvement algorithms enabling them to reach the optimal solution sooner and compensates for the abovementioned small amount of extra time spent on the initialization step. These heuristics are the fastest approximation algorithms available to solve JSSPs and therefore their solutions can be considered to be optimal solutions due to the very small deviation.

The generation of a random initial point and then the use of a tabu search mechanism to generate a set of initial solutions was the heuristic initialization procedure has been proposed [126],[127]. [56] proposed a heuristic initialization based on the weighted sum of priority rules called IPG. Their heuristic initialization was able to generate an initial population near the optimal solution and an enhanced genetic algorithm in their experiments. [134] designed a heuristic to produce an initial solution for a neural network (NN) which solved the crisp JSSP based on priority rules. They also implemented two random initialization methods for the NN and the results of the NN that was initialized by the proposed heuristic were better than the results that were initialized by random methods in terms of both quality and computational time. [131],[132] proposed a robust intelligent technique to produce an initial population that was close to the optimal solution for the JSSP.

2.3.3 Improvement

The last part of the metaheuristic for the JSSP is the improvement algorithm. This type of algorithm begins its procedure from initial solutions to find better solutions by exploring and exploiting the search space of the problem. To complete the procedures of the metaheuristic algorithm, many forms of intelligent approaches have been designed to improve the initial solutions of the JSSP, such as the evolutionary algorithm (EA), genetic algorithm (GA), artificial immune system (AIS), particle swarm optimization (PSO), ant colony optimization (ACO), simulated annealing (SA), Tabu Search (TS), Fuzzy Job Shop Scheduling Problem (FJSSP) and so on.

The **evolutionary algorithm** (EA) [68],[69], [70], [71], [122] utilizes procedures inspired by biological evolution, such as reproduction, mutation, recombination and selection. EAs work based on the notion of dynamically changing population due to the birth of new individuals inheriting genetic materials from parent. Candidate solutions of the JSSP play the role of individuals in a population and the fitness function determines their quality. The first effort in this area was an evolution-based mechanism that considers sequences of dispatching rules for job assignment along the lines of the shifting bottleneck procedure [103]. Next, the EA was hybridized with fuzzy logic [132] to solve the multi-objective flexible JSSP. This hybrid procedure exploits the knowledge representation competences of fuzzy logic and the adaptive competences of the EA.

Genetic algorithms (GAs) for the JSSP have been implemented by many researchers [53], [50], [84], [85], [58], [81], [52], [107], [95], [66], [127]. However, GAs fail to intensify the search through the most promising regions of a neighbourhood. Therefore when implementing a GA, a local search procedure is also usually applied to intensify the search [86],[102],[65],[70] hybridized a GA with a NN. One of the effective hybrid GAs can be found in the work of [67], who hybridized their proposed GA with a SA algorithm. [62] designed a GA that included a G&T algorithm [32] for the initial chromosome generation. [178] proposed a hybrid genetic TS which when implemented produced exceptional results in all benchmarks tested. This is indicated by the three new upper bounds found for the ABZ9, YN1 and YN2 benchmark instances. [45] implemented an efficient hybrid GA with a local search scheme based on a random keys representation. A three-component hybrid GA with a differential evolution and a variable neighbourhood search procedure was developed by [9].

An **artificial immune system** (AIS), which uses clonal selection, hypermutations and a library of antibodies to constructs solutions, was first applied to solve the JSSP by [55]. The AIS has the advantage of a powerful global exploration capability. Further work using the AIS for the JSSP has been conducted in this field [54], [98], [99], [75], [100]. An improved immune algorithm called the modified Taguchi-immune algorithm (MTIA), based on both the features of an AIS and the systematic reasoning ability of the Taguchi method, was proposed by [101]. Both [102] and [103] designed multi-model AISs which emulate the features of a biological immune system. Hybridization of the AIS with ACO was proposed by [104] and the next hybridization of the AIS is designed with a simple and fast SA by [105]. To solve the flexible JSSP, A [155] considered an AIS with different mutation operators to reproduce new individuals.

The usage of **particle swarm optimization** (PSO) methods for the JSSP is relatively recent, and there is growing interest in this optimization method [118]. Although PSO methods can be generally applied to continuous optimization problems, which means that they are therefore unsuitable for discrete problems such as the JSSP, some researchers have proposed various transformation methods for PSO so that this method can also be applied to JSSP. For instance, [61] proposed a PSO algorithm to solve the JSSP by converting the feature of a continuous domain to a discrete domain. [69] combined the PSO algorithm with SA and formed an early form of hybrid PSO. [124] proposed a hybrid PSO-SA algorithm and compared the hybrid with single algorithm implementations. Their hybrid method was found to be more effective and more efficient than single metaheuristic components. The combination of a PSO and TS is another hybrid version to solve the JSSP [94]. [56] designed a computationally effective algorithm which combines PSO with an AIS to solve the minimum makespan problem of the JSSP. Another hybrid PSO [123] was proposed based on a combination of PSO, a SA technique and a multitype individual enhancement scheme [168]. Also, PSO has been hybridized with local search [127].

The application of **ant colony optimization** (ACO) and **bee colony optimization** (BCO) algorithms for the JSSP is rather limited. The first implementations of ACO were proposed about two decades ago [109], [112]. The early implementations of BCO to solve the JSSP are comparatively recent [123]. A simple ACO was executed on the JSSP [120]. [112] presented a hybrid ACO-local search algorithm and focused on the infamous FT10 benchmark instance so that the ACO performance could be evaluated. Hybrid ACO-Fuzzy was proposed by [124] to solve JSSP in reasonable time. ACO was easy to modify for new constrains and multi-objectives cases.

Simulated annealing (SA) [225] is one of the earliest proposed metaheuristics. Several well-known implementations of SA have been proposed [116], [115], [126], [51], [132], [127]. However, the main disadvantage of SA for the JSSP is that it cannot reach good solutions for the JSSP quickly because it is a generic and memoryless technique [4]. In light of this, research on using SA for the JSSP has focused on hybrid versions of SA, some of which have been mentioned in the preceding paragraphs [123], [67], [69], [124]. [119] hybridized SA and TS to find elite solutions inside big valleys. The effectiveness of this method was proved by the finding of 17 new upper bounds for benchmark datasets. Another hybrid SA-TS algorithm has been proposed by [128], where the hybrid algorithm was employed for two individual memory modules; the first memory temporarily avoided further changes in a solution’s elements to keep the presence of good solution elements while the second memory tried to track good solutions found during an iteration so that the best ones could be utilized as the starting point in a subsequent iteration. A hybrid SA based on a novel immune mechanism was designed for the JSSP by [73], which was built based on the notion of attempting to bottleneck the jobs available in each schedule to improve the quality of the final schedules. [64] combined SA, a GA and local search to propose an effective algorithm for solving the JSSP. [72] designed a SA method based on using the features of a block-based neighbourhood.

One of the powerful metaheuristic algorithms to solve the JSSP is **tabu search** (TS), and most state-of-the-art algorithms for this problem include some sort of TS. The main advantage of TS lies in the use of memory in its procedure which speeds up the solution space search [218]. There are several notable early implementations of TS for the JSSP [119], [113], [126], [127]. [92] implemented one of the most well-known proposals for a TS for the JSSP called the TSAB. The computational effectiveness of the TSAB was demonstrated by its ability to solve the Ft10 benchmark instance in 30 seconds on a personal computer. This achievement motivated the development of *i*-TSAB [109], which uses the big valley of phenomena, some elements of path relinking philosophy, and new theoretical properties offers unprecedented. [117] and [96] analysed the contribution to the literature of the TSAB and i-TSAB, respectively. [127] found that the initialization procedure of the TSAB has an important influence on the best found solution of the method, whereas the TSAB strategy to generate a critical path has a negligible influence. [96] emphasized the importance of long-term memory in *i*-TSAB for achieving state-of-the-art performance levels in terms of metaheuristics for the JSSP. Due to the advantages of TS, some researchers have used TS in hybrid metaheuristics, some of which have been mentioned in the preceding paragraphs (e.g., [110], [113], [119]. An early hybrid implementation of TS that should also be mentioned is that of [117], who combined TS and the shifting bottleneck procedure, where the shifting bottleneck heuristic was initially applied to produce schedules and then TS was used to refine solutions in subsequent iterations. More recently, global equilibrium search, path relinking and improved TS based on a new version of the N6 neighbourhood have been combined for the JSSP [91]. Also, [63] have hybridized differential evolution with the TS that was presented by [129] for the *i*-TSAB to solve the JSSP.

Fuzzy Method. Scheduling problem have many uncertainty factors such as (a) the operations processing times, (b) due and release dates and (c) general constraint of precedence among the operations. Therefore, numerous studies have dealt with the operations processing times as fuzzy numbers and gave the constraint of precedence a tree or a collection of trees representation. Some researchers have delved into scheduling problems with uncertain data and utilized fuzzy numbers to address the issue of uncertainty [124], [130]. [131] introduced the ambiguity (fuzziness) of scheduling data into the classical job-shop scheduling problem and discussed the significance of providing answers to the fuzzy job-shop scheduling problem. Fuzzy scheduling models have bulk of attention among other scheduling problems in the scheduling research community. [132] studied the real-world scheduling problems and their data fuzzy nature such as the fuzziness in the operations processing time and their due-dates. In [232]'s study, a multi-objective JSSP for six jobs, six resources and ten jobs, ten resources have formulated. A two objective functions, the fuzzy completion time and the agreement index of fuzzy due-date were used. The researchers utilized fuzzy logic to combine the next machine's load (NML), CR, and SPT priority rules, to fulfil all objective functions. [133] presents the economic lot-size scheduling problem (ELSP) with fuzzy extension for the work fuzzy demands. In the exploration for the optimal or near-optimal solution, a genetic algorithm guided by the fuzzy feasibility constraints and fuzzy total cost function is designed and assists the ELSP. [134] effectively solved a single machine-scheduling problem by applying Fuzzy Logic to advance it to a real-world application. Fuzzy numbers to define operations processing times and due dates have been used, and two objectives: minimization of number of tardy jobs and average tardiness were employed. [73] reached a near-optimal solution for a large-scale JSSP by using hybrid optimization algorithm based on decomposition. The key objective was the total weighted tardiness minimization. To calculate the operations' bottleneck characteristic values, a fuzzy inference system is composed. These values are used to lead solving of sub-problem process in an immune mechanism to promote the efficiency of the optimal. [135] studied a single machine due date assignment scheduling problem with general constraint of precedence among the jobs and fuzzy processing times. Most recently, Fuzzy scheduling studies have been reviewed in [164], [127], [128].

3. CONCLUSSION

This review was conducted to bring together for ease of reference information on the constraints and objectives of the JSSP and the various methodologies that have thus far been applied to solve them. The methodologies were categorized into exact methods, heuristic approaches and metaheuristic algorithms. Significant effort was made to review the metaheuristic algorithms in particular. To achieve this purpose, the metaheuristics applied to the JSSP were divided based on the three steps of preprocessing, initialization and improvement. Preprocessing was reviewed in terms of two tasks: encoding the solution space and decoding the search space. For initialization, three approaches were covered: random initialization, the use of priority rules to generate initial solutions and heuristic initialization. The intelligent techniques applied to improve the initial solution/initial population were reviewed and their advantages and disadvantages were presented in **Error! Reference source not found.**.

With regards to the methodologies applied to solve the JSSPs, metaheuristic algorithms were found to be more effective than the other proposed methods. The encoding scheme and decoding algorithm are key to the successful solving of the JSSP because they affect the exploration and exploitation procedure in the solution space. However, the effective choice of an encoding scheme is dependent on the concepts and ideas applied to the initialization and improvement algorithms.

TABLE 1
ADVANTAGES AND DISADVANTAGES OF METAHEURISTIC ALGORITHMS PROPOSED FOR JSSPS

Algorithm	Advantages	Disadvantage
EA	Easy to adjust its chromosomes and operators for any class of JSSP	Long training times
	Incorporation of JSSP-specific knowledge in its chromosomes and operators	Does not have memory search based on its random techniques
	Easy to understand and no demand for complex knowledge of mathematics	Fails to intensify the search through the most promising regions of a neighbourhood
		Long computational time to converge the optimal solution of the JSSP
GA	Chromosomes share information with each other	Fails to intensify the search through the most promising regions of a neighbourhood
	Easy to tune the chromosomes and the genetic operators	Does not have memory search based on random techniques
	Easy to understand and no demand for complex knowledge of mathematics	Requires more computational time than other algorithms to reach the optimal solution
	Guarantees convergence to schedules near the optimal value	
MA	Has local search	Increases algorithmic complexity and CPU time compared to the GA
	Memes share knowledge and exchange experience	Sensitivity to local search volume and neighbourhood structure of memes on JSSP for avoiding
	Easy to understand and easy to implement	local optimal schedules
	Resistant to being trapped in local optima	Sensitivity to encoding memes on JSSP - Less memory
CA	Has brief space and memory search	Becomes easily trapped in local optimum schedules
	Has dual evolutionary system to cooperate between brief and population spaces	Difficult to design desirable protocol for determining the set of acceptable individuals that are
	Dual inheritance to make interactions by its acceptance and influence functions -	able to update the belief space of the JSSP
	High convergence speed	Difficult to propose an effective protocol for defining how the updated beliefs are able to
		influence the adaptation of the population component of the JSSP
AIS	Powerful global exploration capability	Simple way to solve a JSSP but may not be efficient in practice
	Self-organizing and simple training algorithm	Draws its strength from randomness
	Does not require effort to optimize any system parameters	Setup of the parameters is based on guesswork by the developer
BFA	Suitable for multiobjective JSSP because of its group foraging strategy	Has difficulty in tuning the parameters of the BFA for JSSPs
	Individual bacterium communicate with other	Best suited to continuous optimization problems and unsuitable for discrete problems such as
	Has memory and is suitable for global search	the JSSP
		Has weakness in its neighbourhood search

TABLE 1 (CONTINUED)
ADVANTAGES AND DISADVANTAGES OF METAHEURISTIC ALGORITHMS PROPOSED FOR JSSPS

Algorithm	Advantages	Disadvantage
ICA	Fast convergence Search process can be controlled by assimilation and revolution No demand for complex knowledge of mathematics	Best suited to continuous optimization problems and unsuitable for discrete problems such as the JSSP Has weakness in its neighbourhood search and Long computational time
PSO	Particles update themselves with internal acceleration and velocity Has memory and is suitable for global search Particles give out information to others	Has difficulty in tuning the parameters of the PSO for JSSPs Best suited to continuous optimization problems and unsuitable for discrete problems such as the JSSP - Has weakness in its neighbourhood search
SFLA	Each frog can be seen as a meme carrier Easy to understand with simple steps and a few parameters Possesses the advantages of genetic-based MA and social behaviour-based PSO	No uniform initial population Slow convergence rate to solve JSSPs Has weakness in local searching ability
ACO	Positive feedback Distributed computation avoids premature convergence Has colony memory	Performs poorly for JSSPs larger than 10 jobs and 10 machines No centralized processor to guide ACO towards good solutions Makeup is based on sequences of random decisions
BCO	Fast convergence and high flexibility Fewer setting parameters Memory of elite solutions by waggle dance	Premature convergence in the later search period Accuracy of its optimal results sometimes cannot meet requirements and it needs to apply local search algorithms
SA	Avoids becoming trapped in local optima Search process can be controlled by the cooling schedule Easy to implement	Cannot reach good solutions for JSSPs quickly Is a memoryless technique Has difficulty in defining an effective cooling schedule
TS	Has memory in its procedure, which is known as the tabu list Quick solution space search Local search based	Perfectly attracted to the big valley areas of the solution space Dependent on calculating the critical path in its neighbourhood search Cannot explore the whole solution space
EM	Possesses memory Has constructive cooperation among the particles Moves the points of population towards global optimality by using an attraction-repulsion mechanism	Suitable for continuous optimization problems with bounded variables Since it moves along the direction of total forces for JSSPs it cannot generate a feasible new solution Often easy to reach premature convergence

For example, machine-based encoding is suitable for any algorithm that uses the shifting bottleneck procedure, while random key encoding is useful for metaheuristics with continuous features such as PSO and EM. Job pair relation-based encoding is appropriate for NNs, which are categorized as machine learning algorithms and have been not considered in this review. Note that the choice of encoding scheme is not a limitation as such because every encoding scheme can be adapted and utilized for every algorithm, although it may not turn out to be a suitable choice.

There are two critical points to consider when generating the initial population of metaheuristic algorithms: the diversity of the initially produced solutions and their quality. The random initialization techniques satisfy the diversity property, so most previous researchers have applied this type of technique to the initialization of their proposed metaheuristic algorithms, although the quality of their initial solutions were far below. There are a few heuristic initialization procedures which are able to generate initial solutions that are close to the optimal solution but they do not always maintain a high level of diversity in the initial population. Therefore, the design of an effective heuristic initialization is a clear and significant research gap in the JSSP domain.

References

[1] A.Zai, A, Benmedjdoub, B, Boudhar, M. A branch and bound and parallel genetic algorithm for the job shop scheduling problem with blocking. International Journal of Operational Research, 2012;14(3): 343-365.

[2] Brucker, P, Burke, EK, Groenemeyer, S. A branch and bound algorithm for the cyclic job-shop problem with transportation. Computers & Operations Research. 2012b; 39(12): 3200-3214.

[3] Chen, B, Matis, TI. A flexible dispatching rule for minimizing tardiness in job shop scheduling. International Journal of Production Economics. 2012; 141(1): 360-365.

[4] Chiang, TC, & Fu, LC. Using dispatching rules for job shop scheduling with due date-based objectives. International Journal of Production Research. 2007; 45(14): 3245-3262.

[5] Lu, H, Huang, GQ, Yang, H. Integrating order review/release and dispatching rules for assembly job shop scheduling using a simulation approach. International Journal of Production Research. 2011; 49(3): 647-669.

[6] Moghaddam, RT, Daneshmand-Mehr, M. A Computer Simulation Model for Job Shop Scheduling Problems Minimizing Makespan. Computer & Industrial Engineering. 2005; 48: 811-823.

[7] Shahzad, A, Mebarki, N, IRCCyN, I. Discovering dispatching rules for job shop scheduling problem through data mining. Paper presented at the 8th International Conference of Modeling and Simulation, Tunisi; 2010.

[8] Subramaniam, V, Ramesh, T, Lee, G, Wong, Y, Hong, G. Job shop scheduling with dynamic fuzzy selection of dispatching rules. The International Journal of Advanced Manufacturing Technology. 2000; 16(10),:759-764.

[9] Liu, SQ, Kozan, E. A hybrid shifting bottleneck procedure algorithm for the parallel-machine job-shop scheduling problem. Journal of the Operational Research Society. 2011; 63(2): 168-182.

[10] Bülbül, K. A hybrid shifting bottleneck-tabu search heuristic for the job shop total weighted tardiness problem. Computers & Operations Research. 2011; 38(6): 967-983.

[11] Cheng, HC, Chiang, TC, Fu, LC. Multiobjective job shop scheduling using memetic algorithm and shifting bottleneck procedure. Paper presented at the IEEE Symposium on Computational Intelligence in Scheduling. CI-Sched'09. 2009.

[12] Amirthagadeswaran, KS, Arunachalam, VP. Improved Solutions for Job Shop Scheduling Problmes through Genetic Algorithm with a Different Method of Schedule Deduction. The International Journal of Advanced Manufacturing Technology. 2006; 28:532-540.

[13] Chandrasekaran, M, Asokan, P, Kumanan, S, Balamurugan, T, Nickolas, S. Solving job shop scheduling problems using artificial immune system. The International Journal of Advanced Manufacturing Technology. 2006; 31(5): 580-593.

[14] Coello, C, Rivera, D, Cortés, N. Use of an artificial immune system for job shop scheduling. Artificial Immune Systems. 2003: 1-10.

[15] Ge, H. W., Sun, L., Liang, Y. C., & Qian, F. An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans. 2008; 38(2): 358-368.

[16] Kuczapski, AM, Micea, MV, Maniu, LA, Cretu, VI. Efficient Generation of Near Optimal Initial Populations to Enhance Genetic Algorithms for Job-Shop Scheduling. Information Technology and Control. 2010; 39: 32-37.

[17] Li, J, Pan, Y. A hybrid discrete particle swarm optimization algorithm for solving fuzzy job shop scheduling problem. The International Journal of Advanced Manufacturing Technology. 2013; 66: 583-596.

[18] Ponsich, A, Coello, CA. A Hybrid Differential Evolution-Tabu Search Algorithm for the Solution of Job-Shop Scheduling Problems. Applied Soft Computing. 2013; 13(1): 462-474.

[19] Tamilarasi, A. An enhanced genetic algorithm with simulated annealing for job-shop scheduling. International Journal of Engineering, Science and Technology. 2010; 2(1): 144-151.

[20] Vela, CR, Varela, R, González, MA. Local search and genetic algorithm for the job shop scheduling problem with sequence dependent setup times. Journal of Heuristics. 2010; 16(2): 139-165.

[21] Wang, L, Tang, D. An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem. Expert Systems with Applications. 2011; 38(6): 7243-7250.

[22] Weckman, GR, Ganduri, CV, Koonce, DA. A neural network job-shop scheduler. Journal of intelligent Manufacturing. 2008; 19(2): 191-201.

[23] Yusof, R, Khalid, M, Hui, GT, Yusof, SM. Solving job shop scheduling problem using a hybrid parallel micro genetic algorithm. Applied Soft Computing. 2011; 11(8): 5782-5792.

[24] Zhang, R, Wu, C. A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective. Computers & Operations Research. 2011; 38(5): 854-867.

[25] Zhang, R, Wu, C. A hybrid approach to large-scale job shop scheduling. Applied intelligence. 2010; 32(1): 47-59.

[26] Zribi, N, El Kamel, A, Borne, P. Minimizing the makespan for the MPM job-shop with availability constraints. International Journal of Production Economics. 2008; 112(1): 151-160.

[27] Li, J, Pan, Q, Xie, S. An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems. Applied Mathematics and Computation. 2012; 218(18): 9353-9371.

[28] Zhang, R, Wu, C. A hybrid immune simulated annealing algorithm for the job shop scheduling problem. Applied Soft Computing. 2010; 10(1): 79-89.

[29] Cavory, G, Dupas, R, Goncalves, G. A genetic approach to solving the problem of cyclic job shop scheduling with linear constraints. European Journal of Operational Research. 2005; 161(1): 73-85.

[30] Cheung, W, Zhou, H. Using genetic algorithms and heuristics for job shop scheduling with sequence-dependent setup times. Annals of Operations Research. 2001; 107(1): 65-81.

[31] Essafi, I, Mati, Y, Dauzère-Pérès, S. A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem. Computers & Operations Research. 2008; 35(8): 2599-2616.

[32] Baptiste, P, Flamini, M, Sourd, F. Lagrangian bounds for just-in-time job-shop scheduling. Computers & Operations Research, 2008; 35(3): 906-915.

[33] Nasiri, MM, Kianfar, F. A GES/TS algorithm for the job shop scheduling. Computers & industrial engineering. 2012; 62(4): 946-952.

[34] Schuster, CJ, Framinan, JM. Approximative procedures for no-wait job shop scheduling. Operations Research Letters. 2003; 31(4): 308-318.

[35] Yazdani, M, Amiri, M, Zandieh, M. Flexible job-shop scheduling with parallel variable neighborhood search algorithm. Expert Systems with Applications. 2010; 37(1): 678-687.

[36] Zhang, G, Gao, L, Shi, Y. An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems with Applications. 2011; 38(4): 3563-3573.

[37] Zhang, G, Shao, X, Li, P, Gao, L. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Computers & industrial engineering. 2009; 56(4): 1309-1318.

[38] Zhou, H, Feng, Y, Han, L. The hybrid heuristic genetic algorithm for job shop scheduling. Computers & industrial engineering. 2001; 40(3): 191-200.

[39] Nakano, R, Yamada, T. Conventional genetic algorithm for job shop problems. Paper presented at the Proceedings of the Fourth International Conference on Genetic Algorithms; 1991.

[40] Ombuki, BM, Ventresca, M. Local search genetic algorithms for the job shop scheduling problem. Applied Intelligence. 2004; 21(1): 99-109.

[41] Vinod, V, Sridharan, R. Scheduling a dynamic job shop production system with sequence-dependent setups: An experimental study. Robotics and Computer-Integrated Manufacturing. 2008; 24(3): 435-449.

[42] Azadeh, A, Negahban, A, Moghaddam, M. A hybrid computer simulation-artificial neural network algorithm for optimisation of dispatching rule selection in stochastic job shop scheduling problems. International Journal of Production Research. 2012; 50(2):551-566.

[43] El-Bouri, A, Shah, P. A neural network for dispatching rule selection in a job shop. The International Journal of Advanced Manufacturing Technology. 2006; 31(3): 342-349.

[44] Brucker, P, Kampmeyer, T. A general model for cyclic machine scheduling problems. Discrete applied mathematics. 2008; 156(13): 2561-2572.

[45] Eswaramurthy, V, Tamilarasi, A. Hybridizing tabu search with ant colony optimization for solving job shop scheduling problems. The International Journal of Advanced Manufacturing Technology. 2009; 40(9): 1004-1015.

[46] Heinonen, J, Pettersson, F. Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem. Applied Mathematics and Computation. 2007; 187(2): 989-998.

[47] Huang, KL, Liao, CJ. Ant colony optimization combined with taboo search for the job shop scheduling problem. Computers & Operations Research. 2008; 35(4): 1030-1046.

[48] Zhang, J, Hu, X, Tan, X, Zhong, J, Huang, Q. Implementation of an ant colony optimization technique for job shop scheduling problem. Transactions of the Institute of Measurement and Control. 2006; 28(1): 93-108.

[49] Zhixiang, Y, Jianzhong, C, Yan, Y, Ying, M. Job shop scheduling problem based on DNA computing. Journal of Systems Engineering and Electronics. 2006; 17(3): 654-659.

[50] Wu, C-S, Li, D-C, Tsai, T-I. Applying the fuzzy ranking method to the shifting bottleneck procedure to solve scheduling problems of uncertainty. The International Journal of Advanced Manufacturing Technology. 2006; 31(1-2): 98-106.

[51] Gonzalez-Rodriguez, I, Vela, CR, Puente, J. A memetic approach to fuzzy job shop based on expectation model. Paper presented at the IEEE International Fuzzy Systems Conference, FUZZ-IEEE 2007. 2007.

[52] Yahyaoui, A, Fnaiech, N, Fnaiech, F. A Suitable Initialization Procedure for Speeding a Neural Network Job-Shop Scheduling. IEEE Transactions on Industrial Electronics. 2011; 58(3): 1052 - 1060.

[53] Yang, S, Wang, D. A new adaptive neural network and heuristics hybrid approach for job-shop scheduling. Computers & Operations Research. 2001; 28(10): 955-971.

[54] Yang, S, Wang, D, Chai, T, Kendall, G. An improved constraint satisfaction adaptive neural network for job-shop scheduling. Journal of Scheduling. 2010; 13(1): 17-38.

[55] Zhang, CY, Li, PG, Rao, YQ, Guan, ZL. A very fast TS/SA algorithm for the job shop scheduling problem. Computers & Operations Research. 2008; 35(1): 282-294.

[56] Fattahi, P, Saidi, M.M. A New Approach in Job Shop Scheduling: Overlapping Operation. Journal of Industrial Engineering. 2009a.

[57] Wang, L, Zhou, G, Xu, Y, Wang, S, Liu, M. An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. The International Journal of Advanced Manufacturing Technology. 2012; 60(1): 303-315.

[58] Wong, T, Ngan, S. A comparison of hybrid genetic algorithm and hybrid particle swarm optimization to minimize makespan for assembly job shop. Applied Soft Computing. 2013; 13(3): 1391-1399.

[59] Jamili, A, Shafia, M, Tavakkoli-Moghaddam, R. A hybridization of simulated annealing and electromagnetism-like mechanism for a periodic job shop scheduling problem. Expert Systems with Applications. 2011a; 38(5): 5895-5901.

[60] Moslehi, G, Mahnam, M. A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. International Journal of Production Economics. 2011; 129(1): 14-22.

[61] Naderi, B, Ghomi, S, Aminnayeri, M. A high performing metaheuristic for job shop scheduling with sequence-dependent setup times. Applied Soft Computing. 2010; 10(3): 703-710.

[62] Roshanaei, V, Balagh, AKG, Esfahani, MMS, Vahdani, B. A mixed-integer linear programming model along with an electromagnetism-like algorithm for scheduling job shop production system with sequence-dependent set-up times. The International Journal of Advanced Manufacturing Technology. 2010; 47(5): 783-793.

[63] Roshanaei, V, Naderi, B, Jolai, F, Khalili, M. A variable neighborhood search for job shop scheduling with set-up times to minimize makespan. Future Generation Computer Systems. 2009; 25(6): 654-661.

[64] Tavakkoli-Moghaddam, R, Taheri, F, Bazzazi, M, Izadi, M, Sassani, F. Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints. Computers & Operations Research. 2009; 36(12): 3224-3230.

[65] Zhang, R, Song, S, Wu, C. A hybrid artificial bee colony algorithm for the job shop scheduling problem. International Journal of Production Economics. 2013; 141(1): 167-178.

[66] Lin, TL, Horng, SJ., Kao, TW, Chen, YH, Run, RS, Chen, RJ, Kuo, I. An efficient job-shop scheduling algorithm based on particle swarm optimization. Expert Systems with Applications. 2010; 37(3): 2629-2636.

[67] Yin, M, Li, X, Zhou, J. An efficient job shop scheduling algorithm based on artificial bee colony. Scientific Research and Essays. 2011; 5(24): 2578-2596.

[68] Luh, GC, Chueh, CH. A multi-modal immune algorithm for the job-shop scheduling problem. Information Sciences, 2009; 179(10): 1516-1532.

[69] Qing, R, Wang, Y. A new hybrid genetic algorithm for job shop scheduling problem. Computers and Operations Research. 2012; 39(10): 2291-2299.

[70] Jamili, A, Shafia, MA, Tavakkoli-Moghaddam, R. A hybrid algorithm based on particle swarm optimization and simulated annealing for a periodic job shop scheduling problem. The International Journal of Advanced Manufacturing Technology. 2011b; 54(1): 309-322.

[71] Nie, L, Gao, L, Li, P, Li, X. A GEP-based reactive scheduling policies constructing approach for dynamic flexible job shop scheduling problem with job release dates. Journal of intelligent Manufacturing. 2012: 1-12.

[72] Abdolrazzagh-Nezhad, M, Abdullah, S. Robust start for population-based algorithms solving job-shop scheduling problems. Paper presented at the 3rd Conference on Data Mining and Optimization (DMO). 2011. Malaysia

[73] Abdullah, S, Abdolrazzagh-Nezhad, M. Fuzzy job-shop scheduling problems: A review. Information Sciences. 2014; 278(0): 380-407.

[74] Bagheri, A, Zandieh, M, Mahdavi, I, Yazdani, M. An artificial immune algorithm for the flexible job-shop scheduling problem. Future Generation Computer Systems, 2010; 26(4): 533-541.

[75] Tavakkoli-Moghaddam, R, Azarkish, M, Sadeghnejad, A. A new hybrid multi-objective pareto archive PSO algorithm for a classic job shop scheduling problem with ready time. Paper presented at the Advanced Intelligent Computing. 6th International Conference on Intelligent Computing Theories and Applications; 2010.

[76] Vilcot, G, Billaut, JC. A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. European Journal of Operational Research. 2008;190(2): 398-411.

[77] Chiang, T-C, & Lin, H-J. A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling. International Journal of Production Economics. 2013; 141(1): 87-98.

[78] Gu, J, Gu, M, Cao, C, Gu, X. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. Computers & Operations Research. 2010; 37(5): 927-937.

[79] Hu, Y, Yin, M, Li, X. A novel objective function for job-shop scheduling problem with fuzzy processing time and fuzzy due date using differential evolution algorithm. The International Journal of Advanced Manufacturing Technology. 2011; 56(9-12): 1125-1138.

[80] Lei, D, Wu, Z. Crowding-measure-based multiobjective evolutionary algorithm for job shop scheduling. The International Journal of Advanced Manufacturing Technology. 2006; 30(1-2): 112-117.

[81] Horng, S-C, Lin, S-S, Yang, F-Y. Evolutionary algorithm for stochastic job shop scheduling with random processing time. Expert Systems with Applications. 2012; 39(3): 3603-3610.

[82] Wang, X, Gao, L, Zhang, C, Shao, X. A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. The International Journal of Advanced Manufacturing Technology. 2010; 51(5-8), 757-767.

[83] Qing, R, Wang, Y. A new hybrid genetic algorithm for job shop scheduling problem. Computers and Operations Research. 2012; 39(10): 2291-2299.

[84] Burke, E, Silva, JL. The design of memetic algorithms for scheduling and timetabling problems. Recent Advances in Memetic Algorithms: Springer; 2005.

[85] Caumond, A, Lacomme, P, Tchernev, N. A memetic algorithm for the job-shop with time-lags. Computers & Operations Research. 2008; 35(7): 2331-2356.

[86] Cheng, HC, Chiang, TC, & Fu, LC. A two-stage hybrid memetic algorithm for multiobjective job shop scheduling. Expert Systems with Applications. 2011; 38(9): 10983-10998.

[87] Chiang, T-C, Fu, L-C. A rule-centric memetic algorithm to minimize the number of tardy jobs in the job shop. International Journal of Production Research.2008; 46(24): 6913-6931.

[88] Hasan, SK, Sarker, R, Essam, D, Cornforth, D. Memetic algorithms for solving job-shop scheduling problems. Memetic Computing. 2009; 1(1): 69-83.
[89] Qian, B, Wang, L, Huang, D-X., Wang, X. Scheduling multi-objective job shops using a memetic algorithm based on differential evolution. The International Journal of Advanced Manufacturing Technology. 2008; 35(9-10), 1014-1027.
[90] Yang, J-H, Sun, L, Lee, HP, Qian, Y, Liang, Y-C.. Clonal selection based memetic algorithm for job shop scheduling problems. Journal of Bionic Engineering. 2008; 5(2): 111-119.
[91] Nowicki, E, Smutnicki, C. An advanced tabu search algorithm for the job shop problem. Journal of Scheduling. 2005; 8(2): 145-159.
[92] Frutos, M, Olivera, AC, Tohmé, F. A memetic algorithm based on a NSGAII scheme for the flexible job-shop scheduling problem. Annals of Operations Research. 2010; 181(1): 745-765.
[93] Gao, L, Zhang, G, Zhang, Li, X. An efficient memetic algorithm for solving the job shop scheduling problem. Computers & Industrial Engineering. 2011; 60(4): 699-705.
[94] Ho, NB, Tay, JC. GENACE: An efficient cultural algorithm for solving the flexible job-shop problem. Paper presented at the Congress on Evolutionary Computation. 2004.
[95] Becerra, RL, Coello, CAC. A Cultural Algorithm for Solving the Job Shop Scheduling Problem. Knowledge Incorporation in Evolutionary Computation. Springer; 2005.
[96] Cortés Rivera, D, Landa Becerra, R, Coello Coello, CA. Cultural algorithms, an alternative heuristic to solve the job shop scheduling problem. Engineering Optimization, 2007; 39(1): 69-85.
[97] Lianghui, Z. Cultural Algorithm of Job Shop Scheduling Problem. Compute Engineering. 2009; 35(l3): 196-198.
[98] Xia, Z. Cultural particle swarm optimization algorithm for Job-Shop scheduling problem. Application Research of Computers. 2012; 4: 011.
[99] Raeesi, MR, Kobti, Z. A multiagent system to solve JSSP using a multi-population cultural algorithm Advances in Artificial Intelligence: Springer; 2012: pp. 362-367
[100] Feng, Y JSSZ. Job-Shop Scheduling Study by Dynamic Evaluation based Immune Algorithm. Chinese Journal of Mechanical Engineering. 2005; 3: 004.
[101] Ge, HW, Sun, L, Liang, Y-C. Solving job-shop scheduling problems by a novel artificial immune system Advances in Artificial Intelligence. Springer; 2005. pp: 839-842.
[102] Yu, J-J, Sun, S-D, Hao, J.-H. Multi objective flexible job-shop scheduling based on immune algorithm. Computer Integrated Manufacturing Systems-Beijing. 2006; 12(10): 1643.
[103] Tsai, JT, Ho, WH, Liu, TK, Chou, JH. Improved immune algorithm for global numerical optimization and job-shop scheduling problems. Applied Mathematics and Computation. 2007; 194(2): 406-424.
[104] Hong, L. A novel artificial immune algorithm for job shop scheduling. Paper presented at the International Conference on Computational Intelligence and Natural Computin. CINC'09. 2009.
[105] Luh, GC, Chueh, CH. A multi-modal immune algorithm for the job-shop scheduling problem. Information Sciences, 2009; 179(10): 1516-1532.
[106] Song, XJ, Lu, JY, Sui, M-l. Job-shop scheduling problems based on immune ant colony optimization. Journal of Computer Applications. 2007; 27(5): 1183-1186.
[107] Naderi, B., Khalili, M., & Tavakkoli-Moghaddam, R. A hybrid artificial immune algorithm for a realistic variant of job shops to minimize the total completion time. Computers & Industrial Engineering. 2009; 56(4): 1494-1501.

[108] Wu, C, Zhang, N, Jiang, J, Yang, J, Liang, Y. Improved bacterial foraging algorithms and their applications to job shop scheduling problems. Adaptive and Natural Computing Algorithms. Springer. 2007; pp. 562-569
[109] Jing-jinga, C, Yan-minga, S, Lan-xiub, C. Improved bacteria foraging optimization algorithm for Job-Shop scheduling problems [J]. Application Research of Computers. 2011; 9: 034.
[110] Zhao, F, Jiang, X, Zhang, C, Wang, J. A chemotaxis-enhanced bacterial foraging algorithm and its application in job shop scheduling problem. International Journal of Computer Integrated Manufacturing(ahead-of-print). 2014: 1-16.
[111] Narendhar, S, Amudha, T. A Hybrid Bacterial Foraging Algorithm For Solving Job Shop Scheduling Problems. arXiv preprint arXiv:1211.4971; 2012.
[112] Ge, H., & Tan, G. A Cooperative Intelligent Approach for Job-shop Scheduling Based on Bacterial Foraging Strategy and Particle Swarm Optimization. Computational Intelligence Systems in Industrial Engineering. Springer; 2012. pp: 363-383.
[113] Zhang, J, Zhang, P, Yang, JX, Huang, Y. Solving the job shop scheduling problem using the imperialist competitive algorithm. Paper presented at the Advanced Materials Research; (2012).
[114] Karimi, N, Zandieh, M. Najafi, A. Group scheduling in flexible flow shops: a hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism. International Journal of Production Research. 2011; 49(16): 4965-4977.
[115] Moradinasab, N, Shafaei, R, Rabiee, M, Ramezani, P. No-wait two stage hybrid flow shop scheduling with genetic and adaptive imperialist competitive algorithms. Journal of Experimental & Theoretical Artificial Intelligence. 2013; 25(2): 207-225.
[116] Seidgar, H, Kiani, M, Abedi, M, Fazlollahtabar, H. An efficient imperialist competitive algorithm for scheduling in the two-stage assembly flow shop problem. International Journal of Production Research. 2014; 52(4): 1240-1256.
[117] Shokrollahpour, E, Zandieh, M, Dorri, B. A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem. International Journal of Production Research. 2011; 49(11): 3087-3103.
[118] Goldansaz, SM, Jolai, F, Anaraki, AHZ.. A hybrid imperialist competitive algorithm for minimizing makespan in a multi-processor open shop. Applied Mathematical Modelling. 2013; 37(23): 9603-9616.
[119] Banisadr, A, Zandieh, M, Mahdavi, I. A hybrid imperialist competitive algorithm for single-machine scheduling problem with linear earliness and quadratic tardiness penalties. The International Journal of Advanced Manufacturing Technology, 2013; 65(5-8): 981-989.
[120] Zobolas, G, Tarantilis, C, Ioannou, G. Exact, Heuristic and Meta-heuristic Algorithms for Solving Shop Scheduling Problems. Metaheuristics for Scheduling in Industrial and Manufacturing Applications. Springer; 2008. pp. 1-40
[121] Ahandani, M, Shirjoposht, N, Banimahd, R. Job-shop scheduling using hybrid shuffled frog leaping. Paper presented at the 14th International CSI Computer Conference CSICC. 2009.
[122] Teekeng, W, Thammano, A. A combination of shuffled frog leaping and fuzzy logic for flexible job-shop scheduling problems. Procedia Computer Science. 2011; 6: 69-75.
[123] Junqing-Li, Quan-Ke Pan, Yun-Chia Liang. An effective hybrid tabu search algortihm for multi-objective flexible job shop scheduling problem. Computers &Industrial Engineering. 2010; 59(4):647-662.

[124] Chong, C.S, Low, MYH, Sivakumar, AI, Gay, KL. A bee colony optimization algorithm to job shop scheduling. Paper presented at the Simulation Conference. WSC 06. Proceedings of the Winter. 2006.

[125] Sezgin Kiliç, Cengiz Kahraman. Solution of a Fuzzy Flowshop Scheduling Problem Using a Necessity Measure. Journal of Multiple-valued Logic and Soft computing. 2016; 15(1):51-64

[126] El-Bouri, A, Azizi, N, Zolfaghari, S. A comparative study of a new heuristic based on adaptive memory programming and simulated annealing: The case of job shop scheduling. European Journal of Operational Research, 2007; 177(3): 1894-1910.

[127] Fayad, C, Petrovic. S. A Fuzzy Genetic Algorithm for Real-World Job Shop Scheduling. Dlm. M. Ali & F. Esposito (pnyt.). Ed. Innovations in Applied Artificial Intelligence 3533. pp. 524-533. Springer Berlin Heidelberg. 2005.

[128] Chen, C-M, C.-Y. Liu, M.-H. Chang. Personalized curriculum sequencing utilizing modified item response theory for web-based instruction. Expert Systems with Applications. 2006; 30(2): 378-396.

[129] Mehrabad, MS, A. Pahlavani. A fuzzy multi-objective programming for scheduling of weighted jobs on a single machine. The International Journal of Advanced Manufacturing Technology. 2009; 45(1-2): 122-139.

[130] Li, J, Q-K. Pan, PN. Suganthan, MF. Tasgetiren. Solving fuzzy job-shop scheduling problem by a hybrid PSO algorithm. Dlm. (pnyt.). Ed. Swarm and Evolutionary Computation, Springer; 2012. pp. 275-282.

[131] Petrovic, S, Fayad, C, Petrovic, D, Burke, E, Kendall, G. Fuzzy job shop scheduling with lot-sizing. Annals of Operations Research. 2008; 159(1): 275-292.

[132] Palacios, JJ, MA González, CR. Vela, I. González-Rodríguez, J. Puente. Genetic tabu search for the fuzzy flexible job shop problem. Computers & Operations Research. 2015; 54(0): 74-89.

[133] Palacios, JJ, J Puente, CR Vela, I. González-Rodríguez. Benchmarks for fuzzy job shop problems. Information Sciences. 2016; 329: 736-752.

[134] Tavakkoli-Moghaddam, R, Khalili, M, Naderi, B. A hybridization of simulated annealing and electromagnetic-like mechanism for job shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness. Soft Computing. 2009. 13(10): 995-1006.

[135] Chao, CW, Liao, CJ. A discrete electromagnetism-like mechanism for single machine total weighted tardiness problem with sequence-dependent setup times. Applied Soft Computing. 2012; 12(9): 3079-3087.